



**HAL**  
open science

# Modélisation géométrique de surfaces lisses: Design et Fairing

Stefanie Hahmann

► **To cite this version:**

Stefanie Hahmann. Modélisation géométrique de surfaces lisses: Design et Fairing. Interface homme-machine [cs.HC]. Institut National Polytechnique de Grenoble - INPG, 2001. tel-00011422

**HAL Id: tel-00011422**

**<https://theses.hal.science/tel-00011422>**

Submitted on 19 Jan 2006

**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.

# MEMOIRE d'HABILITATION

présenté par

**Stefanie Hahmann**

pour obtenir le grade de

**L'Habilitation à Diriger des Recherches**

de

**I'INSTITUT NATIONAL POLYTECHNIQUE DE  
GRENOBLE**

Spécialité: Informatique et Mathématiques Appliquées

## **Modélisation géométrique de surfaces lisses: Design et Fairing**

Date de soutenance: 21 décembre 2001

Composition du jury

Pr. Guy MAZARE	Président
Pr. Pere BRUNET	Rapporteur
Pr. Gerald FARIN	Rapporteur
Pr. Jean-Claude LEON	Rapporteur
Pr. Hans HAGEN	Examineur
Pr. Bernard LACOLLE	Examineur
Pr. Claude PUECH	Examineur

Habilitation préparée au sein du laboratoire

**LMC/IMAG**



*Meinen Eltern*



---

# Remerciements

Je tiens à adresser mes plus vifs remerciements aux membres de mon jury: Pere Brunet, Gerald Farin et Jean-Claude Léon qui ont accepté d'être mes rapporteurs, Hans Hagen, Bernard Lacolle et Claude Puech que je remercie pour l'intérêt qu'ils portent à mes travaux, ainsi que Guy Mazaré qui m'a fait l'honneur de présider ce jury.

Les travaux décrits dans ce mémoire ont été effectués au sein de l'équipe Modélisation Géométrique et Approximation du Laboratoire de Modélisation et Calcul à Grenoble. Je tiens à remercier Bernard Lacolle, alors chef d'équipe, pour l'accueil dans son équipe lors de mon arrivée à Grenoble, pour les excellentes conditions de travail, et pour la confiance qu'il m'a accordée.

Je remercie chaleureusement les étudiants dont j'ai dirigé les recherches et qui ont contribué à ces travaux, en particulier Stefan Konz, Riadh Taleb et Alex Yvart.

Un grand merci à Georges-Pierre Bonneau, Nicolas Szafran et Luc Biard pour leur aide et le soutien qu'ils m'ont apportés.

Enfin, je tiens à remercier François Robert, Jean-Louis Soler et Jean Della Dora de m'avoir accueillie si chaleureusement à l'ENSIMAG et de m'avoir offert un environnement d'enseignement idéal, très enrichissant et ouvert.

---

# Table des matières

<b>Synthèse: Modélisation géométrique de surfaces lisses</b>	<b>5</b>
<b>1 Surfaces définies sur des triangulations polygonales quelconques</b>	<b>11</b>
1.1 Continuité géométrique et problème de compatibilité du twist	13
1.2 Etat de l'art	15
1.3 Vers une modélisation hiérarchique de surfaces triangulaires lisses	18
<b>2 Interpolation lisse régulière par subdivision de triangles</b>	<b>20</b>
2.1 Raccordement lisse en un sommet de la triangulation	20
2.2 L'algorithme en 3 étapes	23
2.3 Interpolation de normales — localité stricte du schéma	27
<b>3 Paramètres de forme</b>	<b>28</b>
3.1 Nécessité de paramètres de forme	28
3.2 Généralisation du 4-split	29
3.3 Choix des paramètres de forme	32
<b>4 Interpolation lisse généralisée</b>	<b>35</b>
4.1 Motivation	35
4.2 Compatibilité du twist	36
4.3 Construction des courbes frontière et des tangentes trans-frontière	38
4.4 Exemples	39
<b>5 Design hiérarchique</b>	<b>41</b>
5.1 Hiérarchisation de surfaces linéaires par morceau	42
5.2 Modèle hiérarchique de surfaces lisses	45
5.3 Exemples	48

<b>6 Lissage de surfaces</b> .....	<b>51</b>
6.1 Importance d'une surface lisse .....	51
6.2 Mesures et algorithmes .....	52
6.3 Outils de contrôle visuel .....	54
<b>7 Une approche heuristique pour le lissage</b> .....	<b>55</b>
7.1 Lissage local et itératif de surfaces B-spline bicubiques .....	56
7.2 Recherche heuristique d'une solution optimale .....	59
<b>8 Fairing local et itératif utilisant des masques</b> .....	<b>62</b>
8.1 Mesure de fairness .....	62
8.2 Une étape de lissage et l'algorithme "FairC3" .....	64
8.3 Extension de l'étape de fairing: utilisation de masques de lissage .....	65
<b>9 Bilan et perspectives</b> .....	<b>68</b>
<b>Références</b> .....	<b>71</b>

## **Articles et publications de 1995-2001** **77**

<b>1 Triangular <math>G^1</math> interpolation by 4-splitting domain triangles</b> .....	<b>79</b>
<b>2 Localizing the 4-split method for <math>G^1</math> free-form surface fitting</b> .....	<b>109</b>
<b>3 Smooth irregular mesh interpolation</b> .....	<b>125</b>
<b>4 Polyhedral Modeling</b> .....	<b>137</b>
<b>5 Polynomial surfaces interpolating arbitrary triangulations</b> .....	<b>147</b>
<b>6 Knot-removal surface fairing using search strategies</b> .....	<b>171</b>
<b>7 Shape improvement of surfaces</b> .....	<b>181</b>
<b>8 Visualization techniques for surface analysis</b> .....	<b>201</b>





## **Première Partie**

# **Synthèse: Modélisation géométrique de surfaces lisses**



---

# Introduction

Les travaux de recherche décrits dans ce mémoire d'habilitation se situent dans le domaine de la modélisation géométrique (**Computer Aided Geometric Design**). Ce domaine trouve son origine dans le besoin en courbes et surfaces de forme libre dans la technologie CAD/CAM, et est devenu ces dernières années un domaine de recherche important de l'informatique et des mathématiques appliquées, dont la dimension applicative est cruciale.

Un modèle géométrique est nécessaire par exemple lorsque l'ordinateur est utilisé pour la conception et la fabrication d'une pièce de carrosserie. Il est constitué d'un ensemble de données structurées, et doit en particulier décrire la géométrie de l'objet. A cette fin, on utilise presque toujours les courbes et surfaces paramétriques qui sont l'objet principal de mes recherches. Le modèle géométrique est d'une manière générale à la base du traitement graphique de données. Le champ d'application de ces algorithmes, loin de se réduire à la CFAO (Conception et Fabrication Assistées par Ordinateur), s'étend à des domaines tels que la géographie, avec les méthodes dites de "Scattered Data", à la médecine, par la reconstruction tridimensionnelle de données RMN, à l'industrie du film grâce aux techniques d'animation, à l'architecture...

## Modélisation de Surfaces Lisses

Les recherches développées dans ce mémoire ne concernent que des travaux post-doctoraux et leur sujet est bien distinct de ceux traités pendant la thèse. Il ne s'agit donc pas de prolongations des travaux de thèse.

Ces travaux portent sur le problème de la modélisation de surfaces lisses. L'importance qui est donnée au caractère lisse d'une surface dépend de l'utilisation prévue de la surface, i.e. de l'application. L'exigence en terme de qualité est par exemple très grande dans l'industrie automobile où la moindre imperfection est inacceptable pour la surface d'une voiture. Les surfaces de très haute qualité sont d'ailleurs dites de "classe A" dans le langage de certains concepteurs de logiciels CAO. Historiquement c'est dans ce secteur industriel que sont nées les méthodes de base pour la conception de courbes et surfaces de forme libre: Bézier (Renault), De Casteljau (Citroën), Gordon (General

Motors), Coons (Ford). L'utilisation des technologies CAD/CAM basées sur des surfaces de forme libre type B-spline, s'est imposée dans tous les logiciels de conception assistée. Beaucoup d'objets banals de la vie courante, que ce soit par exemple un grille pain, un sèche cheveux, ou un téléphone portable sont le résultat d'un processus de design basé sur ce type de surfaces. Les surfaces de ces objets, bien que leur objectif principal ne soit pas de plaire visuellement mais de satisfaire une certaine fonctionnalité, ne doivent pas pour autant comporter de défauts. Ces surfaces doivent être *esthétiquement lisses*. D'autres domaines d'applications comme l'aéronautique ou la construction navale nécessitent également des surfaces lisses, mais cette fois pour des raisons fonctionnelles et techniques. Le carénage d'un avion ou la coque d'un bateau par exemple ne doivent pas présenter trop d'oscillations, afin de respecter certaines propriétés aéro- ou hydrodynamiques.

Les travaux décrits dans ce mémoire abordent la modélisation de surfaces lisses selon deux approches différentes. La première approche consiste à faire en sorte - dès le début du processus de conception - que des contraintes liées au lissage de la surface soient respectées. La seconde approche se situe à un stade plus aval du processus de conception: elle consiste à modifier une surface déjà existante, de manière à la rendre lisse, tout en la déformant le moins possible.

### **Surfaces lisses de topologie quelconque**

Modéliser des surfaces lisses de topologie quelconque a été une préoccupation permanente en CAGD (Computer Aided Geometric Design). En effet les paramétrisations classiques de surfaces basées sur un produit tensoriel ne sont pas adaptées à la représentation de topologies quelconques. Une première vague de travaux de recherche a consisté à développer des interpolants polynomiaux de triangulations polygonales quelconques. Mais ces travaux n'ont pas donné de résultats convaincants, principalement à cause du découpage non régulier effectué sur chaque triangle donné en entrée. Plus tard sont apparues les surfaces de subdivision, qui peuvent approximer ou interpoler un maillage de topologie quelconque. Ces surfaces rencontrent actuellement un succès grandissant dans les logiciels destinés à l'animation (Maya d'Alias-Wavefront, Softimage) et plus récemment sont en passe d'être introduites dans les logiciels de CAD/CAM\*. Mais ces surfaces ont le défaut de ne pas avoir de paramétrisation explicite. De plus les surfaces de subdivision interpolantes produisent de mauvais résultats si la triangulation donnée en entrée n'est pas régulière.

L'ambition des recherches menées sur les *surfaces lisses de topologie quelconque* a été de revenir à la source de la première vague de travaux conduits sur les interpolants polynomiaux, mais en introduisant un autre découpage, régulier cette fois ci, des triangles donnés en entrée. Ce découpage régulier a permis d'obtenir de meilleurs résultats que ceux donnés par les précédentes méthodes polynomiales.

---

\* comme le montre la thèse CIFRE de Loïc Le Feuvre avec Dassault Systèmes, que j'encadre depuis septembre 2001.

Un autre avantage fondamental de ce nouveau découpage régulier a été de permettre d'envisager une hiérarchisation des procédés d'interpolation. Les méthodes d'interpolation polynomiales déjà existantes, basées sur un découpage en trois de chaque triangle, par insertion d'un sommet à l'intérieur, ne permettaient en effet pas l'établissement d'un schéma hiérarchique. L'application hiérarchique de ce découpage en trois conduit très rapidement à des triangles presque dégénérés, et à un schéma numériquement instable. Contrairement à ces méthodes, le découpage régulier utilisé dans les nouveaux interpolants présentés dans ce mémoire ne conduit à aucune dégénérescence des triangles. Cette différence cruciale a conduit à rechercher l'établissement d'un schéma d'interpolation invariant par subdivision, i.e. qui donne le même résultat qu'il soit appliqué à une triangulation ou à cette même triangulation après une certaine subdivision régulière. Il devient alors possible d'avoir un *modèle de surface multirésolution, paramétrique, et permettant de décrire des topologies quelconques*, applicable dans des domaines aussi divers que la CAO, la réalité virtuelle ou la médecine. Les modèles multirésolution de topologie quelconque ont donné lieu à de nombreuses publications ces dernières années. Ils permettent en effet d'éditer des modèles complexes à différentes résolutions, de représenter de manière compacte des modèles présentant des caractéristiques à des échelles diverses, ou encore d'accélérer le rendu des surfaces complexes. Le nouveau modèle hiérarchique développé dans ce mémoire permet ainsi toutes ces applications, tout en possédant une paramétrisation polynomiale explicite.

### **Lissage de surfaces**

Le développement d'une nouvelle méthode pour l'interrogation de la qualité de surfaces, capable de détecter des imperfections, lors de ma thèse a ensuite tout naturellement conduit à s'intéresser aux algorithmes permettant d'améliorer la qualité d'une surface déjà construite. Le critère de qualité est dans ce cas l'apparence subjective "lisse" de la surface, d'où le nom *lissage de surfaces* pour ce type d'algorithme.

Deux méthodes de lissage, basées sur la diminution des variations des courbures, vont être introduites. Les premières utilisent des stratégies de recherche heuristiques ou systématiques. Les secondes reposent sur la convolution du réseau des points de contrôle de la surface par des filtres discrets, imposant la continuité de classe  $C^3$  aux nœuds de la surface.

Ce mémoire est structuré de la manière suivante:

#### Chapitre 1

Ce chapitre donne une introduction au thème de la construction de surfaces lisses de topologie quelconque, en exposant le problème de twist que soulève le raccord d'un nombre arbitraire de facettes, et en donnant un bref état de l'art sur les différentes approches développées jusqu'à présent. Les motivations des travaux présentés dans la première partie de ce mémoire concluent ce chapitre.

## Chapitre 2

Une première méthode d'interpolation des sommets d'un réseau triangulaire de topologie arbitraire par une surface  $G^1$  polynomiale par morceaux est présentée. L'originalité de la méthode consiste à effectuer une subdivision en quatre des triangles, appelée 4-split. La surface a une représentation explicite de Bézier triangulaire de degré 5, est affinement invariante, de support local et possède un certain nombre de paramètres de forme. Une première généralisation de la méthode permet en particulier d'accroître la localité du schéma d'interpolation.

## Chapitre 3

Différents travaux sur l'emploi optimal des paramètres de forme de ce type d'interpolant, ainsi qu'une généralisation en vue de libérer les contraintes sur les courbes du bord, faisant de cette méthode un outil de design de formes de topologie quelconque performant, sont présentés dans ce chapitre.

## Chapitre 4

Les travaux précédents incluant l'interpolant polynomial et ses généralisations au niveau des paramètres de forme se sont pourtant avérés ne pas être exploitables pour la multirésolution comme nous l'avions souhaité initialement. Seule la méthode d'interpolation irrégulière qui fait l'objet de ce chapitre est appropriée aux applications multirésolution, car elle possède la propriété d'invariance par subdivision. Plus précisément, l'interpolation appliquée à une certaine subdivision de la triangulation donnée en entrée donne le même résultat que l'interpolation appliquée à la triangulation non subdivisée.

## Chapitre 5

Ce chapitre expose le développement d'un modèle hiérarchique de surfaces lisses basé sur les algorithmes d'interpolation précédemment introduits. La structure de données pour le codage de l'information topologique et géométrique est d'abord décrite pour le cas de surfaces linéaires par morceau et ensuite adaptée aux surfaces lisses.

## Chapitre 6

Une introduction au deuxième thème de ce mémoire - le lissage de surfaces déjà existantes - ainsi qu'aux méthodes de contrôle visuel de la qualité d'une surface est donnée.

## Chapitres 7 et 8

Deux types de méthodes de lissage de surfaces B-spline, basées sur la diminution des variations des courbures, sont introduites. Les premières utilisent des stratégies de recherche heuristiques ou systématiques. Les secondes reposent sur la convolution du réseau des points de contrôle de la surface par des filtres discrets, imposant la continuité de classe  $C^3$  aux nœuds de la surface.

Ce mémoire se conclue par un bilan et des perspectives au chapitre 9.

---

## Chapitre 1

# Surfaces définies sur des triangulations polygonales quelconques

La construction d'une surface à partir d'un ensemble de points contrôlant intuitivement sa forme est devenue, en CAGD, la méthode la plus populaire pour la modélisation de surfaces de forme libres (free-form surfaces). Une surface est généralement définie 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-tensoriel ou triangulaires (Bézier ou NURBS produit tensoriel, B-splines triangulaires, Box-splines) [4, 27, 48, 78, 3]. Ces surfaces, en particulier les NURBS (non uniform rational B-splines) se sont imposées dans les modélisateurs géométriques professionnels pour le design de surfaces de forme libre comme CATIA, Euclide, Cascade, Ideas, mais aussi dans les logiciels destinés à l'infographie comme Maya, Softimage.

Il n'est pourtant pas possible de représenter toute forme géométrique complexe par une seule surface produit tensoriel. Par exemple, quand le bord d'une surface doit avoir l'allure d'une marche d'escalier comme illustré en fig. 1-gauche, alors une représentation produit tensoriel unique ne peut la modéliser sans l'introduction de distorsions non souhaitables de la paramétrisation. Une autre difficulté apparaît lors de la création de formes géométriques lisses de topologie non rectangulaires comme la configuration triangulaire d'un simple coin de valise, ou comme la configuration pentagonale d'un coin de maison avec le trottoir, illustrées en fig. 1. Il est toujours possible de remplir ces régions avec des facettes rectangulaires, mais cela introduit des sommets voisins à trois ou cinq facettes, ce qui est incompatible avec une subdivision régulière du domaine planaire requise pour les surfaces produit tensoriel. Des solutions particulières ont été développées pour des configurations bien précises [84, 30, 86, 102]. D'autres solutions plus générales, comme les S-patches [61] ne sont pas utilisables en pratique car leur évaluation est trop complexe.



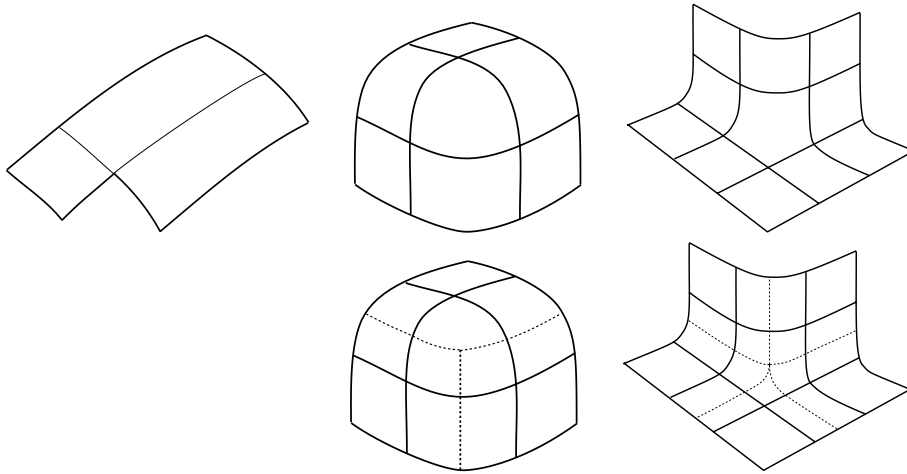


figure 1: Configuration non produit-tensoriel.

D'autre part et de manière plus générale, les définitions classiques de surfaces citées plus haut ne permettent pas la modélisation de surfaces de type topologique arbitraire, p. ex. les surfaces fermées, les surfaces avec une poignée ou avec des branches. Des données triangulées peuvent par contre représenter des types topologiques plus généraux de manière unifiée, comme illustre fig. 2.

Si ces surfaces apparaissent encore rarement en CAO, elles sont courantes en réalité virtuelle, pour la modélisation de personnages, en médecine, pour la modélisation de prothèses, et de manière générale pour la reconstruction tridimensionnelle d'objets réels.

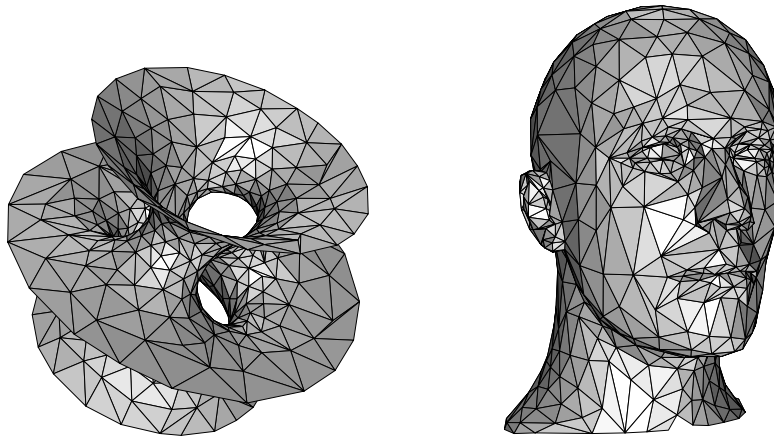


figure 2: Triangulations polygonale de type topologique arbitraire.

Depuis une dizaine d'années deux types de modélisation surfaciques basés sur des triangulations arbitraires se sont développés en parallèle, avec des applications (initialement) bien opposées:

- *les surfaces paramétriques* interpolant une triangulation polygonale quelconque,
- *les surfaces de subdivision* approximant ou interpolant un maillage triangulaire ou rectangulaire.

Les algorithmes de subdivision raffinement récursivement un maillage, jusqu'à convergence vers une surface lisse. Les schémas non-interpolants [12, 17, 60]

gènèrent des formes lisses et agréables, ils sont rapides et d'une complexité réduite, ce qui explique leur récent succès dans la communauté informatique graphique. Les schémas de subdivision triangulaires interpolants [20, 104] ne produisent pas toujours des surfaces de qualité suffisante. De manière plus générale, les surfaces de subdivision n'admettent pas de définition analytique. C'est la raison pour laquelle elles ne sont actuellement pas adaptées à la modélisation géométrique en vue d'une fabrication numériquement assistée comme le sont les surfaces paramétriques. Certains travaux très récents sur les contraintes de bord [58], l'évaluation en un paramètre [94], ou encore l'intersection et le trimming [59] tentent néanmoins d'adapter les surfaces de subdivision au domaine de la CAO.

Les travaux présentés dans la première grande partie de ce mémoire aux chapitres 2-5 concernent des *surfaces paramétriques interpolantes*. Ces surfaces interpolent un ensemble de points de  $\mathbb{R}^3$  arbitrairement distribués, dont le type topologique est déterminé par une triangulation polygonale surfacique reliant les points. Cette triangulation peut être quelconque. Elle consiste en un ensemble de faces triangulaires, d'arêtes et de sommets ainsi qu'en les informations d'adjacences entre ces entités, le tout devant représenter une variété de dimension 2. Les surfaces sont triangulaires par morceau, et sont construites en bijection avec les facettes triangulaires du maillage. Elles sont lisses dans le sens où elles ont un plan tangent continu entre les facettes. Une telle continuité est appelé continuité géométrique d'ordre 1, et est notée  $G^1$ . Ce type de surfaces est le mieux adapté aux problèmes de CAO, de reconstruction 3D, de modélisation de formes complexes et de topologie quelconque pour la médecine, ou la réalité virtuelle.

Dans ce chapitre d'introduction, nous commençons en sect. 1.1 par préciser la définition de la continuité géométrique, et par éclairer les difficultés que soulève le raccord d'un nombre arbitraire de facettes. Dans la section 1.2 les travaux sont passés en revue dans un état de l'art avec discussion. Enfin la section 1.3 précise les motivations des travaux présentés dans la première partie de ce mémoire qui sont liées au développement d'un schéma d'interpolation surfacique paramétrique hiérarchique.

### 1.1) Continuité géométrique et problème de compatibilité du twist

Le concept de continuité géométrique s'est révélé être très avantageux par rapport à celui de continuité paramétrique. Il est plus souple et géométriquement plus significatif. De plus, le raccordement paramétrique  $C^1$  entre facettes surfaciques dépend de la paramétrisation des facettes, et n'est donc pas invariant par transformation des paramètres. On dit que deux facettes surfaciques de courbe frontière commune se raccordent avec la continuité  $G^1$  si le plan tangent est bien définie le long de la courbe commune. Ceci implique qu'en chaque point de la courbe frontière commune, toutes les dérivées premières

sont co-planaires.

Nous allons brièvement traiter la continuité  $G^1$  entre deux facettes avant d'expliquer par la suite le problème qui se pose lorsque l'on veut raccorder un nombre arbitraire de facettes partageant un sommet. La méthode de résolution de ce problème, appelé *incompatibilité du twist*, permet de classer les différents schémas d'interpolation de triangulations polygonales.

### Continuité $G^1$ entre deux facettes triangulaires [23, 27, 18, 80]

Soient  $S^{i-1}$  et  $S^i$  deux facettes de Bézier triangulaires de degré  $n$ . La facette  $S^{i-1}$  (resp.  $S^i$ ) est paramétrée par  $(u_{i-1}, u_i)$  (resp.  $(u_i, u_{i+1})$ ). Les notations correspondent à celles données dans la figure 3.

Deux facettes  $S^{i-1}$  et  $S^i$  se raccordent avec continuité  $G^1$  si et seulement si

$$(C1) \quad S^{i-1}(0, u_i) = S^i(u_i, 0)$$

(C2) Il existe des fonctions réelles  $\Phi_i(u_i), \mu_i(u_i), \nu_i(u_i)$  telles que

$$\Phi_i(u_i)S_{u_i}^i(u_i, 0) = \nu_i(u_i)S_{u_{i+1}}^i(u_i, 0) + \mu_i(u_i)S_{u_{i-1}}^{i-1}(0, u_i),$$

$$(C3) \quad S_{u_i}^i(u_i, 0) \times S_{u_{i+1}}^i(u_i, 0) \neq 0,$$

$$(C4) \quad \mu_i(u_i)\nu_i(u_i) > 0.$$

$S_{u_{i+1}}^i$  et  $S_{u_{i-1}}^{i-1}$  sont les dérivées trans-frontière de  $S^{i-1}$  et  $S^i$  respectivement. La condition (C1) exprime la continuité  $C^0$  entre deux patches. La condition (C2) traduit la coplanarité des vecteurs tangents. La condition (C3) assure un plan tangent bien défini, et la condition (C4) garantie la bonne répartition des dérivées transverses sur ce plan tangent de part et d'autre de la courbe frontière commune.

### Problème de compatibilité du twist

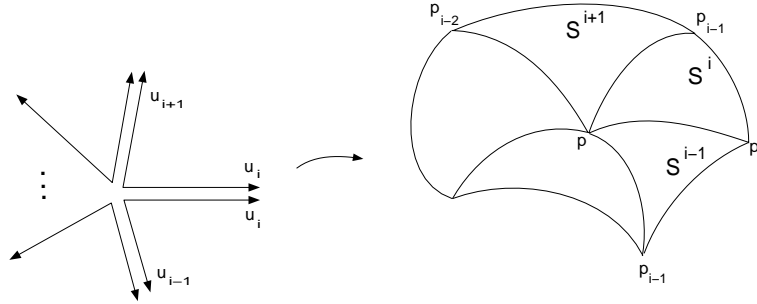


figure 3: Paramétrisation de  $n$  patches autour d'un sommet commun.

Considérons à présent le problème du raccordement  $G^1$  de  $n$  patches de Bézier  $S^i$ ,  $i = 1, 2, \dots, n$  ayant un sommet commun  $p$ , comme illustré en fig. 3. La condition (C2) doit être satisfaite entre toutes les paires de patches  $S^{i-1}, S^i$ ,  $i = 1, \dots, n$  (les indices sont toujours pris modulo  $n$  autour d'un sommet de degré  $n$ ). En dérivant l'équation (C2) par rapport à  $u_i$ , et en évaluant en  $u_i = 0$ , il vient:

$$\begin{aligned} \Phi_i'(0)S_{u_i}^i(0, 0) + \Phi_i(0)S_{u_i u_i}^i(0, 0) &= \nu_i'(0)S_{u_{i+1}}^i(0, 0) + \mu_i'(0)S_{u_{i-1}}^{i-1}(0, 0) \\ &+ \nu_i(0)S_{u_i u_{i+1}}^i(0, 0) + \mu_i(0)S_{u_{i-1} u_i}^{i-1}(0, 0), \quad i = 1, \dots, n. \end{aligned}$$

Ce système d'équations fait apparaître deux types de quantités: celles qui ne dépendent que des courbes séparant les patches ( $S_{u_i}^i(0,0)$ ,  $S_{u_i u_i}^i(0,0)$ ,  $S_{u_{i+1}}^i(0,0)$ ,  $S_{u_{i-1}}^{i-1}(0,0)$ ), et celles qui font intervenir les dérivées croisées ( $S_{u_{i-1} u_i}^{i-1}(0,0)$ ,  $S_{u_i u_{i+1}}^i(0,0)$ ). Ainsi si l'on suppose que les courbes séparant les patches sont fixées, alors un système d'équations liant les dérivées croisées, encore appelées twists, doit être vérifié. Or il se trouve que la nature cyclique de ce système d'équations fait qu'il ne possède en général pas de solution lorsque le degré  $n$  du sommet  $p$  est pair. C'est le problème de compatibilité du twist. La manière de résoudre ce problème est le point principal qui différencie les diverses approches que nous allons présenter dans la section suivante.

## 1.2) Etat de l'art

Pour commencer cet état de l'art, nous énumérons quatre propriétés géométriques, qu'il est souhaitable de voir vérifier par un algorithme d'interpolation de données par des surfaces lisses. Ces propriétés déterminent en grande partie la qualité des résultats obtenus, ainsi que la faisabilité et l'intérêt pratique des algorithmes. Par la suite, les différents travaux sont groupés selon la méthode de résolution du problème de compatibilité du twist, décrit dans la section précédente. Enfin une discussion portant sur un comparatif de la qualité des résultats des divers algorithmes conclue cet état de l'art.

### Propriétés géométriques

- 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. La forme de la surface peut ainsi être contrôlée localement, et son édition peut être effectuée en temps réel.
- 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.
- La *représentation polynomiale explicite* et un *degré bas* des patches : chaque patch est donné par un petit nombre de points de contrôle, ceci permet un stockage optimisé de la surface.
- Les *paramètres de forme* offerts par la méthode permettent de satisfaire automatiquement certaines propriétés géométriques tout en contrôlant localement la forme de la surface.

### Résolution de la compatibilité du twist

Le point principal différenciant toutes les méthodes existantes est la résolution du problème de compatibilité du twist. Cinq approches principales se distinguent clairement et regroupent la quasi totalité des travaux portant sur l'interpolation de triangulations polygonales.

*L'approche par subdivision Clough-Tocher* (Farin [24], Piper [80], Shirman et Séquin [92]) construit plusieurs sous-facettes pour chaque face du réseau. Chaque triangle domaine est subdivisé en trois sous-triangles en joignant le barycentre avec les trois sommets, c'est le découpage dit "à la Clough-Tocher". Le problème de compatibilité du twist est ainsi déplacé des sommets au centre de chaque triangle où il est résolu.

*L'approche par combinaisons convexes* (Nielson [72], Gregory [30], Hagen [33]), résout le problème de twist par l'introduction de quelques termes rationnels dans une combinaison convexe de plusieurs facettes polynomiales. Le dénominateur du twist devient nul, ce qui implique un twist non défini, et par là même supprime le problème de compatibilité du twist.

*L'approche par facettes implicites* (Bajaj et Ihm [1]) utilise la classe des facettes polynomiales implicites avec une singularité en chaque sommet.

*L'approche par facettes dégénérées* (Peters [77], Bohl et Reif [9]) utilise la classe des facettes polynomiales dégénérées en chacun de ces sommets afin de faciliter la solution du problème de compatibilité du twist.

*L'approche des courbes frontières* (Herron [47], Peters [76], Loop [62]) consiste à choisir des courbes frontière compatibles avec le problème du twist.

Les nouveaux schémas que nous développons dans ce mémoire et dont la motivation ainsi qu'une description détaillée suivra, apportent deux nouvelles approches à cette liste:

Une *approche par 4-split* ([41, 42, 43, 8]) qui combine l'idée de la subdivision avec celles des courbes frontière compatibles en introduisant une nouvelle subdivision du triangle domaine pour résoudre le problème de twist.

Une *approche hiérarchique* ([44, 45, 103]) qui permet le calcul des courbes frontière généralisées et qui pour la première fois rend le schéma d'interpolation hiérarchisable.

## **Discussion**

Si un interpolant offre des paramètres de forme libres, il est toujours possible, à posteriori, d'appliquer un algorithme de fairing qui généralement améliore la qualité de la surface. Le critère lisse (ou fair en anglais) a un caractère global, il est donc en contradiction avec la nature "locale" des schémas. Ce qui importe pour un schéma d'interpolation de triangulation n'est donc pas seulement la capacité théorique de produire des surfaces lisses, mais c'est également les possibilités concrètes pour le choix des paramètres de forme produisant des surfaces lisses, sans pour autant entrer en contradiction avec la localité des algorithmes. La plupart des algorithmes ne sont d'ailleurs pas satisfaisants à cet égard.

Une discussion critique des différentes approches donne finalement:

- de manière générale, le problème d'interpolation avec continuité  $G^1$  reste un problème difficile.

- en particulier, les méthodes de subdivision de type Clough-Tocher 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 permettant de choisir les courbes frontière sans contraintes ainsi que l'utilisation de facettes polynomiales de bas degré. Mann [68] et Hoschek/Lasser [48] qualifient la méthode de Shirmann/Séquin [92, 93] comme celle donnant les meilleurs résultats visuels.
- les méthodes par combinaisons convexes ont l'inconvénient de donner des surfaces rationnelles de degré assez élevé. D'après Mann et al. [68], les patches de Gregory permettent d'obtenir des qualités à peu près satisfaisantes. Dans la thèse de Du [18], ces surfaces rationnelles ont été utilisées pour la reconstruction 3D. Mais tout calcul sur la surface (dérivation, courbures, lignes de réflexion, etc) devient très coûteux.
- 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 secondes 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. 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 compatibilité du twist est originale.
- La méthode de Peters impose sur les courbes frontière la contrainte d'être  $C^2$ -compatibles avec des secondes formes fondamentales spécifiées aux sommets. Cette méthode ne possède pas suffisamment de paramètres libres afin de contrôler localement la forme de la surface finale. Elle présente néanmoins une originalité théorique dans la résolution du problème de compatibilité du twist, avec des facettes triangulaires et/ou quadrilatères.

De manière générale, Mann et al. [68] concluent dans leur article que les méthodes par subdivision et combinaisons convexes souffrent toutes de problèmes de même nature: il peut y avoir des ondulations - même à l'intérieur des patches, et des zones plates. Les courbures fortes se concentrent autour des sommets et le long des courbes tandis que l'intérieur des patches reste relativement plat. L'origine probable de ces phénomènes, d'après les auteurs, semble être la forme inadaptée et la mauvaise paramétrisation des courbes frontière, qui induiraient une distribution non-uniforme des courbures se répercutant à l'intérieur des patches.

### 1.3) Vers une modélisation hiérarchique de surfaces triangulaires lisses

Mes contributions sur les surfaces paramétriques interpolant un maillage triangulaire quelconque décrites dans ce mémoire ont été motivées par plusieurs facteurs. Les travaux de recherche sur les triangulations polygonales pour la visualisation ou l'infographie (modélisation, reconstruction 3D, animation), notamment autour des algorithmes de construction, de simplification, d'optimisation et de lissage [88, 46, 51 50, 49, 98, 16], ont connu récemment une croissance exponentielle. Parallèlement, depuis une dizaine d'années maintenant, les méthodes de multi-résolution ont connu également un succès grandissant en modélisation géométrique et en visualisation [96, 66, 87, 22, 5, 6].

Dans cette mouvance, la motivation centrale des travaux présentés dans la première partie de ce mémoire a été le développement d'un *schéma multi-résolution pour des surfaces paramétriques*, applicable dans des domaines aussi divers que la CAO, la réalité virtuelle ou la médecine. De précédents travaux ont permis de franchir une première étape vers ce but [29], mais ils sont restreints à l'utilisation de surfaces produit-tensoriels, et ne peuvent donc pas modéliser des surfaces complexes de topologie arbitraire, telles que celles requises dans les domaines d'applications visés.

Dans la mise au point de ce schéma multi-résolution de surfaces paramétriques, deux éléments sont à distinguer: d'une part le domaine de définition de la surface, et sa hiérarchisation, et d'autre part, l'algorithme d'interpolation lui-même.

- *Triangulations Hiérarchiques.*

Compte tenu de la nécessité de modéliser des surfaces de topologie arbitraire, le domaine de paramétrisation qui s'est imposé est une triangulation polygonale. Comme il a été évoqué plus haut dans ce chapitre d'introduction, ces triangulations permettent en effet de décrire les types de surfaces complexes qui nous intéressent. Ceci étant acquis, il devient nécessaire pour mettre au point un modèle hiérarchique, de pouvoir en un premier temps hiérarchiser le domaine de paramétrisation choisi. De l'état de l'art des méthodes d'interpolation de triangulation par des surfaces polynomiales se dégageait le fait que tous ces algorithmes sont basés sur un découpage en trois des triangles, par insertion d'un sommet à l'intérieur du triangle, et connexion de ce nouveau sommet avec les trois sommets du triangle (découpage dit "à la Clough-Tocher"). Ce découpage ne peut évidemment pas être utilisé dans un schéma hiérarchique, puisqu'il conduirait très rapidement à l'introduction de triangles quasi-dégénérés, et donc à des algorithmes numériquement instables. Par contre, le découpage - ou subdivision - régulier d'un triangle en quatre, par connexion des milieux des arêtes (découpage dénommé "4-Split"), est un procédé de hiérarchisation stable d'une triangulation. Ce procédé est d'ailleurs largement utilisé dans le domaine des surfaces de subdivision [20], ou celui des ondelettes

géométriques triangulaires [64]. C'est la raison pour laquelle ce type de subdivision a été choisi pour le découpage du domaine de paramétrisation triangulaire.

- *Interpolant invariant par subdivision.*

Une fois le domaine de paramétrisation choisi (triangulation polygonale), et le procédé de subdivision fixé (subdivision 4-split), un algorithme d'interpolation des triangulations hiérarchiques ainsi définies doit être déterminé. En plus des propriétés souhaitables déjà évoquées plus haut dans ce chapitre (paramétrisation polynomiale, bas degré, localité du schéma, ...), il devient nécessaire d'assurer l'invariance du schéma relativement à la subdivision de la triangulation. L'utilisateur d'un système interactif basé sur ce schéma doit pouvoir subdiviser le domaine de paramétrisation sans pour autant voir de modifications sur la surface résultante.

Ces deux éléments ont été le fil conducteur des travaux présentés dans la première partie de ce mémoire (chap. 2-5). Ils expliquent en particulier l'évolution vers plus de généralité des différents interpolants présentés, dans le but d'en obtenir un qui satisfasse la condition d'invariance par subdivision nécessaire à la hiérarchisation du schéma.



---

**Chapitre 2****Interpolation lisse régulière  
par subdivision de triangles**

Rappelons tout d'abord que les caractéristiques de *localité* et de *représentation polynomiale de bas degré* sont des propriétés qu'un schéma d'interpolation d'une triangulation polygonale devrait posséder pour des raisons de rapidité d'évaluation, et de facilité d'exécution de calculs sur la surface. Le problème de compatibilité du twist pourrait aisément être résolu avec une approche du type Clough-Tocher (découpage en trois du triangle domaine par introduction d'un sommet à l'intérieur). Mais une telle approche serait inexploitable dans une structure hiérarchique. Une subdivision répétée du domaine paramétrique de base mènerait en effet au bout de quelques étapes à des triangles quasi-dégénérés, donc à des problèmes de stabilité numérique. Cela nous a amené à choisir une nouvelle direction.

Ce chapitre présente une méthode originale d'interpolation d'une triangulation polygonale quelconque, qui est locale, polynomiale de bas degré (degré 5), affinement invariante et possède des paramètres de forme. L'originalité de la méthode consiste à effectuer une subdivision en quatre des triangles, appelée **4-split** dans la suite, en joignant le milieu des arêtes des triangles du domaine de paramétrisation. Tout d'abord nous allons exposer en section 2.1 comment le problème de compatibilité du twist se pose concrètement dans notre cas, comment il est résolu, et en quoi le 4-split des triangles domaine en permet une résolution satisfaisante. Ensuite nous présenterons en sect. 2.2 l'algorithme sans rentrer en détail dans les formules mathématiques. En sect. 2.3 une première généralisation de la méthode de base sera brièvement évoquée. Elle vise en particulier à accroître la localité du schéma d'interpolation.

**2.1) Raccordement lisse en un sommet de la triangulation**

Il a déjà été mentionné dans l'état de l'art du chapitre précédent, que la concep-

tion d'un interpolant polynomial d'une triangulation polygonale quelconque se focalise autour de la résolution du problème de compatibilité du twist. Ce premier paragraphe formalisera ce problème, et permettra aussi de fixer les notations qui seront utiles tout au long de ce mémoire.

#### 4-split

L'idée générale de la méthode d'interpolation par 4-split consiste à faire correspondre à chaque triangle en entrée, quatre patches polynomiaux formant une entité nommée **macro-patches**, comme illustré en fig. 4. Il existe donc une bijection entre les triangles d'entrée et les macro-patches. A l'intérieur d'un macro-patch, chacun des patches polynomiaux est paramétré sur un des sous-triangles obtenus par subdivision 4-split. Les arêtes des triangles originaux sont ainsi associées aux courbes frontière des macro-patches, qui sont polynomiales par morceau.

Les patches de la surface que nous souhaitons construire sont construits localement autour des sommets. Soit  $\mathbf{p} \in \mathbb{R}^3$  un sommet et  $\mathbf{p}_1, \dots, \mathbf{p}_n$  ses sommets voisins ordonnées selon l'orientation de la triangulation. L'entier  $n$  est appelé le degré du sommet  $\mathbf{p}$ .  $S^1, \dots, S^n$  désignent les  $n$  macro-patches autour d'un sommet  $\mathbf{p}$ . La paramétrisation ainsi choisie est illustré en fig. 4.

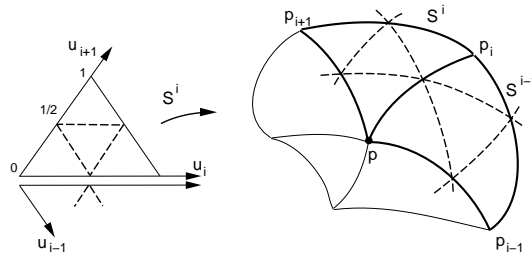


figure 4: Paramétrisation des patches autour d'un sommet.

#### Les fonctions scalaires

Pour que la surface finale ait un plan tangent bien définie en tout point, la condition (C2) de continuité  $G^1$  doit être satisfaite entre les facettes (voir sect. 1.1). Notre méthode pour assurer un raccord  $G^1$  entre des patches polynomiaux a été au départ influencé par les travaux de Loop [62]. Par contre, le schéma de Loop lui même n'est pas conçu pour l'interpolation, mais seulement pour l'approximation d'une triangulation polygonale. Cela est dû au fait que dans l'algorithme de Loop le bas degré polynomial et la localité du schéma imposent de trop fortes restrictions aux courbes frontière, impliquant une oscillation des courbes lorsque la triangulation est interpolée.

Faisons d'abord les mêmes hypothèse que Loop sur les fonctions scalaires intervenant dans la continuité  $G^1$ :  $\nu \equiv \mu \equiv \frac{1}{2}$  et  $\Phi_i(0) =: \Phi^0$ ,  $\Phi'_i(0) =: \Phi^1$ . Les conditions de continuité  $G^1$  entre deux facettes  $S^{i-1}$  et  $S^i$  s'écrivent alors de la manière suivante :

$$\Phi_i(u_i) S_{u_i}^i(u_i, 0) = \frac{1}{2} S_{u_{i+1}}^i(u_i, 0) + \frac{1}{2} S_{u_{i-1}}^{i-1}(0, u_i), \quad (1)$$

$i = 1, \dots, n(\text{mod } n)$ . Comme illustré en fig. 3, l'indice  $i$  fait référence à l'arête  $i$  entre les sommets  $\mathbf{p}$  et  $\mathbf{p}_i$ .

Avant de pouvoir construire le raccord  $G^1$  en un sommet commun à  $n$  patches, la fonction  $\Phi$  doit être déterminée. Pour cela on exprime les équations de continuité (1) en  $u_i = 0$ . On obtient le système de  $n$  équations suivant :

$$M_c \bar{\mathbf{r}}^1 = 0, \quad (2)$$

où

$$M_C = \begin{bmatrix} \Phi^0 & -\frac{1}{2} & 0 & \cdots & 0 & -\frac{1}{2} \\ -\frac{1}{2} & \Phi^0 & -\frac{1}{2} & & & 0 \\ 0 & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & & & 0 \\ 0 & & & & & -\frac{1}{2} \\ -\frac{1}{2} & 0 & \cdots & 0 & -\frac{1}{2} & \Phi^0 \end{bmatrix}, \quad \bar{\mathbf{r}}^1 = \begin{bmatrix} S_{u_1}^1(0,0) \\ \vdots \\ S_{u_n}^n(0,0) \end{bmatrix}.$$

Le déterminant de la matrice  $M_C$  devant s'annuler, cela permet de calculer la valeur de  $\Phi^0 = \Phi_i(0) = \cos(\frac{2\pi}{n})$ . Elle dépend du degré  $n$  du sommet considéré. Le même calcul peut être effectué au sommet  $\mathbf{p}_i$ , on obtiendrait la valeur  $\Phi(1) = 1 - \cos(\frac{2\pi}{n_i})$  qui dépend de  $n_i$ , le degré de ce sommet. La fonction  $\Phi_i$  doit interpoler ces valeurs aux extrémités tout en respectant la localité du schéma et un bas degré. C'est un des passages cruciaux de notre construction  $G^1$  [41], qui se différencie de celle de Loop.

Si nous prenions  $\Phi_i(u_i)$  linéaire alors  $\Phi'_i(0)$  dépendrait de  $n_i$ , et par conséquent les twists (dont le calcul fait intervenir la dérivée  $\Phi'_i(0)$ ) dépendraient aussi des  $n_i$  et le schéma ne serait plus local. Donc pour séparer le calcul des dérivées des  $\Phi_i$  des valeurs associées aux sommets voisins, le degré des  $\Phi_i$  devrait être  $\geq 2$ , comme dans Loop, si nous choissions  $\Phi_i$  en un seul morceau polynomial. Etant donnée que le degré des facettes est lié au degré des  $\Phi_i$  par les conditions (1), cela aboutirait à des facettes de degré trop élevé. Afin d'obtenir une surface de plus bas degré possible, il est donc souhaitable de trouver  $\Phi_i$  de degré minimal, tout en séparant les calculs autour d'un sommet de ceux autour des sommets voisins.

Le 4-split du triangle domaine nous permet de concilier ces deux objectifs, en choisissant la fonction  $\Phi_i$  **linéaire par morceau**. Ainsi la dérivée de  $\Phi_i$  en 0 reste indépendante de la valeur de  $\Phi_i$  en 1, tout en conservant un degré 1 pour la fonction  $\Phi_i$ . C'est précisément cette possibilité qui fait que le découpage 4-split permet une résolution satisfaisante du problème de compatibilité du twist.

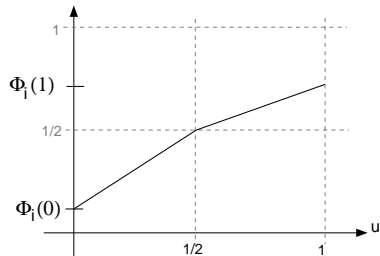


figure 5: fonction scalaire  $\Phi_i$ , linéaire par morceau.

### Raccord $G^1$ en un sommet commun à $n$ facettes

Les conditions de continuité  $G^1$  en un sommet de degré  $n$  de la triangulation consistent en les  $n$  équations (2) mettant en relation les **dérivées premières** des courbes frontière au sommet ( $r_i^1 = S_{u_i}^i(0, 0)$ ). Les **dérivées secondes** et les twists sont aussi impliqués lorsque l'on veut établir la continuité  $G^1$  en un sommet. Pour le voir il suffit de dériver les équations (1) par rapport à  $u_i$  et de les évaluer en  $u_i = 0$ , ce qui donne sous forme matricielle:

$$M_T \bar{t} = \Phi^1 \bar{r}^1 + \Phi^0 \bar{r}^2, \quad (3)$$

où

$$M_T = \begin{bmatrix} \frac{1}{2} & 0 & \cdots & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \cdots & & 0 \\ & & \ddots & & \\ 0 & \cdots & & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \cdots & & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}, \quad \bar{t} = \begin{bmatrix} t_1 \\ \vdots \\ t_n \end{bmatrix} = \begin{bmatrix} S_{u_1 u_2}^1(0, 0) \\ \vdots \\ S_{u_n u_1}^n(0, 0) \end{bmatrix},$$

$$\bar{r}^1 = \begin{bmatrix} S_{u_1}^1(0, 0) \\ \vdots \\ S_{u_n}^n(0, 0) \end{bmatrix}, \quad \bar{r}^2 = \begin{bmatrix} S_{u_1 u_1}^1(0, 0) \\ \vdots \\ S_{u_n u_n}^n(0, 0) \end{bmatrix}.$$

Notons que la matrice  $M_T$  est circulante [15]. Elle est singulière quand le degré  $n$  du sommet est pair. Il n'y a donc en général pas de solution de ce système pour les twists  $\bar{t}$  quand les dérivées premières  $\bar{r}^1$  et secondes  $\bar{r}^2$  des courbes frontière ont été fixées arbitrairement auparavant. C'est le **problème de compatibilité du twist**.

Le point clef pour le résoudre dans ce cas est de construire des courbes frontière telles que les vecteurs colonne  $\bar{r}^1$  et  $\bar{r}^2$  appartiennent à l'espace image de la matrice  $M_T$ . La solution  $\bar{t}$  du système contient alors les vecteurs twist compatibles avec le réseau de courbes.

### 2.2) L'algorithme en 3 étapes

Notre algorithme, dont les détails sont donnés en [41], s'exécute en trois étapes:

- (1) construction des courbes frontière,
- (2) construction des tangentes trans-frontière,
- (3) remplissage des facettes.

#### (1) Construction des courbes frontière

Les courbes frontière sont construites en correspondance avec les arêtes de la triangulation en entrée. Cette étape est 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 (2), (3),
- 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 indépendantes de celles en les sommets opposés. Les données relatives aux courbes qui entrent dans les systèmes (2) et (3) 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 degré 3** constituée de deux segments se joignant avec la continuité  $C^1$ . Ici encore, le 4-split est à l'origine de la baisse du degré.

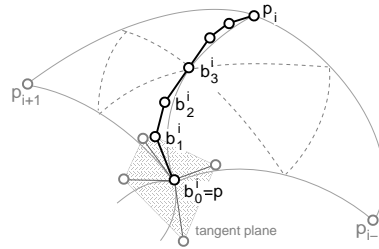


figure 6: Points de contrôle des courbes frontière incidentes au sommet  $p$ .

Adoptons pour la suite les notations matricielles suivantes pour les points de Bézier des courbes frontière entre  $p$  et  $p_i$ ,  $i = 1, \dots, n$ :

$$\bar{\mathbf{b}}_0 := \begin{bmatrix} \mathbf{b}_0^1 \\ \vdots \\ \mathbf{b}_0^n \end{bmatrix}, \quad \bar{\mathbf{b}}_1 := \begin{bmatrix} \mathbf{b}_1^1 \\ \vdots \\ \mathbf{b}_1^n \end{bmatrix}, \quad \bar{\mathbf{b}}_2 := \begin{bmatrix} \mathbf{b}_2^1 \\ \vdots \\ \mathbf{b}_2^n \end{bmatrix}, \quad \bar{\mathbf{p}} := \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_n \end{bmatrix}, \quad \mathbf{p} := \begin{bmatrix} \mathbf{p} \\ \vdots \\ \mathbf{p} \end{bmatrix}.$$

Le choix des points de Bézier des segments de courbe incidents au sommet  $p$  doit être conforme au système (2) et (3). Dans [41] il est montré en détail que les points suivants vérifient ces deux systèmes, i.e. définissent des dérivées premières et secondes qui sont dans l'espace image de  $M_T$ , et des dérivées premières qui sont dans le noyau de  $M_C$ . Le problème de compatibilité du twist est ainsi résolu, les courbes frontière sont dites "**twist compatibles**":

$$\begin{aligned} \bar{\mathbf{b}}_0 &= \alpha \bar{\mathbf{p}} + B^0 \bar{\mathbf{p}}, \\ \bar{\mathbf{b}}_1 &= \alpha \bar{\mathbf{p}} + B^1 \bar{\mathbf{p}}, \\ \bar{\mathbf{b}}_2 &= [(\gamma_0 + \gamma_1)\alpha + \frac{\gamma_2}{3}] \bar{\mathbf{p}} + B^2 \bar{\mathbf{p}}, \end{aligned} \tag{4}$$

où  $B^0, B^1, B^2$  sont des matrices  $n \times n$  définies par :

$$\begin{aligned} B_{ij}^0 &= \frac{1 - \alpha}{n} \\ B_{ij}^1 &= \frac{1 - \alpha + \beta \cos\left(\frac{2\pi(j-i)}{n}\right)}{n} \\ B_{ij}^2 &= \frac{(\gamma_0 + \gamma_1)(1 - \alpha) + \gamma_1 \beta \cos\left(\frac{2\pi(j-i)}{n}\right)}{n} + \gamma_2 \begin{cases} 1/6 & \text{si } j = i - 1, i + 1 \\ 1/3 & \text{si } j = i \\ 0 & \text{sinon.} \end{cases} \end{aligned} \tag{5}$$

Les points de contrôle  $\mathbf{b}_0^k, \mathbf{b}_1^k, \mathbf{b}_2^k$  et  $\mathbf{b}_3^k = \mathbf{b}_3^i$  du segment de courbe opposé, sont calculés en appliquant les formules (4) et (5) aux sommets voisins  $\mathbf{p}_i$  de  $\mathbf{p}$ .  $k$  est l'indice de  $\mathbf{p}$  relativement au voisinage de  $\mathbf{p}_i$ .

*Remarque : Paramètres libres*

Les paramètres libres  $\alpha, \beta, \gamma_1, \gamma_2$  contrôlent l'interpolation et les dérivées premières et secondes. Il est important pour une apparence globalement lisse de la surface de les choisir de manière optimale. Nous reviendrons sur ce problème dans le chapitre 3.

*Remarque : tangentes régulières au sommet*

Les formules (4) et (5) impliquent que les points  $\mathbf{b}_1^i$  et  $\mathbf{b}_2^i$  sont disposés, à une transformation affine près, sur les sommets d'un polygone régulier autour de  $\mathbf{p}$ . C'est ce point qui distingue principalement les résultats présentés dans ce chapitre et les chapitres suivantes. Dans le chapitre 3, cette contrainte de régularité sera relâchée pour les dérivées secondes, mais maintenue pour les dérivées premières; alors que dans le chapitre 4 elle sera relâchée pour toutes les dérivées.

## (2) Construction des tangentes trans-frontière

Une fois les courbes frontières compatibles aux twists trouvées, la deuxième étape vers la construction d'un réseau de facettes  $G^1$ -continues consiste à définir les tangentes trans-frontières pour chaque courbe frontière. Ces tangentes sont déterminées par une formulation de la continuité  $G^1$  équivalente à (1) :

$$\begin{aligned} S_{u_{i+1}}^i(u_i, 0) &= \Phi_i(u_i)S_{u_i}^i(u_i, 0) + \Psi_i(u_i)\mathbf{V}_i(u_i), \\ S_{u_{i-1}}^{i-1}(0, u_i) &= \Phi_i(u_i)S_{u_i}^i(u_i, 0) - \Psi_i(u_i)\mathbf{V}_i(u_i). \end{aligned} \tag{6}$$

Les fonctions scalaires  $\Psi_i$  et les fonctions vectorielles  $\mathbf{V}_i$  sont déterminées par trois conditions:

- continuité  $G^1$  le long de la courbe frontière,
- les contraintes de twist aux points extrémités,
- être consistantes avec les courbes frontières.

Ces conditions fixent certaines valeurs de ces fonctions qui ensuite doivent être interpolées tout en maintenant le degré de ces fonctions le plus bas possible. Ici encore le 4-split est important: au lieu d'obtenir le produit  $\Psi_i(u_i)\mathbf{V}_i(u_i)$  en un seul morceau polynomial de degré 4, le 4-split nous permet de choisir pour ce terme **deux morceaux polynomiaux de degré 3** se raccordant continûment.

*Remarque : Le degré des facettes*

Les rubans de tangence sont construits sous forme cubique par morceaux ainsi que les courbes frontière. Ceci laisse penser les sous-facettes pourraient être quartiques. Mais il s'est avéré impossible en général d'obtenir des facettes quartiques sans que le schéma ne devienne global. Ce point a été traité plus

précisément dans la thèse de Taleb [97]. Les sous-facettes doivent ainsi être de degré 5.

### (3) Remplissage des macro-facettes

Dorénavant, les macro-facettes sont considérées individuellement. La subdivision 4-split du domaine mène à la construction de quatre facettes triangulaires par macro-facette. Nous ne donnerons pas les valeurs des points de contrôle dans ce mémoire mais expliquerons brièvement comment les calculer. La rangée frontière et la première rangée intérieure des points de contrôle de la macro-facette sont obtenues à partir des courbes frontières et des tangentes trans-frontières. Ces points assurent un raccordement  $G^1$  entre les macro-facettes voisines. Pour les formules exactes de ces points, voir [41]. Les points de contrôle intérieurs restants sont choisis de telle manière à réaliser un raccordement de continuité  $C^1$  entre les quatre sous-facettes.

Les conditions de la continuité  $C^1$  entre les sous-facettes internes d'une macro-facette sont illustrées en fig. 7. Toute paire de triangle le long d'une courbe interne doit constituer un parallélogramme, voir [27, 4].

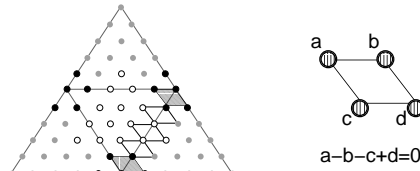


figure 7: Conditions de parallélogrammes pour le raccordement  $C^1$  entre les quatres sous-facettes le long des trois courbes internes d'une macro-facette.

Nous avons pu montrer [41], que la première et la dernière paire de triangles adjacents le long de chaque courbe (voir fig. 7) 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. 8.a),
- (b) calculer les troisième et le quatrième points de Bézier le long de chaque courbe interne en utilisant la seconde et la quatrième condition de parallélogramme (voir fig. 8.b),
- (c) choisir arbitrairement les trois points de Bézier inconnus de la facette centrale, ils sont aussi des paramètres de forme libres (voir fig. 8.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. 8.d).

La surface est ainsi globalement  $G^1$  continue. Six points de contrôle par macro-facette restent libres pour un contrôle supplémentaire de sa forme.

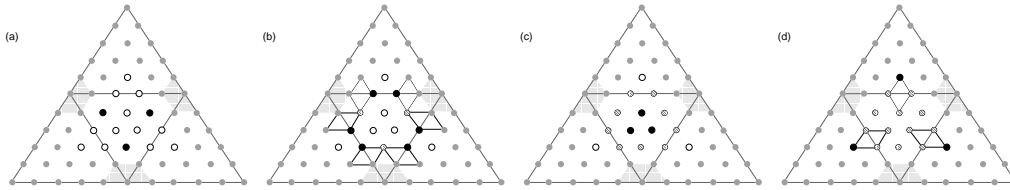


figure 8: Quatre étapes pour remplir la macro-facette avec la continuité  $C^1$  : (a) choisir les trois twists du patch central, qui sont libres pour le contrôle local de la forme. (b) calculer le troisième et le quatrième point de Bézier le long de chaque arête en utilisant les conditions de la  $C^1$ -continuité. (c) choisir les trois derniers points de Bézier du patch central, qui sont aussi des paramètres de forme libres. (d) calculer les trois points de Bézier restants, en utilisant les conditions de la  $C^1$ -continuité.

### 2.3) Interpolation de normales — localité stricte du schéma

La méthode de base du 4-split qui vient d'être présentée 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 en tenant compte des sommets voisins de celui-ci, comme le montre les équations (4). 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.

La généralisation que nous proposons dans [43] repose sur la remarque suivante : le voisinage d'un sommet  $p$ , intervenant dans tous les calculs autour de ce sommet, peut être remplacé par un autre ensemble de  $n$  points non nécessairement égaux aux points voisins, sans pour autant affecter la continuité  $G^1$ . La méthode devient alors strictement locale. De plus l'interpolation des vecteurs normaux en les sommets devient alors possible, ce qui offre un outil supplémentaire de design.



## Paramètres de forme

### 3.1) Nécessité des paramètres de forme

Il a été déjà évoqué dans le chapitre 1 qu'un schéma d'interpolation devrait pouvoir offrir suffisamment de degrés de liberté tout en maintenant un bas degré polynomial de l'interpolant. Ces degrés de liberté sont indispensables au design de formes lisses et agréables (plaisantes), d'où leur nom de *paramètres de forme*. Il est évident que la triangulation polygonale dont les sommets sont à interpoler est prédominante pour la détermination de la forme de la surface. Idéalement, la surface résultante devrait se comporter comme une peau lisse enveloppant la structure polygonale et arrondissant les arêtes et les sommets.

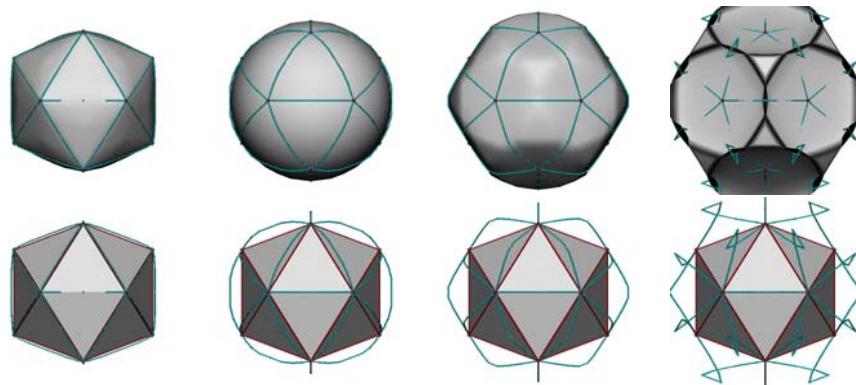


figure 9: Interpolation de l'icosaèdre pour différentes valeurs du paramètre  $\beta$ : 0.01, 0.12, 0.2, 0.4. La ligne du haut montre la surface et les polygones de contrôle des courbes. La ligne du bas montre les polygones de contrôle et la triangulation polygonale à interpoler.

L'exemple de l'icosaèdre permet une illustration simple de l'effet d'un des paramètres de forme offert par la méthode de 4-split. L'icosaèdre est montré en bas en fig. 9 ensemble avec les courbes frontières, différentes longueurs pour les dérivées premières ont été choisies. Puisque les sommets de cet icosaèdre se trouvent sur la sphère unité, une reproduction de la sphère paraît souhaitable et peut en effet être obtenue avec ce schéma pour un choix des paramètres de forme. Une erreur maximale de  $10^{-3}$  a été obtenue après quelques essais manuels des paramètres de forme.

Concernant les paramètres de forme deux difficultés majeures se posent. Premièrement, posséder un grand nombre de degré de liberté sans qu'ils aient une signification géométrique n'est pas souhaitable. Deuxièmement, quand le nombre de paramètres de forme est grand, comme dans notre schéma, un choix automatique des valeurs doit être mis à disposition. Ces deux aspects, l'interprétation géométrique et le choix automatique des paramètres de forme, ont guidé les travaux présentés dans ce chapitre.

La méthode du 4-split présentée au chapitre 2 offre un certain nombre de degrés de liberté dont l'interprétation et le choix automatique ont fait l'objet d'une publication [42] (inclue page 125 de ce mémoire). Il s'est avéré par la suite que cette méthode de base souffre de contraintes de régularité trop fortes sur les dérivées premières et secondes. Une généralisation du 4-split a donc été ensuite développée, qui relâche les contraintes sur les dérivées secondes. Une étude détaillée du problème de compatibilité du twist et de sa solution proposée pour le 4-split a rendu possible cette généralisation. Les résultats correspondants ont été publiés dans [8] (inclue page 137 de ce mémoire).

### 3.2) Généralisation du 4-split

#### Degrés de liberté sur les courbes frontière

La forme des courbes frontière des facettes composant la surface interpolante est prépondérante dans la détermination de la forme de la surface. On peut obtenir une surface localement aplatie si les courbes frontière sont presque plates. Mais si elles sont rondes, la facette correspondante sera également ronde. Ce phénomène est illustré en fig. 9. D'où l'importance d'avoir un maximum de liberté dans la construction des courbes frontière. Or, la méthode du 4-split offre en chaque sommet un seul degré de liberté vectoriel, la normale au plan tangent, et trois degrés de liberté scalaires,  $\beta, \gamma_1, \gamma_2$ , pour la construction de tous les segments de courbes autour d'un sommet commun (cf. étape (1) de l'algorithme en sect. 2.2). Le paramètre  $\beta$  contrôle la longueur des tangentes au sommet, les paramètres  $\gamma_1, \gamma_2$  agissent sur les dérivées secondes.

La construction des courbes frontière polynomiales est contrainte par le problème de compatibilité du twist qui impose en chaque sommet de sévères conditions sur les dérivées premières et secondes. Une solution particulière a été adoptée dans le 4-split du chapitre précédent. Tandis que les paramètres ( $\beta$ , vecteur normal) sur les dérivées premières sont géométriquement intuitifs, les paramètres  $\gamma_1, \gamma_2$  pour les dérivées secondes le sont moins. Ce constat nous a amené à développer une généralisation de la méthode du 4-split qui permet un libre choix de la dérivée seconde pour chaque segment de courbe.

**Solution du problème de twist revisitée**

Les courbes frontière polynomiales sont soumises aux conditions de continuité et de twist aux sommets dont nous rappelons les équations ici (voir aussi chap. 2):

$$\begin{aligned} M_C \bar{\mathbf{r}}^1 &= 0, \\ M_T \bar{\mathbf{t}} &= \Phi^1 \bar{\mathbf{r}}^1 + \Phi^0 \bar{\mathbf{r}}^2. \end{aligned}$$

On rappelle que  $\bar{\mathbf{r}}^1$ ,  $\bar{\mathbf{r}}^2$  désignent les vecteurs colonne  $n \times 3$  des dérivées premières et secondes des  $n$  segments de courbes frontière en un sommet de degré  $n$ , et  $\bar{\mathbf{t}}$  le vecteur  $n \times 3$  composé des  $n$  vecteurs twist.

Pour que le système de twist 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  $M_T$  que l'on notera  $\text{Im}(M_T)$ , et  $\bar{\mathbf{r}}^1$  doit en plus appartenir au noyau de la matrice  $M_C$  que l'on notera  $\text{Ker}(M_C)$ . La méthode de Loop ainsi que le 4-split proposent deux choix particuliers de  $\bar{\mathbf{r}}^1$  et  $\bar{\mathbf{r}}^2$  dans  $\text{Im}(M_T) \cap \text{Ker}(M_C)$  et  $\text{Im}(M_T)$  respectivement. Afin de construire  $\bar{\mathbf{r}}^1$  et  $\bar{\mathbf{r}}^2$  avec plus de libertés, il est nécessaire de caractériser les deux espaces  $\text{Im}(M_T)$  et  $\text{Ker}(M_C)$ .

**Nouveau calcul des dérivées premières**

Le fait d'avoir fixé les fonctions scalaires qui combinent les trois dérivées partielles le long des courbes frontière dans la condition  $G^1$  (Condition (C2) en sect. 1.1) entre deux facettes à des valeurs constantes  $= \frac{1}{2}$  implique que les matrices  $M_C$  et  $M_T$  ont une structure particulière. Elles sont circulantes. Par une analyse propre des deux matrices on peut démontrer que le noyau de la matrice  $M_C$  est engendré par les deux vecteurs

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

Or  $\mathbf{k}^a$  et  $\mathbf{k}^b$  appartiennent aussi à l'espace image de la matrice  $M_T$ , car il existe deux vecteurs  $\tilde{\mathbf{k}}^a$  et  $\tilde{\mathbf{k}}^b$  tels que

$$M_T \tilde{\mathbf{k}}^a = \mathbf{k}^a \quad \text{et} \quad M_T \tilde{\mathbf{k}}^b = \mathbf{k}^b,$$

où

$$\tilde{\mathbf{k}}^a = \begin{bmatrix} \vdots \\ \cos\left(\frac{2i\pi}{n}\right) + \tan\left(\frac{\pi}{n}\right) \sin\left(\frac{2i\pi}{n}\right) \\ \vdots \end{bmatrix}, \quad \tilde{\mathbf{k}}^b = \begin{bmatrix} \vdots \\ \tan\left(\frac{\pi}{n}\right) \cos\left(\frac{2i\pi}{n}\right) - \sin\left(\frac{2i\pi}{n}\right) \\ \vdots \end{bmatrix}.$$

Le vecteur  $\bar{\mathbf{r}}^1$  peut donc être écrit comme une combinaison linéaire de  $\mathbf{k}^a$  et  $\mathbf{k}^b$

$$\begin{bmatrix} \bar{\mathbf{r}}^1 \\ \vdots \\ \vdots \end{bmatrix}_{n \times 3} = \begin{bmatrix} \mathbf{k}^a \\ \vdots \\ \vdots \end{bmatrix}_{n \times 1} (\mathbf{a})_{1 \times 3} + \begin{bmatrix} \mathbf{k}^b \\ \vdots \\ \vdots \end{bmatrix}_{n \times 1} (\mathbf{b})_{1 \times 3} \quad (7)$$

où les vecteurs  $\mathbf{a}$  et  $\mathbf{b}$  peuvent être choisis arbitrairement.  $\mathbf{a}$  et  $\mathbf{b}$  sont donc des degrés de liberté pour le choix des dérivées premières autour d'un sommet.

### Interprétation géométrique de l'équation (7):

L'équation (7) peut être interprétée géométriquement pour faciliter le bon choix des vecteurs  $\mathbf{a}$  et  $\mathbf{b}$ .

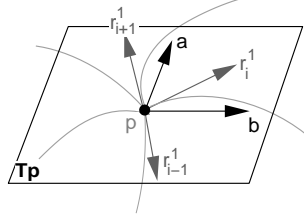


figure 10: Interprétation géométrique des 2 vecteurs libres  $\mathbf{a}$  et  $\mathbf{b}$  dans la construction des dérivées premières.

Les lignes de  $\bar{\mathbf{r}}^1$  en (7) sont les tangentes des courbes frontière autour du sommet  $\mathbf{p}$  (voir fig. 10). Elles sont coplanaires dans le plan tangent  $T_p$  en  $\mathbf{p}$ .  $\{\mathbf{a}, \mathbf{b}\}$  est une base de ce plan tangent et  $\mathbf{k}^a$  et  $\mathbf{k}^b$  sont les coordonnées des  $\mathbf{r}_i^1$  dans cette base. Ces coordonnées,  $\mathbf{k}_i^a = \cos(\frac{2\pi i}{n})$ ,  $\mathbf{k}_i^b = \sin(\frac{2\pi i}{n})$  signifient que les points de contrôle de la première dérivée forment une transformation affine d'un polygone régulier à  $n$  cotés. Cette transformation affine est explicitement donnée par le choix des vecteurs de base du plan tangent  $\mathbf{k}^a$  et  $\mathbf{k}^b$ .

#### Remarque:

Puisqu'avec cette méthode on peut choisir les vecteurs de base du plan tangent, le schéma peut donc aussi interpoler des vecteurs normaux pré-définis.

### Nouveau calcul des dérivées secondes

La deuxième condition de continuité porte sur les dérivées secondes des courbes autour d'un sommet. Elles doivent, comme les dérivées premières, être dans l'espace image de  $M_T$ . On peut démontrer que

$$\text{rang}(M_T) = \begin{cases} n & \text{si } n \text{ est impair} \\ n - 1 & \text{sinon.} \end{cases}$$

Quand  $n$  est impair, on peut donc choisir arbitrairement un vecteur  $\bar{\mathbf{r}}^2$ , i.e.  $n$  dérivées secondes  $\mathbf{r}_i^2$ , car il existe toujours un vecteur  $\tilde{\mathbf{r}}^2$  tel que

$$\bar{\mathbf{r}}^2 = M_T \tilde{\mathbf{r}}^2.$$

Quand  $n$  est pair, seules  $n - 1$  dérivées secondes  $\mathbf{r}_i^2$  peuvent être choisies arbitrairement. En pratique, et quelque soit la parité de  $n$ , il suffit de se donner un vecteur  $\tilde{\mathbf{r}}^2$  quelconque et de choisir comme dérivées secondes  $\bar{\mathbf{r}}^2 = M_T \tilde{\mathbf{r}}^2$ .

### Conclusion

La résolution du problème de compatibilité du twist par un choix de courbes frontière compatibles aux conditions  $G^1$  aux sommets a été généralisée dans ce paragraphe.

### 3.3) Choix des paramètres de forme

Dans un modeleur géométrique, les paramètres de forme jouent un rôle clef, car ils permettent de créer des formes très diverses pour une même structure de contrôle. Au paragraphe 3.1 nous avons retenu deux conditions nécessaires sur les paramètres de forme:

- avoir une interprétation géométrique intuitive,
- offrir un choix automatique de leurs valeurs (valeurs par défaut).

La satisfaction de ces deux conditions permet en même temps de répondre à la question principale:

*Est-ce que la méthode marche bien ?*

Ou plus précisément, comment peut-on obtenir la forme souhaitée à partir d'une triangulation donnée ? Le choix optimal des paramètres de forme n'est pas toujours simple et dépend beaucoup des applications. La forme de la surface finale est très sensible au "bon" choix de ces paramètres. Il s'y rajoute la contrainte que le schéma doit rester local, en particulier que les deux morceaux de courbes correspondants à une arête doivent être construits de façon indépendante.

Nous proposons deux types de procédures pour un choix des paramètres de forme:

- (1) design interactif,
- (2) choix automatique basé sur des
  - a) méthodes heuristiques,
  - b) méthodes d'optimisation,
  - c) méthodes mixtes.

Pour le design interactif un modeleur graphique 3D a été développé en collaboration avec Georges-Pierre Bonneau qui, après lecture de la triangulation polygonale, calcule une surface interpolante par défaut, et permet ensuite la manipulation de la surface ou de la triangulation grâce à des "manettes géométriques" (picking et déplacement de points, tangentes, normales,...) mises à disposition. La localité du schéma permet d'effectuer des manipulations en temps réel à l'écran. Si le nombre de sommets est trop élevé, le choix interactif des paramètres libres devient trop fastidieux. Pour tout réseau surfacique il est de toute manière indispensable de disposer de procédures de choix automatique des paramètres de forme, qui seront présentées dans la suite.

Pour déterminer les paramètres de forme, il peut être tentant d'avoir recours à des critères portant sur les courbes joignant les sommets à interpoler. En effet, la forme de ces courbes influence en grande partie la forme finale de la surface. Mais une telle méthode aboutirait à un algorithme global, puisqu'il lierait les valeurs des paramètres de forme en tous les sommets. Ainsi les deux segments des courbes frontière doivent être déterminés de façon indépendante l'un de

l'autre pour garantir la localité du schéma. Il suffit pour cela de choisir aux deux extrémités (aux sommets) les dérivées premières et secondes, la condition d'un raccord  $C^1$  fait ensuite que la courbe cubique par morceau est entièrement déterminée.

Les  $n$  segments de courbe, i.e. les  $n$  dérivées premières et secondes, autour d'un sommet sont déterminés en même temps, car il n'y a pas assez de degré de liberté pour le faire séparément. On procède en deux étapes: d'abord le choix des dérivées premières et puis celui des dérivées secondes.

### Choix heuristique des dérivées premières optimales

Deux degrés de liberté vectoriels sont à disposition pour le calcul des  $n$  dérivées premières au sommet, ce sont les vecteurs  $\mathbf{a}$  et  $\mathbf{b}$ , base du plan tangent. En pratique, il faut les déterminer de telle façon que les  $n$  tangentes des courbes frontières approximent au mieux certaines tangentes qualifiées d'optimales, notées  $\mathbf{r}_{opt}^1$ . Une règle heuristique basée sur la géométrie locale de la triangulation (celle de Piper [80]) permet en un premier temps de définir des tangentes optimales qui seront ensuite approximées au sens des moindres carrés:

$$\|\bar{\mathbf{r}}^1(\mathbf{a}, \mathbf{b}) - \mathbf{r}_{opt}^1\|^2 \rightarrow \min_{\mathbf{a}, \mathbf{b}}.$$

### Choix heuristique des dérivées secondes

Quand le degré  $n$  du sommet est pair, alors les  $n$  dérivées secondes peuvent être choisies librement, sinon seules  $n - 1$  dérivées secondes sont libres.

Une manière heuristique de procéder pour chaque courbe est d'essayer d'approcher la courbe cubique définie par interpolation des deux sommets opposés sur une arête et des deux tangentes précédemment calculées. Les courbes cubiques sont connues pour ne posséder au plus qu'un point d'inflexion, autrement dit, elles ondulent peu. Puisque les tangentes ont été estimées en fonction de la géométrie de la triangulation sous-jacente, le calcul des dérivées seconde par cette méthode se fait également en fonction de la triangulation.

### Une approche variationnelle

Un autre principe pour la détermination de paramètres de forme consiste à employer une minimisation d'énergie afin d'obtenir des courbes lisses. Le point de jonction entre les deux segments est d'abord fixé de manière heuristique (appartenance à une courbe cubique comme au paragraphe précédent par exemple). Il s'appelle *point cible*, car en réalité il ne sera qu'approché par la courbe. Il offre d'ailleurs la possibilité de contrôler intuitivement et interactivement la forme de la courbe: en déplaçant ce point la courbe le suit. La nécessité de fixer ce point est due au fait que les deux segments de courbe doivent être déterminées indépendamment afin de garantir la localité de l'algorithme.

Ensuite la dérivée seconde de chaque segment de courbe cubique peut être déterminée séparément en minimisant une fonctionnelle d'énergie détaillée plus bas.

Cette approche variationnelle a pour objet de produire des courbes esthétiques et d'éviter les ondulations. Elle est un outil classique dans le domaine du "Variational Design" en CAGD. Ne citons que les travaux les plus connus, car à la fin de ce mémoire nous y reviendrons [74, 34, 11, 7]. 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 [90, 71, 79], en version linéarisée, de la manière suivante:

$$E(\mathbf{b}_2) = \sum_{i=1}^n \left[ \int_0^{\frac{1}{2}} \|X_i''(t)\|^2 dt + \omega^2 \int_0^{\frac{1}{2}} \|X_i'(t)\|^2 dt \right],$$

où  $X_i(t)$  est une courbe cubique définie sur  $[0, \frac{1}{2}]$  par les points de contrôle  $\mathbf{b}_0 = \mathbf{p}$ ,  $\mathbf{b}_1^i$ ,  $\mathbf{b}_2^i$ ,  $\mathbf{b}_3^{i*}$ , et  $\omega$  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  $\mathbf{b}_3^{i*}$ .

Les différents principes pour la détermination des paramètres de forme optimaux que nous venons de décrire ne sont que des exemples d'approches que nous avons développés et expérimentées. Les publications [42, 97, 8] contiennent toutes un paragraphe à ce sujet.

### Lissage des macro-patches

Le remplissage des macro-patches n'a pas changé par rapport à la version de base du 4-split. Comme dans le cas des courbes, les paramètres libres optimaux peuvent être choisis de manière heuristique, variationnelle ou mixte. Il y a six points de contrôle libre pour un remplissage lisse des patches. En pratique, nous utilisons la minimisation d'énergie suivante ( $k = 2$  ou  $3$ ):

$$E_k(X) = \int_{\Delta} \left( \left( \frac{\partial^k X}{\partial u^k} \right)^2 + \left( \frac{\partial^k X}{\partial v^k} \right)^2 + \left( \frac{\partial^k X}{\partial w^k} \right)^2 \right) dudvdw.$$

Bien que pour certaines triangulations des règles heuristiques donnent parfois de meilleurs résultats, l'approche variationnelle est une méthode permettant d'assurer à interpoler l'interpolation par des macro-facettes lisses, quelque soit la triangulation.

---

**Chapitre 4****Interpolation lisse généralisée****4.1) Motivation**

La méthode de base de 4-split présentée au chapitre 2 reposait sur un choix particulier des dérivées premières et secondes autour des sommets de la triangulation: celles-ci se situent, à une transformation affine près, sur les sommets d'un polygone régulier. La généralisation présentée dans le précédent chapitre a permis de relâcher cette contrainte pour les dérivées secondes. Les degrés de liberté ainsi requis ont été utilisés pour obtenir des formes plus lisses. Néanmoins les dérivées premières doivent encore vérifier les contraintes de régularité. Or les dérivées premières influent fortement l'allure des courbes polynomiales. Ainsi si les arêtes de la triangulation sont disposées plutôt régulièrement autour de chaque sommet, alors une disposition régulière des dérivées premières conviendra, i.e. produira des courbes polynomiales frontière dont l'allure suit naturellement, tout en les lissant bien sur, les arêtes de la triangulation. Mais lorsque la triangulation est irrégulière, la contrainte de régularité des dérivées premières peut engendrer des ondulations inacceptables.

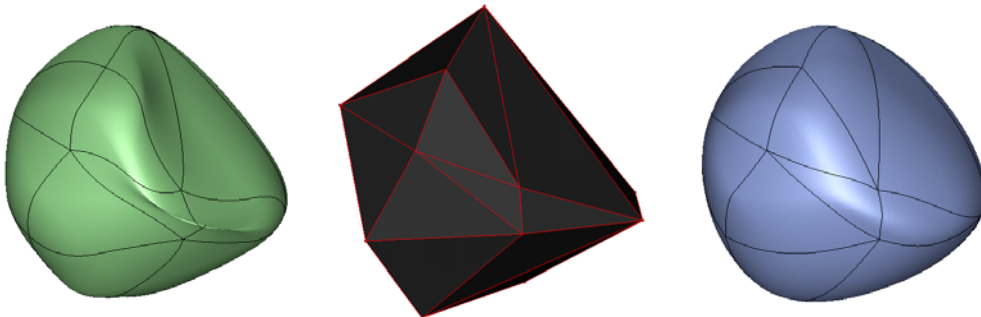


figure 11: Au milieu: triangulation irrégulière. A gauche: interpolation présenté au chapitre 3. A droite: interpolation par la méthode généralisée présentée dans ce chapitre.

Ce phénomène est illustré en fig. 11. La triangulation polygonale se trouve au milieu. La partie gauche montre une surface obtenue par la méthode décrite dans le chapitre précédent. Deux types de problèmes sont visibles. Lorsque les arêtes incidentes à un sommet sont à peu près de même longueur, mais non-



régulièrement réparties autour du sommet, alors des ondulations des courbes frontière apparaissent tangentiellement à la surface. Lorsque des arêtes courtes et d'autres longues sont incidentes à un même sommet, alors des ondulations orthogonales à la surface voient le jour. La partie droite de la fig. 11 montre le résultat obtenu avec la méthode qui est introduite dans ce chapitre, permettant de relâcher la contrainte de régularité des dérivées premières. Les ondulations disparaissent clairement.

Cette section 4.1 a expliqué et illustré en quoi la contrainte de régularité portant sur les dérivées premières peut conduire à des ondulations indésirables des surfaces interpolantes. Les deux prochaines sections montrent comment cette contrainte imposée dans les méthodes introduites aux chapitres 2 et 3 peut être relâchée. L'algorithme résultant diffère des précédentes dans les deux phases de construction des courbes frontière et des tangentes trans-frontière. Par contre, la phase de remplissage de l'intérieur des macro-patches est inchangée. La section suivante 4.2 détaillera la résolution de la compatibilité du twist et la section 4.3 décrira brièvement la construction des courbes frontière et des dérivées trans-frontière.

## 4.2) Compatibilité du twist

Pour pouvoir relâcher la contrainte de régularité des dérivées premières, il est nécessaire de revenir à la source des méthodes introduites précédemment. La contrainte de régularité est en effet une conséquence de la simplification choisie pour les fonctions scalaires  $\mu$  et  $\nu$  intervenant dans la condition (C2) de continuité  $G^1$ , voir section 2.1. Ces fonctions ont été choisies dans les méthodes précédentes constantes, égales à  $\frac{1}{2}$ . Ce choix impliquait lui-même la valeur de la fonction  $\Phi$  en 0:  $\Phi(0) = \cos(\frac{2\pi}{n})$ , et par suite la disposition régulière des dérivées premières autour de chaque sommet. Il est donc nécessaire de ne plus supposer les fonctions  $\mu$  et  $\nu$  constantes. Dès lors le problème de compatibilité du twist prend une forme différente que dans les méthodes précédentes.

En évaluant en un sommet, i.e. en 0, la condition générale (C2) de continuité  $G^1$ , il vient:

$$(C2^0) \quad \Phi_i(0)S_{u_i}^i(0,0) = \nu_i(0)S_{u_{i+1}}^i(0,0) + \mu_i(0)S_{u_{i-1}}^{i-1}(0,0).$$

On rappelle que  $i$  est l'indice tournant autour d'un sommet  $p$  de la triangulation. L'indice  $i$  a été rajouté aux fonctions scalaires  $\Phi_i, \mu_i, \nu_i$ , traduisant le fait que contrairement aux méthodes précédentes, les valeurs de ces fonctions autour d'un sommet ne seront en général pas égales.

Dans les méthodes précédentes, la démarche suivie a été de choisir les valeurs des fonctions scalaires, et d'en déduire les valeurs des dérivées des courbes frontière. Maintenant la démarche opposée est adoptée: on suppose fixées les dérivées premières  $S_{u_i}^i(0,0)$  à des valeurs quelconques, et on cherche les valeurs correspondantes des fonctions scalaires. Bien entendu, les dérivées

premières doivent être choisies dans un même plan, le plan tangent, puisque la surface recherchée est  $G^1$ . Ainsi les vecteurs  $S_{u_{i-1}}^{i-1}(0,0)$ ,  $S_{u_i}^i(0,0)$ ,  $S_{u_{i+1}}^i(0,0)$  sont coplanaires, et la relation ( $C2^0$ ) traduit précisément cette coplanarité.

Pour simplifier les équations suivantes, on réutilise les notations  $\mathbf{r}_i^1 := S_{u_i}^i(0,0)$  pour les dérivées premières,  $\mathbf{r}_i^2 := S_{u_i u_i}^i(0,0)$  pour les dérivées secondes et  $\mathbf{t}_i := S_{u_i u_{i+1}}^i(0,0)$  pour les twists. L'évaluation de deux produits vectoriels par  $\mathbf{r}_{i-1}^1$  et  $\mathbf{r}_i^1$  des membres gauche et droit de ( $C2^0$ ), suivie par celle des produits scalaires avec un vecteur normal  $\mathbf{n}$  au plan tangent, fournit les conditions nécessaires et suffisantes sur les valeurs  $\mu_i(0)$ ,  $\nu_i(0)$ ,  $\Phi_i(0)$ :

$$\mu_i(0) = \frac{|\mathbf{r}_i^1, \mathbf{r}_{i+1}^1, \mathbf{n}|}{|\mathbf{r}_{i-1}^1, \mathbf{r}_{i+1}^1, \mathbf{n}|} \Phi_i(0), \quad \nu_i(0) = \frac{|\mathbf{r}_{i-1}^1, \mathbf{r}_i^1, \mathbf{n}|}{|\mathbf{r}_{i-1}^1, \mathbf{r}_{i+1}^1, \mathbf{n}|} \Phi_i(0). \quad (8)$$

L'interprétation géométrique de ces valeurs est illustrée en fig. 12. Le degré de liberté  $\Phi_i(0)$  se dégageant de ces conditions provient de la coplanarité des trois vecteurs  $\mathbf{r}_{i-1}^1$ ,  $\mathbf{r}_i^1$ ,  $\mathbf{r}_{i+1}^1$ .

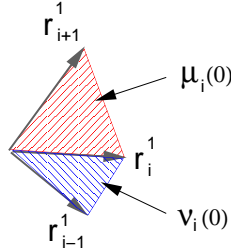


figure 12: Interprétation géométrique des fonctions scalaires: Les valeurs  $\mu_i(0)$ ,  $\nu_i(0)$  sont proportionnelles aux aires des triangles.

Le problème de compatibilité du twist apparaît en dérivant la condition ( $C2^0$ ) en 0:

$$\Phi_i(0)\mathbf{r}_i^2 = \mu_i'(0)\mathbf{r}_{i-1}^1 - \Phi_i'(0)\mathbf{r}_i^1 + \nu_i'(0)\mathbf{r}_{i+1}^1 + \mu_i(0)\mathbf{t}_{i-1} + \nu_i(0)\mathbf{t}_i. \quad (9)$$

Dans les méthodes précédentes, les termes  $\nu_i'(0)$  et  $\mu_i'(0)$  s'annulaient puisque  $\nu_i$  et  $\mu_i$  étaient constantes. Et la compatibilité du twist était résolue en imposant un choix particulier des dérivées premières et secondes. Dans la nouvelle méthode, la compatibilité du twist peut être résolue en choisissant des valeurs quelconques pour les dérivées premières  $\mathbf{r}_i$  et pour les twists  $\mathbf{t}_i$ , et en déduisant de (9) la valeur de dérivées secondes  $\mathbf{r}_i^2$ . Notons qu'aucun système linéaire n'est à résoudre, puisque la dérivée seconde  $\mathbf{r}_i^2$  n'apparaît que dans une seule des équations (9). Si cette résolution est plus simple dans ce sens, elle est par contre moins intuitive que dans les méthodes précédentes dans la mesure où les twists ont une signification géométrique moins directe que celles des dérivées secondes, qui étaient auparavant les paramètres libres. Il n'en reste pas moins possible de choisir des dérivées secondes optimales, et de déterminer par la suite des twists de telle manière à ce que les dérivées secondes réelles imposées par (9) approximent aux moindres carrés les dérivées secondes optimales choisies.

### 4.3) Construction des courbes frontière et des tangentes trans-frontière

La résolution de la compatibilité du twist présentée dans la section précédente 4.2 fournit les informations suivantes en chaque sommet:

- \* les dérivées premières et secondes  $r_i^1, r_i^2$
- \* les twists  $t_i$
- \* les valeurs en zéro des fonctions scalaires  $\Phi_i, \nu_i, \mu_i$ .

Dans cette section, la construction des courbes frontière et des tangentes trans-frontière compatibles avec ces informations est décrite brièvement. Les détails ont été publiés dans [44] (inclus page 147 dans ce mémoire).

L'écriture symétrique de la condition de continuité  $G^1$ , i.e. l'analogue de la condition (6) sans la contrainte de régularité, prend la forme suivante:

$$\begin{aligned} 2 \mu_i(u_i) S_{u_{i+1}}^i(u_i, 0) &= \Phi_i(u_i) S_{u_i}^i(u_i, 0) + \mathbf{V}_i(u_i) \\ 2 \nu_i(u_i) S_{u_{i-1}}^{i-1}(0, u_i) &= \Phi_i(u_i) S_{u_i}^i(u_i, 0) - \mathbf{V}_i(u_i). \end{aligned} \quad (10)$$

Dans les algorithmes précédents, le système (10) était résolu en choisissant la courbe frontière  $S^i$   $C^1$  cubique par morceau interpolant les dérivées premières et secondes, et en déterminant les dérivées trans-frontière vérifiant (10) et interpolant les dérivées premières et les twists. Dans le nouveau cadre généralisé, cette démarche n'est plus possible, car elle aboutirait à des dérivées trans-frontière  $S_{u_{i+1}}^i, S_{u_{i-1}}^{i-1}$ , rationnelles, puisque les fonctions  $\mu_i(u_i), \nu_i(u_i)$  ne sont plus constantes, mais linéaires, dans (10). Pour assurer qu'une solution polynomiale au système (10) existe, la seule manière de procéder est de faire en sorte que le produit  $\mu_i(u_i)\nu_i(u_i)$  apparaisse en facteur dans les deux fonctions vectorielles  $S_{u_i}^i(u_i, 0)$  et  $\mathbf{V}_i(u_i)$ . Ainsi une simplification peut s'effectuer dans les membres gauche et droit de (10), et une solution polynomiale peut être déterminée.

#### Courbes frontière

Pour que le produit  $\mu_i(u_i)\nu_i(u_i)$  apparaisse en facteur dans la dérivée  $S_{u_i}^i$  le long de la courbe frontière, tout en assurant une interpolation des dérivées premières et secondes aux sommets, ainsi qu'un raccord  $C^1$  au point de jonction entre les deux morceaux de courbe, il faut choisir des **courbes frontière de degré 5 par morceau**. Un rapide calcul du nombre de contraintes en présence permet de s'en convaincre, les détails étant fournis dans l'article [44]:

$2 \times 3 = 6$	interpolation des positions, dérivées premières et secondes en
	chaque extrémité,
$2 \times 2 = 4$	mise en facteur de $\mu_i(u_i)\nu_i(u_i)$ pour chaque morceau de courbe,
	$2$ conditions de continuité $C^0, C^1$ ,
<hr style="width: 10%; margin-left: 0;"/>	
total: $12$	qui correspond bien au nombre de points de contrôle disponibles

dans deux morceaux de courbes de degré 5.

### Dérivées trans-frontière

Il faut d'abord vérifier que la fonction vectorielle  $V_i(u_i)$  peut être choisie de bas degré, avec le produit  $\mu_i(u_i)\nu_i(u_i)$  en facteur. Les contraintes portant sur  $V_i(u_i)$  proviennent des valeurs des dérivées premières et secondes et des twists déjà fixées en les sommets. Il en suit que la position et la dérivée de  $V_i$  sont imposées en les deux extrémités de la courbe, i.e. pour  $u_i = 0$  et 1. Comme de plus  $V_i$  doit admettre le produit  $\mu_i(u_i)\nu_i(u_i)$  en facteur,  $V_i$  peut être choisi de la forme  $\mu_i(u_i)\nu_i(u_i)W_i$ , où  $W_i$  est une fonction de degré 2 par morceau. Ce degré est suffisant pour interpoler la position et la dérivée de  $V_i$  en les extrémités, et pour assurer une continuité  $C^0$  de  $V_i$  en  $1/2$ , i.e. au point de jonction entre les deux morceaux de courbe. La continuité de  $V_i$  est en effet imposée par la continuité des dérivées trans-frontière.

Une fois ce calcul de  $V_i$  effectué, et les courbes frontière de degré 5 par morceau calculées, alors les dérivées trans-frontière  $S_{u_{i+1}}^i(u_i, 0)$ ,  $S_{u_{i-1}}^{i-1}(0, u_i)$  peuvent être déterminées de degré 4 par morceau. En effet,  $S_{u_i}^i(u_i, 0)$  est de degré 4,  $V_i$  est de degré 4, et la simplification par  $\mu_i(u_i)$  ou  $\nu_i(u_i)$  dans (10) aboutit à un membre de droite de degré 4 donc à des dérivées trans-frontière  $S_{u_{i+1}}^i(u_i, 0)$ ,  $S_{u_{i-1}}^{i-1}(0, u_i)$  de degré 4, et finalement à des patches de degré 5 comme dans les méthodes précédentes.

### 4.4) Exemples

Cette méthode se distingue de celles précédemment développées, par un gain en flexibilité dans la construction des courbes frontières. C'est ce que nous allons illustrer dans cette section.

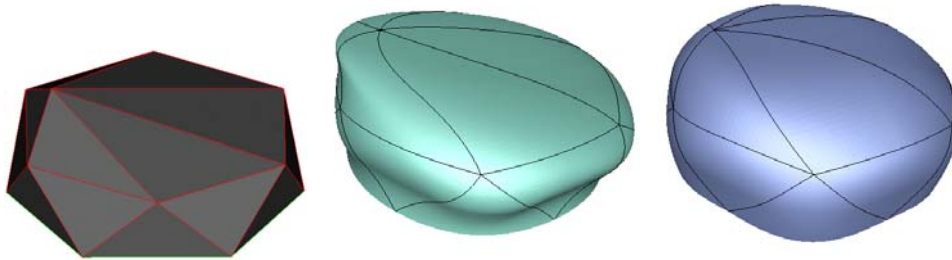


figure 13: gauche: triangulation ouverte avec sommet déplacé, milieu: surface interpolante obtenu avec la méthode optimisée du chap. 4, droite: surface interpolante avec dérivées premières arbitraires.

Un premier exemple comparatif montre l'importance d'un choix arbitraire des premières dérivées en les sommets. A gauche en fig. 14 on voit une simple triangulation ouverte dont le sommet central supérieur a été déplacé. Les angles entre les arêtes incidentes en ce sommet varient beaucoup, et les longueurs des arêtes sont également fortement différentes. C'est typiquement ce que nous appelons une triangulation irrégulière. Les courbes frontières devraient refléter ce comportement. La surface au milieu a été obtenue par l'ancienne méthode. Bien qu'elle soit lisse, les courbes frontières ondulent car ni les

dérivées premières ni les longueurs des courbes ne peuvent être adaptées à ce type de triangulation. A droite en fig. 15 la surface interpolante obtenue par la nouvelle méthode illustre l'effet positif du gain en flexibilité de cette méthode d'interpolation dite "généralisée". Figure 11 en est également une illustration.

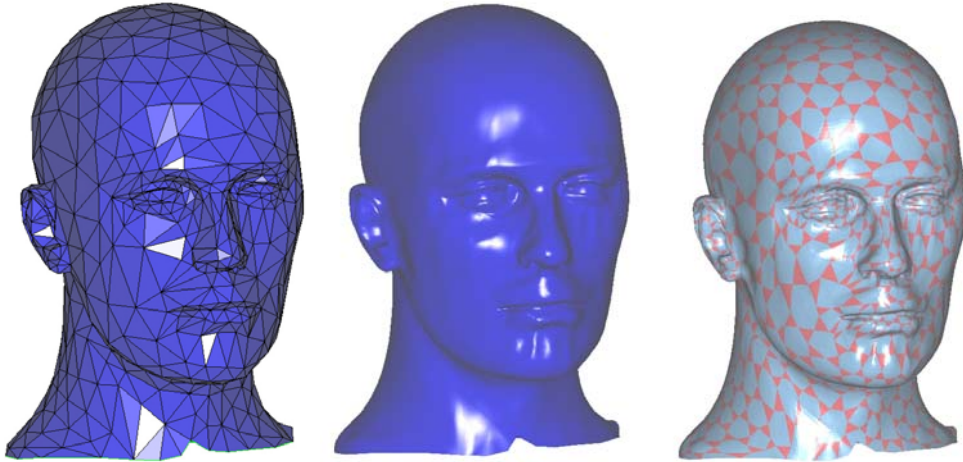


figure 16: Mannequin. gauche: triangulation en entrée, milieu: surface interpolante  $G^1$ , droite: polygone de contrôle.

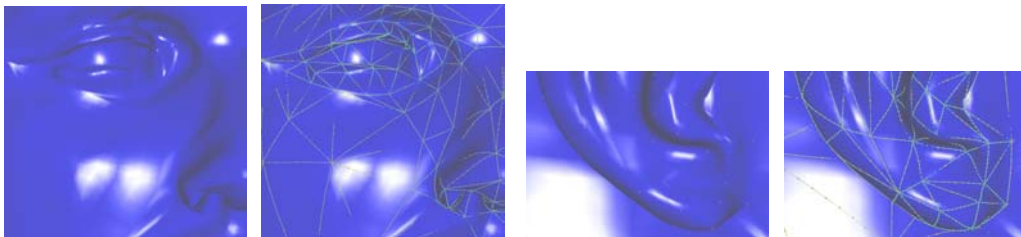


figure 17: ZOOMS sur le mannequin, avec et sans polygone de contrôle des courbes frontière.

Les figures 16 et 17 montrent le résultat de ce schéma d'interpolation généralisé sur une triangulation très complexe (mannequin data set, courtesy University of Washington). De gauche à droite en fig. 16 sont illustrés la triangulation polygonale, la surface interpolante et le réseau de contrôle de Bézier de la surface. A droite, les macro-patches sont visibles, leur patch central est coloré en rouge.

Fig. 17 montre deux détails de la surface interpolant le mannequin, à chaque fois sans et avec le polygone de contrôle des courbes frontière. Il se confirme la théorie que les courbes frontière arrivent à bien suivre les arêtes de la triangulation. Cela est possible grâce au choix libre de toutes les dérivées premières en les sommets.

---

## Chapitre 5

# Design hiérarchique

Les méthodes décrites dans les chapitres précédents permettent d'interpoler une triangulation polygonale quelconque par une surface lisse. Il est ainsi possible de modéliser aisément des surfaces lisses de topologie quelconque. La localité des algorithmes présentés permet par ailleurs de traiter des triangulations dont la taille n'est limitée que par la capacité de stockage. L'édition d'un paramètre quelconque de la surface - position d'un sommet interpolé, normale en ce sommet, direction d'une dérivée première, ... - se fait interactivement quelque soit la taille de la triangulation interpolée.

Cependant plusieurs raisons conduisent à envisager la hiérarchisation des algorithmes d'interpolation précédemment introduits.

- Une modification sur une large échelle d'une surface interpolant une triangulation complexe n'est pas possible en pratique: même si l'édition d'un paramètre se fait interactivement, une modification globale nécessiterait l'édition un à un d'un très grand nombre de paramètres. La hiérarchisation permet au contraire très facilement l'obtention de changements globaux, grâce à l'édition d'un petit nombre de paramètres dans les étages supérieurs de la hiérarchie.
- Les surfaces réelles présentent fréquemment des complexités locales, i.e. sont constituées en majorité de formes simples, mais font apparaître en des endroits précis des variations d'une finesse plus élevée. Il est intéressant pour ce type de surfaces de pouvoir hiérarchiser localement le modèle. C'est le cas par exemple pour la représentation sur un visage, de congés arrondis raccordant des pièces de carrosserie, ou d'aspérités locales sur un organe humain.
- Le rendu, ou visualisation, d'une surface complexe peut être accéléré par une modélisation hiérarchique. Plus l'objet est loin de l'observateur, moins de détails sont nécessaires pour une même précision à l'écran. A l'opposé, lorsqu'un observateur est très proche d'une surface, il n'en visualise qu'une petite partie, et il est donc intéressant de ne transmettre au rendu que la partie visible, éventuellement localement plus ou moins détaillée suivant la

distance de l'observateur. Là encore une modélisation hiérarchique capable de raffinements locaux est profitable.

### 5.1) Hiérarchisation de surfaces linéaires par morceau

Avant de décrire en section 5.2 notre modèle hiérarchique de surfaces lisses, nous commençons par en présenter les fondements dans le cas plus simple des surfaces linéaires par morceau. La section prochaine 5.2 décrira uniquement les changements nécessaires à leurs adaptation à l'interpolant surfacique  $G^1$  introduit dans le chapitre précédent.

#### Structures de données

##### *Information topologique*

La hiérarchisation choisie est une méthode de raffinement local fonctionnant par insertion de sommets. Un nouveau sommet est inséré au milieu d'une arête déjà existante, les deux faces voisines - dites faces parentes - étant subdivisées en quatre faces filles en reliant le nouveau sommet au milieu des quatre autres arêtes des faces voisines. Ce schéma d'insertion est illustré en fig. 18.

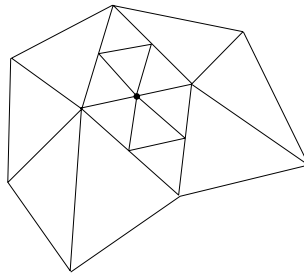


figure 18: Insertion d'un sommet.

Deux types de sommets sont alors créés: des sommets dites *éditables*, et d'autres dits *non-éditables*. En effet, un sommet dont une face parente est subdivisée mais pas l'autre ne peut pas être déplacé car il se trouve par construction au milieu de l'arête de la face non-subdivisée. Ainsi, si l'on subdivise une seule fois une seule arête, on crée un sommet éditable et quatre sommets non-éditables, comme illustré en fig. 18. Lors d'une prochaine subdivision d'une face voisine, un sommet non-éritable peut devenir éritable.

La structure de données adoptée pour coder cette hiérarchie est double, reflétant ainsi la hiérarchie naturelle des niveaux élevés vers les niveaux profonds pour les faces d'une part, et d'autre part celle des niveaux bas vers les niveaux supérieurs pour les sommets insérés. Il est en effet naturel de faire pointer une face vers ses quatre faces filles d'une part et d'autre part de faire pointer un sommet inséré vers ces sommets parents, i.e. les sommets de l'arête qu'il subdivise. En fait pour des raisons d'efficacité d'implémentation, un sommet inséré est codé en le faisant pointer vers les deux faces adjacentes à l'arête qu'il subdivise, plutôt que vers les sommets de cette arête. Plus précisément,

l'information topologique et hiérarchique est entièrement codée grâce aux deux structures de données suivantes:

#### Structure face

```
typedef struct s_Face
Vertex *cornerVertex[3];
    /* pointers to the three corner vertices */
Vertex *edgeVertex[3];
    /* pointers to the three middle edge vertices */
struct s_Face *childF[4];
    /* pointers to the four sub-faces */
Face;
```

#### Structure arête

```
typedef struct s_Vertex
struct s_Face *parentF[2];    /* pointers to the parent faces */
Vertex;
```

#### Information géométrique

L'information géométrique est constituée dans le cas des surfaces linéaires par morceau, uniquement par les positions des sommets dans l'espace affine. Cette position pourrait être tout simplement stockée par les coordonnées affines des sommets relativement à un repère absolu. Mais pour les applications d'édition interactive qui nous intéressent, il est profitable de retenir les coordonnées affines relativement à un repère dépendant des informations géométriques situées au niveau directement supérieur à celui du sommet inséré. Cette technique a été introduite dans le cadre des surfaces produit tensoriel par Forsey et Bartels [29]. Ainsi lorsqu'une modification d'un sommet situé à un niveau élevé a lieu, alors cette modification entraîne en cascade un changement de toutes les positions des sommets situés dans la sous-hiérarchie, de telle manière à ce que le mouvement global de la surface induit par le changement d'un seul sommet conserve les *variations relatives* de forme. Ce comportement est illustré par une simple ligne polygonale en fig. 19. Le changement de position d'un sommet situé en haut de la hiérarchie obtenue par subdivision des arêtes en deux, introduit un changement global "naturel" de la forme polygonale, en conservant les variations locales.

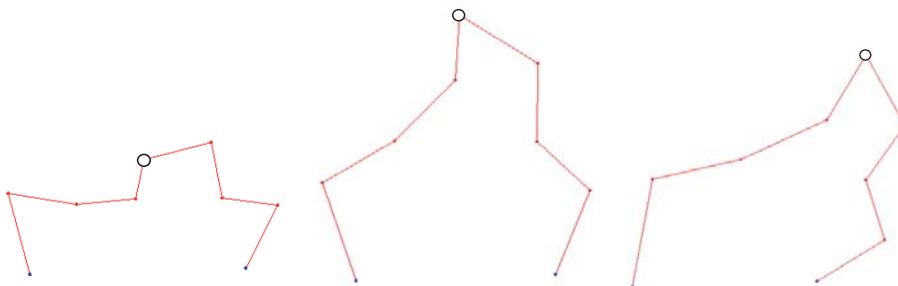


figure 19: Variations relatives. o: point édité, les autres suivent automatiquement.



Le repère local attaché à un sommet est choisi de telle manière à conserver les variations normales et tangentielles à la surface. L'origine en est le milieu de l'arête que le nouveau sommet vient subdiviser. Son axe normal est dirigé selon la moyenne des deux normales des faces voisines de l'arête subdivisée. Un deuxième axe est dirigé selon l'arête subdivisée. Enfin le troisième axe est déterminé par orthonormalisation, et en respectant l'orientation des triangles. Ce repère local est illustré en fig. 20.

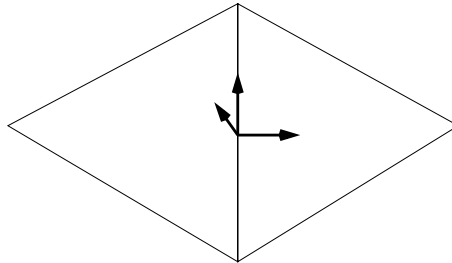


figure 20: Repère local.

Dans un souci d'efficacité, cette base est stockée dans la structure de données codant les sommets, bien qu'il soit possible simplement de la réévaluer en utilisant les pointeurs vers les deux faces parentes, déjà stockées dans la structure. Par ailleurs un drapeau est utilisé pour cocher si le sommet inséré se situe au bord de la surface. Au final, la structure de données codant les sommets est la suivante:

structure de données pour les sommets.

```
typedef struct s_Vertex
short int Border;    /* flag : the vertex is a border vertex */
double W[SPACEDIM]; /* local coordinates of the vertex */
double LF[4*SPACEDIM]; /* origin & axes of the local frame */
struct s_Face *parentF[2]; /* pointers to the parent faces */
Vertex;
```

### Exemples de modèles hiérarchiques

La figure 21 illustre un exemple simple de raffinement, d'édition à un niveau fin, puis d'édition à un niveau grossier d'un icosaèdre. Les variations locales de forme sont conservées lors de l'édition du sommet dans le niveau élevé de la hiérarchisation.

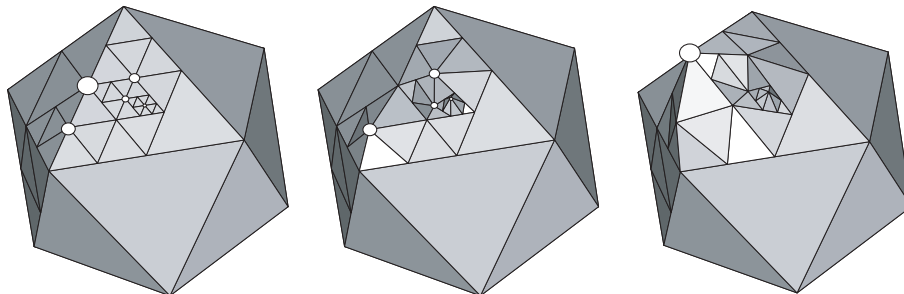


figure 21: Edition hiérarchique de l'icosaèdre. Les cercles remplis indiquent les sommets éditables. Le diamètre est proportionnel à la hauteur du sommet dans la hiérarchie.

La figure 22 montre un exemple plus complexe obtenue à partir de données de terrain sur une grille produit tensoriel, représentant une partie du Grand Canyon. La hiérarchie a été produite par un algorithme de raffinement, en partant d'une triangulation de base constituée de deux triangles couvrant le domaine carré, et en subdivisant les arêtes pour lesquelles l'une au moins des deux faces voisines a une distance maximum verticale aux données supérieure à une valeur fixée. Les sommets insérés sont déplacés exactement sur la surface interpolant bilinéairement les données irrégulières. Le bas de la figure illustre un mouvement global du modèle hiérarchique, obtenu en ne modifiant que deux sommets situés dans les niveaux élevés de la hiérarchie.

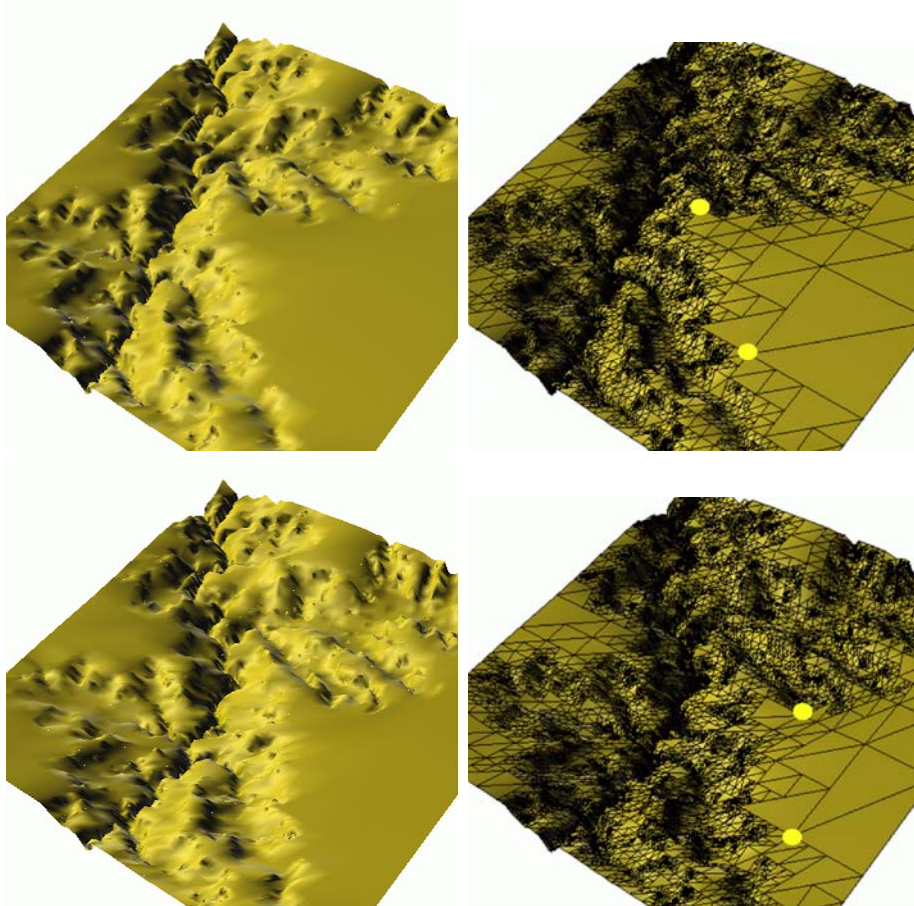


figure 22: Edition hiérarchique Canyon. En haut l'original, en bas après changement des deux sommets encerclés.

## 5.2) Modèle hiérarchique de surfaces lisses

La section précédente a présenté une structure de données modélisant des triangulations polygonales hiérarchiques. L'idée de base pour hiérarchiser une triangulation a été d'insérer un sommet dit éditable au milieu d'une arête, et de subdiviser régulièrement les deux faces parentes à cette arête. De cette manière, la position du sommet peut être modifiée, tout en maintenant la continuité  $C^0$  de l'interpolant linéaire. Une étape d'édition de ce modèle peut se

décomposer en deux actions élémentaires de nature distincte: l'une est uniquement topologique, et l'autre est géométrique.

- *Insertion d'un nouveau sommet par subdivision de l'arête et de ses deux faces voisines.*

Cette action ne modifie pas la géométrie de l'interpolant. Le nouveau sommet inséré se trouve en effet au milieu de l'arête subdivisée, soit précisément sur l'interpolant linéaire. De même la subdivision régulière des deux faces voisines ne fait que découper l'interpolant, elle ne le modifie pas.

- *Déplacement géométrique du sommet inséré.*

Cette action contrairement à la précédente ne change pas la topologie du modèle, mais agit uniquement sur les informations géométriques du nouvel élément inséré, en l'occurrence les coordonnées affines du nouveau sommet.

Dans le développement d'un modèle hiérarchique de surfaces lisses, ces deux actions doivent exister. Cela signifie en particulier pour la première action que l'interpolant lisse doit pouvoir être subdivisé sans en modifier la géométrie. Si cette action de découpage topologique est triviale pour l'interpolant linéaire, il n'en est pas de même pour l'interpolant  $G^1$ . Pour la deuxième action cela signifie que l'on doit pouvoir changer les informations géométriques attachées aux nouvelles entités insérées, et recalculer localement l'interpolant  $G^1$ . De plus, si l'utilisateur fournit de nouveau les informations géométriques correspondantes à l'interpolant de niveau élevé, il faut que le calcul local redonne cet interpolant.

Autrement dit, comme il avait été énoncé dans les motivations, sect. 1.3, il faut que l'interpolant soit **invariant par subdivision**. Cette invariance par subdivision est illustrée en fig. 23 pour une courbe. A gauche se trouve l'interpolant de niveau élevé (fig. 23.a). Pour subdiviser l'arête de droite, on commence par évaluer les informations géométriques de l'interpolant lisse pour le point dont le paramètre est situé au milieu de cette arête (fig. 23.b). Dans cet exemple fictif ces informations géométriques sont constituées par la position du sommet, et la tangente en ce sommet. Ensuite on recalcule localement à l'arête de droite l'interpolant, en utilisant ces informations, ainsi que celles attachées aux deux sommets voisins. L'interpolant est dit invariant par subdivision si ce calcul local à un niveau plus fin donne le même résultat que l'interpolant initial (fig. 23.c).

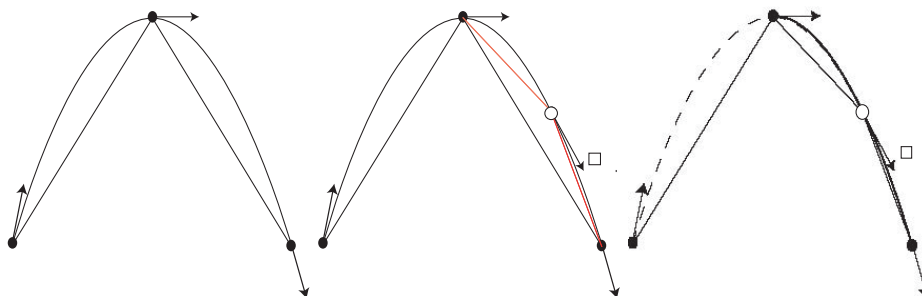


figure 23: Invariance par subdivision.

En ce qui concerne l'interpolant développé dans le chapitre 4, les informations géométriques à prendre en compte sont de deux types:

- informations attachées aux sommets:
  - position du sommet
  - $n$  dérivées premières, où  $n$  est le degré du sommet,
  - $n$  twists,
- informations attachées aux faces:
  - 6 points de contrôle libres par macro-patch (on rappelle qu'à chaque face de la triangulation polygonale correspond un macro-patch de l'interpolant lisse.

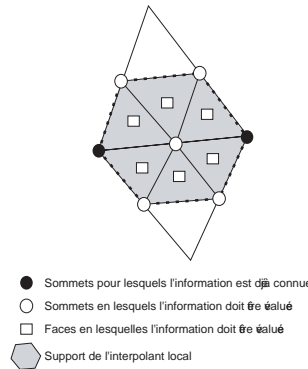


figure 24: Informations géométriques pour l'interpolant local.

La figure 24 illustre les différentes informations géométriques entrant en jeu dans le recalcul local de l'interpolant.

Puisque l'interpolant lisse du chapitre 4 est strictement local, il est effectivement possible de recalculer cet interpolant autour du sommet inséré, en utilisant les informations géométriques illustrées en fig. 24. Le recalcul local ne modifiant pas les deux premières rangées de points de contrôle le long de la frontière du support (ligne hachée en fig. 24), on est assuré d'obtenir à nouveau une surface  $G^1$ . La difficulté est de vérifier que la nouvelle surface  $G^1$  obtenue est la même que précédemment. Pour être sûr de réobtenir cette surface, il faut et il suffit de fixer tous les paramètres libre du schéma d'interpolation local en conformité avec cette surface. En effet, fixer tous les paramètres libres implique l'unicité de l'interpolant local, et le choix conforme de ces paramètres libres signifie que la précédente surface  $G^1$  est un interpolant local pour ces paramètres. Ces deux constats impliquent bien que l'interpolation locale redonne la surface  $G^1$  existante, i.e. que le schéma d'interpolation du chapitre 4 est invariant par subdivision.

Il reste donc à voir comment fixer tous les paramètres libres du schéma d'interpolation du chapitre 4 en conformité avec la surface  $G^1$  existante. Outre les informations géométriques énumérées en fig. 24, ces paramètres libres comprennent:

- \* Les valeurs des fonctions scalaires  $\Phi, \mu, \nu$  dans l'équation (C2) de continuité  $G^1$ .

\* Les valeurs de la fonction vectorielle  $V$  dans les équations symétriques (10) de continuité  $G^1$ .

Si les fonctions  $\Phi$ ,  $\mu$ ,  $\nu$  et  $V$  sont fixées aux extrémités des arêtes en raison de l'interpolation des dérivées premières et des twists, leurs valeurs à l'intérieur de l'arête n'étaient contrainte précédemment que par l'objectif d'obtenir des patches de degré minimum. En ce qui concerne les fonctions scalaires  $\Phi$ ,  $\mu$ ,  $\nu$  cette contrainte a conduit à choisir des fonctions linéaires en un seul morceau sur l'ensemble de l'intervalle de définition, ce qui est en conformité avec la surface existante qui est définie sur le niveau supérieur de la hiérarchie. Par contre, la fonction  $V$  a été choisie de la forme  $V = \nu\mu W$ , avec  $W$  de degré 2 par morceau afin de pouvoir interpoler les positions et la dérivée aux extrémités, et d'assurer une continuité  $C^0$  au milieu de l'arête. Ce choix n'est pas conforme avec la surface existante, il faut en effet recourir à une fonction  $W$  définie sur un seul morceau. Pour cela, on peut choisir  $W$  en un seul morceau cubique, comme illustré en fig. 25.

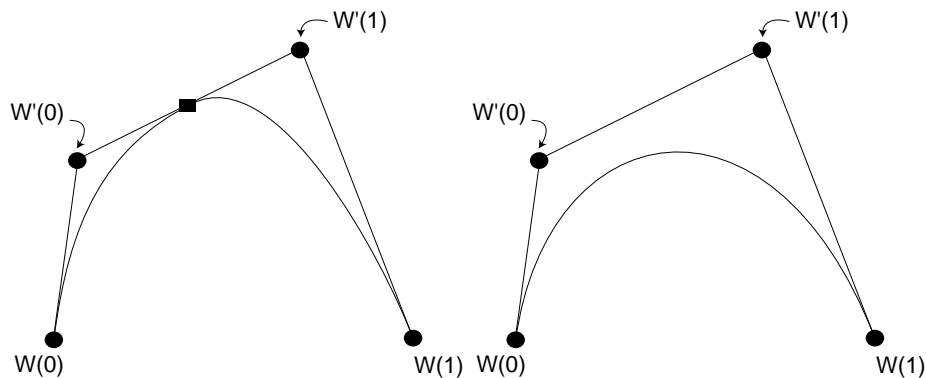


figure 25: Ancien et nouveau choix de  $W$ .

Ce choix de  $W$  implique une valeur de  $V$  en conformité avec la surface  $G^1$  existante et assure ainsi un interpolant invariant par subdivision.

### 5.3 Exemples

En fig. 26 nous illustrons l'invariance par subdivision du schéma d'interpolation sur l'exemple du cube. Les quatre figures à gauche du haut vers le bas représentent la triangulation initiale, la surface lisse interpolante, la surface avec le réseau des points de contrôle, et uniquement le polygone de contrôle. Dans chaque macro-patch, le patch de Bézier central est en rouge, et les trois patches de Bézier externes sont en bleu. Les quatre figures de droite représentent les mêmes éléments, après subdivision d'une arête. Le réseau des points de contrôle se densifie, par contre la surface ne change pas, illustrant ainsi l'invariance par subdivision.

La figure 27 représente une surface définie sur trois niveaux de subdivision. Le niveau supérieur de subdivision est la surface cubique déjà illustrée dans la colonne de gauche de la figure 26. Le deuxième niveau de subdivision permet de modéliser les pattes, le cou et la tête. Le troisième niveau modélise la queue.

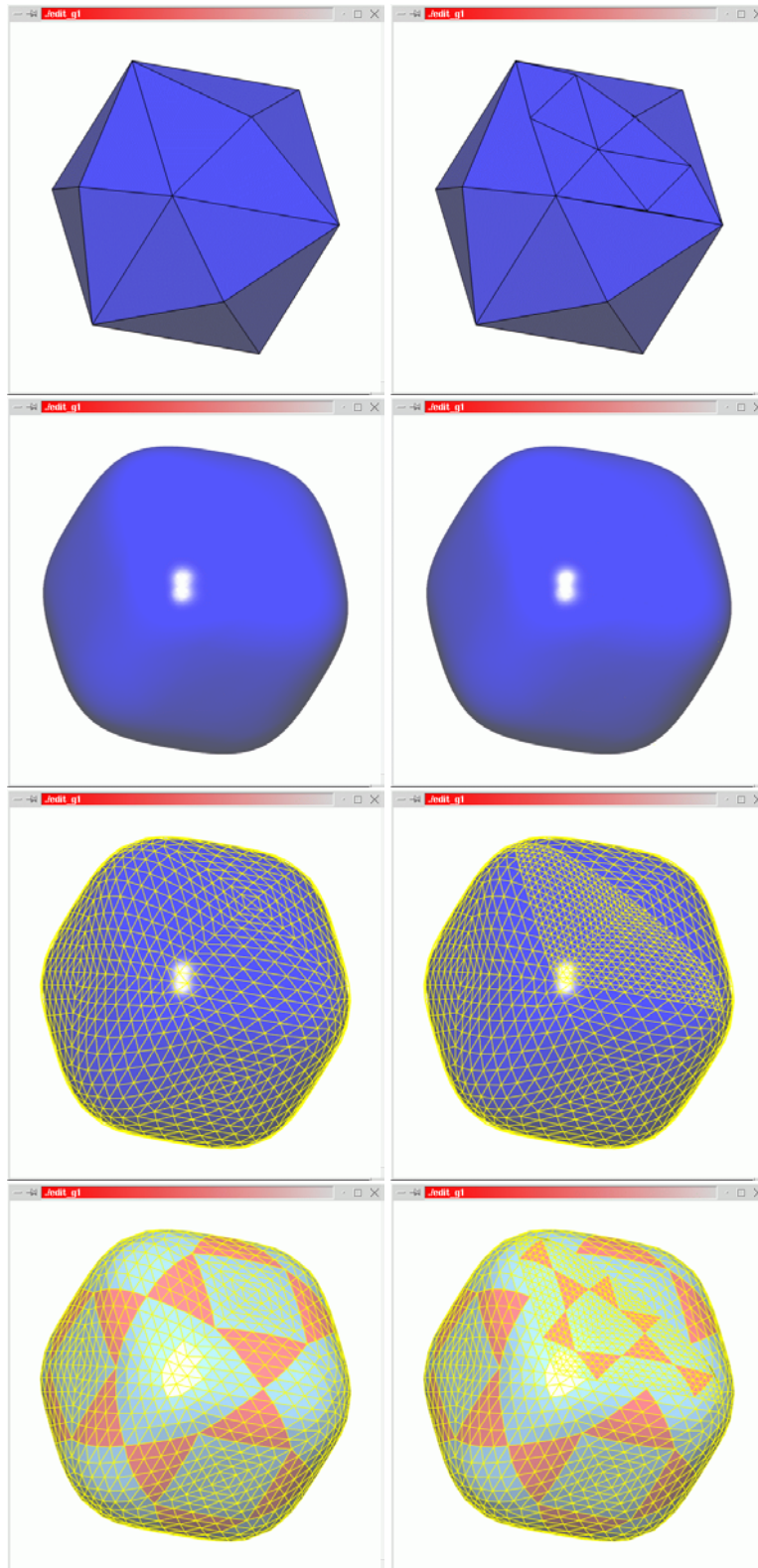


figure 26: Illustration de l'invariance par subdivision sur l'exemple du cube.



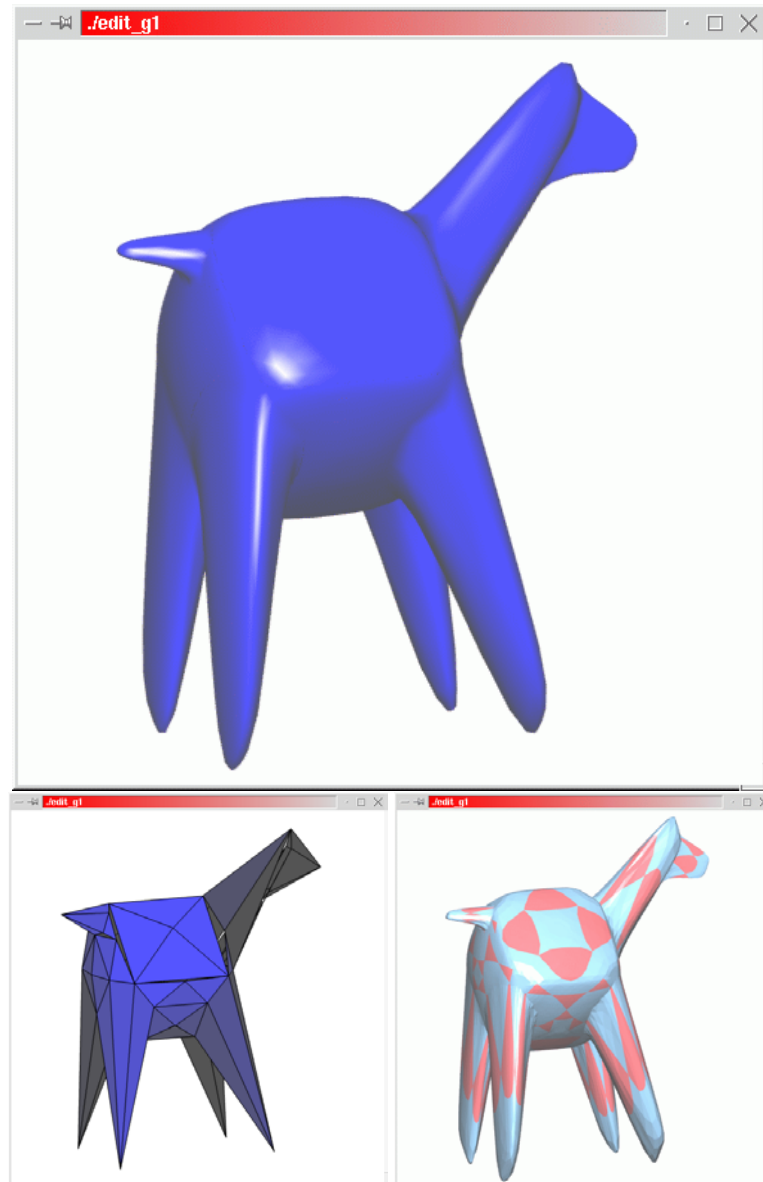


figure 27: Exemple de modèle défini sur 3 niveaux hiérarchiques.

---

## Chapitre 6

# Lissage de surfaces

Dans les chapitres suivants nous allons aborder la deuxième thématique de ce mémoire, dédiée au lissage de surfaces, qui complétera ainsi la thématique globale de mes travaux post-doctoraux, consacrés à la modélisation géométrique de surfaces lisses. Contrairement aux chapitres précédents qui portaient sur la construction d'une surface lisse en interpolant des données, les chapitres suivants porteront sur le lissage d'une surface déjà construite. Chronologiquement, mes travaux sur ce dernier sujet datent d'avant les travaux sur la construction de surfaces de topologie arbitraire, mais l'actualité du thème multirésolution fait qu'il a été préférable d'inverser l'ordre de présentation.

Les paragraphes 6.1 et 6.2 introduisent le très vaste sujet du lissage et le paragraphe 6.3 conclut ce chapitre en abordant les aspects de contrôle visuel de la qualité des surfaces, sur lesquels j'ai rédigé un état de l'art pour un livre [40]. Enfin les chapitres 7 et 8 suivants présenteront divers algorithmes de lissage pour certains types de surfaces déjà construites.

### 6.1) Importance d'une surface lisse

Le problème lié au "fairness" (qui peut approximativement se traduire par caractère lisse) est d'importance centrale durant le processus de design de surfaces de forme libre. Les courbes et surfaces sont souvent issues d'un processus de reconstruction d'un objet à partir de données discrètes de mesure. Celles-ci sont en général entachées d'erreurs de mesure, ce qui rend le procédé du design de "belles" courbes et surfaces complexes. Il y a deux manières de procéder à la reconstruction. D'une part, l'étape de la conception peut intégrer directement la notion de fairness en utilisant les paramètres libres afin de satisfaire un certain critère de lissage. Cette manière de procéder est dénommée *design lisse*. D'autre part, certains défauts de surface peuvent persister après avoir interpolé ou approximé des données: il est alors nécessaire de procéder à un *lissage* à posteriori, appelé *fairing*, qui corrige des parties de surface sans pour autant trop modifier la surface. Les travaux introduits dans ce chapitre et développés dans les deux chapitres suivants appartiennent à ce



second type de méthodes.

## 6.2) Mesures et algorithmes

Comme le mot "fairness" (caractère lisse) dans le monde du design (création, conception) est principalement lié à l'esthétique, il est très difficile de le décrire et de le définir de manière unique. Une surface est dite fair ou esthétiquement lisse quand elle est *visuellement plaisante*. Elle ne doit pas posséder

- de bosses, de creux, de zones plates, d'irrégularités surfaciques même très minimales de tout genre.

Le caractère lisse d'une surface est ainsi principalement lié à la notion de

- courbure et aux variations des courbures.

Une autre interprétation de la notion lisse fait référence à

- la continuité (géométrique ou paramétrique).

La fairness d'une surface réfléchissante, comme celle d'une carrosserie de voiture, peut également être caractérisée en terme de

- comportement des lignes de réflexion de la lumière.

D'autres critères plus techniques d'une imperfection surfacique peuvent être

- le dépassement d'une valeur limite de courbure imposée par exemple par l'outil de fraisage,

La diversité des caractérisations que nous venons de présenter fait qu'il n'y a pas de formule mathématique unique permettant de mesurer la fairness. Les travaux qui ont marqué ce domaine de recherche et d'application jusqu'aujourd'hui se sont inspirés de la physique. Quand on prend par exemple une fine latte en bois et qu'on lui applique des forces latérales aux extrémités, la courbe décrite par cette latte déformée sera toujours lisse. Physiquement, la latte minimise toujours l'énergie de tension  $\int_0^l \kappa(s)^2 ds$ . Les méthodes, dites "variationnelles" utilisent certaines énergies ou quantités physiques comme mesure de fairness. Pour les surfaces paramétriques on peut rencontrer :

- l'énergie de plaques minces (strain energy) [74]

$$\int_S (\kappa_1^2 + \kappa_2^2) dS, \quad (11)$$

avec  $\kappa_1, \kappa_2$  les courbures principales et  $dS$  l'élément de surface.

- l'énergie de plaques minces approchée [74, 34, 11]

$$\int_{\Omega} (\kappa_1^2 + \kappa_2^2) dudv, \quad (12)$$

- l'énergie de plaques minces linéarisée

$$\int_{\Omega} \left( \frac{\partial^2 S}{\partial u^2} \right)^2 + 2 \left( \frac{\partial^2 S}{\partial u \partial v} \right)^2 + \left( \frac{\partial^2 S}{\partial v^2} \right)^2 dudv, \quad (13)$$

- "jerk energy" approchée [36]

$$\alpha \int_a^b \int_c^d \left| \frac{\partial^3 X(u, v)}{\partial u^3} \right|^2 dudv + \beta \int_a^b \int_c^d \left| \frac{\partial^3 X(u, v)}{\partial v^3} \right|^2 dudv, \quad (14)$$

- les mesures de fairness aplatissantes, arrondissantes et roulantes [81]

$$\int_a^b \int_c^d \left| \frac{\partial C}{\partial u} \times \frac{\partial C}{\partial v} \right| dudv, \quad (15)$$

où  $C = K(u, v)N(u, v)$ ,  $X(u, v) \frac{H(u, v)}{K(u, v)}N(u, v)$ , ou  $[K(u, v) + H(u, v)^2]N(u, v)$ .

- les variations de courbures [69]

$$\int \left( \frac{d\kappa_1}{d\bar{e}_1} \right)^2 + \left( \frac{d\kappa_2}{d\bar{e}_2} \right)^2 dS, \quad (16)$$

où  $\bar{e}_1, \bar{e}_2$  sont les directions principales.

Quand les Bézier ou B-splines sont à la base des représentations surfaciques, l'énergie (12) peut être minimisée par une optimisation [63]. Elle est non-linéaire, car la fonctionnelle d'énergie (12) dépend non-linéairement des points de contrôle de la surface. Des méthodes de lissage non-linéaires basées sur l'énergie (16) existent également [69], mais elles ont souvent des temps d'évaluation des fonctionnelles et des temps de convergence des méthodes numériques itératives très grands [69].

L'approche commune qui évite ce problème consiste à supposer qu'une surface paramétrique est assez plate et d'en conclure que la métrique sur la surface ne diffère pas beaucoup de celle du plan. Ainsi les mesures de fairness basées sur les courbures  $\kappa_1^2 + \kappa_2^2$ ,  $|\kappa_1 \kappa_2|$ ,  $(\kappa_1 + \kappa_2)^2$  peuvent être approximées par des combinaisons des dérivées premières, secondes ou tierces qui sont linéaires en les coefficients. C'est l'approche qui a été à l'origine des mesures (13) et (14). Bien que cette supposition ne soit pas réaliste, elle est largement acceptée [74, 13 56]. Dans [31] une simplification plus sophistiquée de (11) pour des surfaces paramétriques est proposée.

Pour revenir au problème de départ, i.e. passer une surface lisse à travers un nuage de points mesurés, il est proposé par plusieurs auteurs de combiner l'expression de l'énergie avec celle de l'erreur totale au sein d'une minimisation par moindres carrés, voir p.ex. [36]. De manière générale, les paramètres libres d'une courbe ou surface sont utilisés soit au cours du processus de design, soit au lissage post-processing afin de minimiser des critères d'énergie. L'approche variationnelle a également été utilisée pour le développement de schémas de subdivision pour plaques minces [100]. A côté des méthodes variationnelles il existe des méthodes de fairing qui se basent sur d'autres mesures de fairness. Celles qui feront l'objet des deux chapitres suivants en sont des exemples.

Il existe également ce qu'on peut appeler le *pré-lissage des données*. Ces méthodes discrétisent les invariants surfaciques (par exemple les courbures)

par des différences finies et traduisent ainsi les mesures de fairness aux données discrètes [21]. Ainsi le fairing peut être appliqué aussi aux triangulations polygonales de topologie arbitraire, i.e. à des surfaces linéaires par morceau. Cela peut être utile si les sommets de la triangulation proviennent de capteurs 3D et sont entachés d'erreurs. Une méthode effective de lissage consiste à atténuer le bruit dans une triangulation en supprimant les fréquences hautes [98, 16]. D'autres approches consistent à minimiser une mesure de fairness discrétisée [101, 56]. Pour plus de références sur ce domaine qui n'est que voisin de celui du lissage de surfaces de forme libre, le lecteur pourra consulter la bibliographie des travaux cités ci-dessus.

### **6.3) Outils de contrôle visuel**

La fairness est principalement une caractéristique esthétique d'une surface qui ne peut se mesurer réellement que visuellement. Cela est évidemment très subjectif, mais c'est la démarche qui s'est imposée par exemple dans les logiciels de CAO, car les concepteurs de surfaces d'automobiles se servent des lignes de réflexion simulées pour décider si la qualité d'une surface est bonne. Bien qu'un algorithme de lissage a besoin de quantifier la fairness mathématiquement, ce qui décide souvent en pratique de la qualité d'une surface est son allure visuelle plaisante. Dans cette optique des méthodes de contrôle visuel des surfaces ont été développées.

Pour l'ouvrage collectif de C. Bajaj intitulé "Data Visualization Techniques" [2] j'ai été invitée à rédiger un état de l'art sur les méthodes de visualisation pour l'analyse et le contrôle de qualité de surfaces, en particulier utilisées dans le contexte du lissage [40]. Ce chapitre détaille les méthodes de simulation basées sur la réflexion lumineuse d'une surface mesurant ainsi sa fairness globale, les méthodes visualisant la convexité des courbes et surfaces, les méthodes à offset variable et à carte de couleur pour l'analyse des courbures surfaciques et d'autres propriétés se quantifiant par un nombre scalaire, et les méthodes des lignes caractéristiques (courbures, géodésiques) également pour l'analyse de certaines propriétés géométriques.

---

## Chapitre 7

# Une approche heuristique pour le lissage

Puisque le lissage est une propriété globale d'une surface la plupart des méthodes de lissage déjà citées dans le chapitre précédent sont également de nature globale. Une autre approche, *locale* celle-ci, consiste à lisser une surface déjà existante en modifiant *itérativement* un petit ensemble de points de contrôle à chaque fois afin d'obtenir une forme globalement plus lisse. L'idée commune à ces algorithmes est de converger, par des itérations locales, vers la solution d'un problème global.

Il existe une approche de ce type [32] qui se base sur un critère de lissage très classique, une simplification linéarisée de l'énergie de plaque mince:  $E = \int \int_A X_{uu}^2 + X_{uv}^2 + X_{vv}^2 dudv$ . A chaque itération un seul point de contrôle est déplacé afin de minimiser la mesure de lissage. L'algorithme est rapide et semble obtenir de bons résultats. L'inconvénient, qui d'ailleurs est commun à toutes les méthodes utilisant des critères variationnels simplifiés, est que la fonctionnelle  $E$  n'approxime l'énergie de plaques minces que sous certaines conditions qui ne sont pratiquement jamais vérifiées par les surfaces paramétriques. Plus de détails à ce sujet se trouvent dans [31].

Une autre approche de type local itératif, due à Farin et al. [25] se base sur une mesure de lissage très rapide à évaluer: la somme des sauts de la dérivée de la courbure en les nœuds de la courbe  $\xi = \sum_i |\kappa'(t_i^-) - \kappa'(t_i^+)|$ . Cette mesure qui ne s'applique qu'aux courbes (cubiques  $C^2$  par morceau) est d'origine géométrique. L'algorithme utilise des suppressions/réinsertions de nœuds comme étape du lissage. L'inconvénient est qu'il ne converge pas vers une courbe optimale. Les auteurs proposent simplement d'arrêter les itérations au premier minimum qu'atteint la mesure de lissage. Ces méthodes itératives sont locales, donc très rapides et elles permettent également de restreindre si désiré la zone à lisser sur la surface.

Les travaux que nous allons présenter dans ce chapitre et qui ont été publiés

dans [37] et [38] portent sur la continuation des travaux de Farin et Sapidis en les étendant aux surfaces B-splines. Le problème de la non-convergence héritée de l'algorithme sur des courbes est plus complexe dans le cas des surfaces. Deux approches heuristiques sont alors introduites dans cet algorithme de lissage pour la résolution de ce problème. La section 7.1 décrit l'opération de base effectuée lors de chaque itération de notre algorithme, et propose un premier algorithme de lissage. La section 7.2 reprend la même itération, mais propose des heuristiques plus élaborées pour lisser une surface.

### 7.1) Lissage local et itératif de surfaces B-spline bicubiques

Dans cette section, nous présentons un algorithme de lissage pour des surfaces B-spline  $C^2$  cubiques. Cette algorithme est basé sur une suite d'itérations modifiant localement la surface. Les sections 7.1.1 et 7.1.2 présentent des méthodes de lissage de courbes, desquelles notre algorithme s'est inspiré. La section 7.1.3 présente une généralisation naïve de ces méthodes aux surfaces, qui a le défaut de ne pas être locale. Enfin la section 7.1.4 présente la généralisation localisée et un algorithme de lissage basé sur cette généralisation.

#### 7.1.1 La mesure de lissage de Kjellander

Kjellander [55] a été le premier à proposer un algorithme de lissage pour des courbes cubiques par morceau et  $C^2$  continues basé sur une mesure de lissage particulièrement adaptée à ce type de courbes: la somme des sauts des discontinuités de la dérivée de la courbure en les nœuds (points de jonctions) de la courbe:

$$\xi = \sum_i z_i, \quad \text{avec} \quad z_i = |\kappa'(t_i^+) - \kappa'(t_i^-)|. \quad (17)$$

Ce critère géométrique est motivé par le fait qu'une courbe est considérée lisse quand

*sa fonction de courbure est continue, a le signe appropriée, et est le plus proche possible d'une fonction monotone par morceau, avec le moins de morceaux possibles.*

Kjellander se rapproche de cet idéal en diminuant itérativement les sauts des dérivées de la courbure. Les itérations de Kjellander sont néanmoins globales, car elles entraînent un recalcul de la courbe entière due au fait qu'il s'agit de courbes Hermite  $C^2$ . La généralisation aux surfaces [57], produit tensoriel d'Hermite, a le même défaut. La méthode est itérative mais globale pour chaque itération.

#### 7.1.2 L'algorithme KRR de Farin/Sapidis

L'annulation de la mesure locale  $z_i$  (17) en un nœud d'une courbe  $C^2$  cubique par morceau est équivalent à la rendre  $C^3$ , donc  $C^\infty$  en ce nœud, ce qui revient également au remplacement de deux segments de courbes par un seul. L'algorithme de Farin et Sapidis [25, 26, 85] exécute cette opération pour des courbes B-spline, en modifiant seulement localement un petit nombre de trois ou cinq points de contrôle par une opération que l'on notera KRR (Knot

Removal and Reinsertion). A chaque itération le nœud  $t_i$  avec la plus grande valeur  $z_i$  est sélectionné. L'opération KRR enlève ce nœud de la suite des nœuds. La nouvelle courbe ainsi obtenue approxime l'ancienne, car un point de jonction a disparu, deux segments cubiques sont maintenant approximés par un seul. Cette procédure est locale, elle ne fait intervenir que trois ou cinq points de contrôle. Ensuite, le même nœud  $t_i$  est réinséré, pour ne pas changer la structure initiale de la courbe. Cette réinsertion par contre ne modifie pas la forme de la courbe.

A chaque itération une mesure  $z_i$  est mise à 0, et en même temps les valeurs  $z_{i-1}$  et  $z_{i+1}$  sont modifiées. Il se peut donc que cette opération sur le nœud  $t_i$  n'entraîne pas une diminution de la mesure globale  $\xi$ . L'algorithme ne converge donc pas forcément vers le minimum global de la fonction  $\xi$ , et de plus la suite des valeurs de  $\xi$  au cours du lissage n'est pas monotone décroissante. Il s'arrête au premier minimum atteint.

### 7.1.3 KRR pour des surfaces B-spline

Une définition produit tensoriel pour des surfaces polynomiales offre la possibilité d'appliquer directement la plupart des algorithmes sur des courbes (De Casteljau, élévation de degré, dérivation, ...), en fixant un des paramètres et en exécutant l'algorithme dans la direction de l'autre paramètre. Cette procédure est répétée pour chacune des valeurs du premier paramètre. Enfin une dernière application de l'algorithme dans la direction du premier paramètre permet d'obtenir le résultat final.

Soit  $\mathbf{u} = (u_i)_{i=0}^{n+4}$  et  $\mathbf{v} = (v_i)_{i=0}^{m+4}$  deux suites de nœuds et

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m \mathbf{d}_{ij} N_{i,4} N_{j,4}, \quad (u, v) \in [u_3, u_{n+1}] \times [v_3, v_{m+1}]$$

une surface produit tensoriel B-spline bi-cubique où  $\mathbf{d}_{ij} \in \mathbb{R}^3$  sont les points de contrôle,  $N_{i,4}$  les fonctions de base B-spline d'ordre 4 et  $(u_k, v_l)$  ( $k, l = 0, \dots, n, m$ ) les nœuds intérieurs.

La suppression d'un nœud surfacique intérieur correspond en fait à la suppression de toute une ligne de nœuds dans la grille sous-jacente des nœuds intérieurs  $(u_k, v_l)$ . L'opération KRR sur un nœud  $v_l$ , noté  $v_l$ -KRR correspond ainsi à l'application d'un KRR unidimensionnel aux  $n + 1$  courbes B-spline obtenues pour  $i$  constant.

Après réinsertion du même nœud  $v_l$ , 3 ou 5 rangées de points de contrôle ont été modifiées par cette opération par rapport à la surface initiale. Le nombre de rangées modifiées dépend de l'algorithme de suppression utilisé, (voir [65, 26, 85, 38] pour les détails). Une étape de KRR surfacique en direction  $u$ , un  $u_k$ -KRR, fonctionne de manière analogue.

*Remarque:*

Une étape  $v_l$ -KRR rend les dérivées partielles tierces en direction  $v$  de la surface continue, i.e.  $\frac{\partial^3 S}{\partial v^3}(u, v_l^-) = \frac{\partial^3 S}{\partial v^3}(u, v_l^+)$ . Un résultat analogue existe pour l'étape  $u_k$ -KRR. Par contre, il est important à noter que l'application successive de  $u_k$ -KRR et puis  $v_l$ -KRR, ou l'inverse, ne rend la surface généralement pas  $C^3$ -continue au nœud  $(u_k, v_l)$ . Seule la continuité de la dernière application KRR est garantie à la fin de cette double procédure. En effet l'application du deuxième opérateur de lissage "casse" le résultat du premier opérateur. C'est d'ailleurs la remarque qui a motivé les travaux qui seront présentés au chapitre 8.

### 7.1.4 Localisation du KRR, et algorithme de lissage

Une modification du KRR surfacique qui le rend mieux adapté pour le lissage de surfaces permet dans la suite de présenter un algorithme itératif de lissage pour les surfaces B-spline produit tensorielles bicubiques par morceau qui contrairement à la méthode globale de Kjellander est local. C'est plutôt une extension des travaux de Farin/Sapidis mais qui apporte en plus une solution au problème de convergence.

Il s'avère que le KRR surfacique décrit en section 7.1.3, n'est pas utilisable en tant que tel comme un pas de lissage "local", car 3 voir 5 lignes entières de points de contrôle de la surface sont modifiés. Ce n'est pas suffisamment local. C'est peut-être la raison pour laquelle Farin/Sapidis proposent (dans la conclusion de [26]) comme extension de leur méthode aux surfaces de simplement considérer les rangées et colonnes de points de contrôle comme courbes B-spline et de les lisser indépendamment les unes les autres avec leur algorithme courbe. Cela n'est pas vraiment un lissage de surface, car aucun critère surfacique n'est diminué.

Notre algorithme essaye d'être une "vrai" extension surfacique. La démarche suivie profite de deux observations :

- Une étape de KRR surfacique ne change pas le nombre de points de contrôle de la surface.
- Le but de rendre une dérivée partielle troisième continue au nœud  $(u_k, v_l)$  peut être également atteint en n'appliquant le KRR unidimensionnel qu'à 3 (ou 5) (voir dans l'article) courbes B-spline. C'est ce que nous appelons le *KRR localisé*.

Chaque étape de KRR localisé que ce soit en  $v_l$  ou  $u_k$  ne modifie ainsi que  $3 \times 3$  (ou  $5 \times 5$ ) points de contrôle de la surface et rend la dérivée partielle troisième continue en direction  $v$  ou  $u$  resp. Les formules et explications détaillées se trouvent publiées dans [38].

L'algorithme de lissage qui en résulte mesure la fairness au nœud  $(u_k, v_l)$  par une *mesure locale*  $(z_{kl}^u + z_{kl}^v)$  où  $z_{kl}^u$  (ou  $z_{kl}^v$  resp.) est égal à la différence positive de la dérivée partielle en  $u$  (ou  $v$  resp.) d'une fonction de courbure normalisée au nœud  $(u_k, v_l)$ . Nous avons fait ce choix d'une mesure dépendant

de la paramétrisation dans un souci de rapidité d'évaluation. Il serait possible de la remplacer par une mesure intrinsèque qui impliquerait l'évaluation des courbures principales et des directions principales. Ceci est extrêmement coûteux comme le montrent les algorithmes de lissage de Séquin/Moreton [69].

La somme des mesure locales

$$\xi := \sum_{k,l} z_{kl}^u + z_{kl}^v \tag{18}$$

forme la *mesure globale de fairness* que l'algorithme doit minimiser. Au départ la mesure locale est évaluée en chaque nœud afin de déterminer le nœud représentant la plus grande mesure locale. La partie de la mesure locale la plus grande détermine la direction du  $u/v$ -KRR localisé à appliquer comme étape de lissage. Quand p.ex.  $z_{kl}^u > z_{kl}^v$ , alors un  $u_k$ -KRR est appliqué. Ensuite les valeurs locales sont mises à jour pour répéter ce procédé jusqu'à ce qu'un certain critère décide d'arrêter les itérations.

### 7.2 Recherche heuristique d'une solution optimale

L'algorithme de lissage présenté dans le paragraphe précédent sélectionne le nœud et la direction pour laquelle la valeur de la mesure locale est la plus grande. C'est la valeur qui contribue le plus à la mesure globale de fairness  $\xi$  (18) de la surface. Le fait que le KRR localisé la mette à zéro laisse donc supposer une diminution de  $\xi$ . Mais un certain nombre de mesures locales en des nœuds voisins sont également modifiées lors d'une étape de lissage sur un nœud. Il est donc possible qu'après plusieurs itérations  $\xi$  ne diminue pas. Un minimum de  $\xi$  atteint par ces itérations n'est que local et une autre suite d'itérations de KRR localisé peut conduire à un autre minimum local de la fonction  $\xi$ , qui représente peut-être une surface plus lisse. Quelle est donc la suite d'étapes  $u/v$ -KRR localisés qui mène au minimum global de  $\xi$  ?

Une recherche systématique de la solution optimale reviendrait à fixer un nombre maximal d'itérations  $k$  et à parcourir un arbre de profondeur  $k$  où chaque nœud parent a  $q$  fils ( $q$  étant le nombre de nœuds intérieurs de la surface, candidats au KRR).

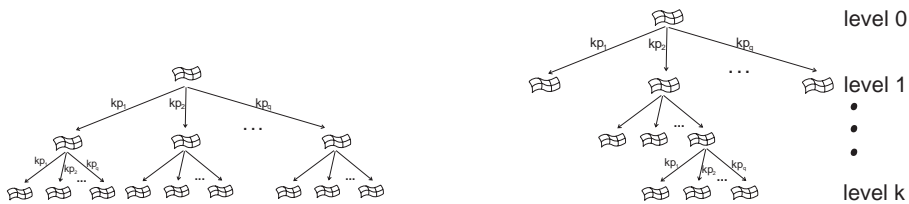


figure 28: A gauche: arbre de recherche systématique. à droite: exemple d'un arbre BFS. "kp<sub>i</sub>" désigne les couples de nœuds intérieurs (knot pair).

La complexité de cette recherche serait  $O(q^k)$ . Comme la complexité de cette recherche systématique d'une solution optimale est trop élevée pour permettre un temps d'exécution acceptable, nous proposons l'emploi d'une méthode heuristique d'optimisation. Ce type de méthodes est très peu utilisé en CAGD.



A la connaissance de l'auteur elles n'ont été utilisées en CAGD que pour la construction de triangulations optimales d'un ensemble de points 2D [89]. Pourtant leur application est très bien adaptée au problème du lissage. En effet dans la grande majorité des algorithmes de lissage existants la mesure mathématique de fairness à minimiser n'est pas directement liée à la mesure de fairness visuelle (ligne de réflexion), qui elle est déterminante pour un jugement sur la qualité. Il n'est donc pas primordial de trouver le minimum global tant qu'on en est proche et que le résultat visuel est satisfaisant.

Nous avons choisi de poursuivre deux approches pour la sélection de la suite de nœuds en lesquels l'opération de lissage  $u/v$ -KRR localisée est appliquée dans le but de minimiser le plus possible la mesure globale de fairness.

### **Best First Search**

Le Best-First-Search [82], noté BFS, cherche à réduire la complexité polynomiale de la recherche systématique du minimum global dans l'arbre, (voir fig. 28-gauche). L'idée est de ne parcourir que les chemins intéressants, i.e. les chemins qui mènent probablement à la surface la plus lisse.

Notre algorithme de lissage BFS [38] fonctionne de la manière suivante. Partant de la surface initiale le premier niveau de l'arbre est créé en exécutant l'opération KRR à tous les  $q$  nœuds intérieurs en partant à chaque fois de la surface initiale. Chaque fils de la racine représente ainsi une nouvelle surface ayant subi une étape de lissage. Toutes ces surfaces sont triées par rapport à leur mesure globale de fairness  $\xi$  (18). La même procédure est appliquée à la surface ayant la plus grande valeur  $\xi$  (i.e. ayant eu la première position dans la liste triée et éliminée ensuite). Puis tous ses fils sont rangés dans la liste triée. Et la procédure recommence avec le premier élément de la liste. L'algorithme se termine après une profondeur préfixée de l'arbre  $k$ . La complexité "best case" est de  $O(k \cdot q)$ , (voir en fig. 28-droite), celle du "worst case" est  $O(q^k)$  comme pour la recherche systématique.

### **Simulated Annealing (recuit simulé)**

Le simulated annealing, noté SA, est une méthode heuristique pour la résolution de grands problèmes d'optimisation. Elle assure à une "grande probabilité" de trouver un minimum global d'une fonction coût. En 1983, des chercheurs IBM ont développé cet algorithme en analogie avec un processus heuristique en physique de matériaux: celui du refroidissement optimal de métal, qui est une tâche très complexe.

Le SA est un algorithme itératif qui permet de faire des "bons pas" (ceux qui diminuent le coût) et des "mauvais pas" (ceux qui augmentent le coût). Ces mauvais pas sont nécessaires pour pouvoir sortir d'un minimum local de la fonction coût. La probabilité qu'un mauvais pas soit accepté diminue lors de l'avancée des itérations. La topographie d'une fonction de coût unidimensionnelle est illustrée en fig. 29. L'algorithme offre beaucoup de paramètres qui sont à fixer en fonction du problème à résoudre (comme par exemple la fonction probabilité décroissante), et qui génèrent les conditions d'arrêt.

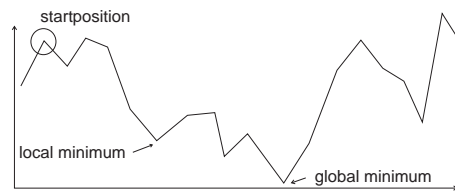


figure 29: Topographie d'un processus d'optimisation SA unidimensionnel.

Dans le cas du lissage la fonction coût à minimiser est la mesure globale de fairness  $\xi$  (18). Pour chaque itération KRR un nœud est choisi soit au hasard soit en fonction de la mesure locale. Le SA ne garantit pas de trouver le minimum global, mais son utilisation s'est avéré très rapide et efficace. Des exemples sont donnés dans [37] et encore plus détaillés dans [38] (inclus page 171 de ce mémoire).

---

**Chapitre 8****Fairing local et itératif  
utilisant des masques**

Nous avons pu montrer dans le chapitre précédent que l'opération de lissage KRR qui est globale sur une surface peut être modifiée de telle manière à ce qu'elle donne lieu à un algorithme de lissage local de surfaces B-spline. De plus, l'utilisation de méthodes heuristiques a permis d'optimiser la recherche d'une surface la plus lisse possible tout en conservant des temps d'exécution très courts. Un inconvénient des méthodes présentées en chapitre 7 ressort de la remarque faite en section 7.1.3: l'opération locale de lissage faite en un nœud à chaque itération de l'algorithme n'assure pas la continuité  $C^3$  "totale" en ce nœud, mais seulement la continuité  $C^3$  le long d'une seule direction paramétrique en ce nœud.

Dans le présent chapitre nous apportons un nouvel algorithme de lissage local et itératif qui à chaque itération rend la surface localement  $C^3$  continue en un nœud. L'opération de lissage se base sur les conditions de continuité  $C^3$  en un nœud de la surface et s'applique comme un filtre à un sous-ensemble (local) de points de contrôle. Plusieurs extensions de cette idée de base ont été développées et ont donné lieu à une publication dans une revue [39] (jointe page 181 de ce mémoire).

**8.1) Mesure de fairness**

L'objectif de ce paragraphe est l'établissement et l'évaluation de la mesure locale de lissage ainsi que la description d'une étape de lissage. L'algorithme vise à lisser des surfaces B-spline bicubiques de continuité  $C^2$ . C'est la classe de surfaces la plus souvent utilisée en pratique.

**Notations**

Rappelons brièvement les notations importantes pour la suite. Soient deux

entiers positifs  $n, m \geq 3$  et deux séquences de nombre réels  $\mathbf{u} = (u_i)_{i=0}^{n+4}$ ,  $\mathbf{v} = (v_j)_{j=0}^{m+4}$  avec  $u_i < u_{i+4}$  et  $v_j < v_{j+4}$ . Les fonctions de base B-spline cubiques par morceau associées avec la suite de nœuds  $\mathbf{u}$  sont notées  $N_{i,4,\mathbf{u}}$  (ou alors simplement  $N_{i,4}$ ). Elles sont normalisées pour se sommer à 1. Une surface produit tensoriel B-spline  $S$  d'ordre (4,4) est alors définie par

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m \mathbf{d}_{ij} N_{i,4}(u) N_{j,4}(v), \quad (u, v) \in \Omega := [u_3, u_{n+1}] \times [v_3, v_{m+1}],$$

où  $\mathbf{d}_{ij} \in \mathbb{R}^3$  sont les points de contrôle formant le réseau de contrôle de  $S$ .

Les couples de nœuds  $(u_k, v_l)$  dont les indices  $(k, l)$  appartiennent à  $I = \{(4, 4), (4, 5), \dots, (4, m), \dots, (n, m)\}$  sont les nœuds intérieurs de la surface. Nous les appelons les *couples de nœuds libres*, car ils sont les candidats pour les étapes de lissage. Une surface B-spline bicubique est composée de  $(n-2)(m-2)$  patches polynomiaux qui se raccordent avec continuité  $C^2$  aux nœuds quand tous les nœuds sont simples (i.e.  $u_i < u_{i+1}$ ,  $v_j < v_{j+1}$ ,  $i, j = 3, \dots, n, m$ ).

Il est important de remarquer que la propriété du *support local* des fonctions de base implique que chaque point de contrôle  $\mathbf{d}_{ij}$  n'a qu'une influence locale sur la surface, i.e. une influence aux seuls paramètres  $(u, v) \in [u_i, u_{i+4}] \times [v_j, v_{j+4}]$ .

### La mesure de lissage en fonction des points de contrôle

Le principe de lissage est encore le même que celui de Kjellander [57], Farin/Sapidis [25] et celui du chapitre précédent, à savoir qu'une surface B-spline bicubique de classe  $C^2$  est considérée être plus lisse au couple de nœuds  $(u_k, v_l)$ ,  $(k, l) \in I$ , quand elle y est de continuité  $C^3$ .

Une étape locale de lissage exécutée au nœud  $(u_k, v_l)$  a ainsi pour but de diminuer (voir de mettre à 0) la somme des différences des dérivées partielles d'ordre 3 en  $(u_k, v_l)$ . Et l'algorithme itératif a pour but de diminuer la somme de ces mesures locales, globalement sur toute la surface, i.e. en tous les nœuds libres.

Une surface  $S$  est de continuité  $C^3$  au point  $(u, v)$  si et seulement si toutes les dérivées partielles d'ordre 3  $S_{u^\nu v^\mu} = \frac{\partial^{\nu+\mu} S}{\partial u^\nu \partial v^\mu}$ , ( $\nu + \mu = 3, \nu, \mu \in \mathbb{N}_0$ ) de  $S$  sont continues en  $(u, v)$ . Dans le cas d'une représentation B-spline de la surface  $S$  nous pouvons profiter d'une certaine régularité des fonctions de base, qui fait que toutes les dérivée partielles mixtes d'ordre 3

$$\mathbf{X}_{u^\nu v^\mu}(u, v) = \nu! \mu! \sum_{i=\nu}^n \sum_{j=\mu}^m \mathbf{d}_{ij}^{(\nu, \mu)} N_{i,4-\nu}(u) N_{j,4-\mu}(v), \quad (19)$$

pour  $\nu, \mu \geq 1$ , sont continues. Les  $\mathbf{d}_{ij}^{(\nu, \mu)}$  sont des combinaisons linéaires de certains points de contrôle qui se calculent en appliquant les formules de dérivation connues pour les surfaces B-spline [27].

Cette continuité est due au fait que les fonctions B-spline  $N_{i,4-\nu}$  et  $N_{j,4-\mu}$  sont continues quand  $\nu + \mu = 3$  et  $\nu, \mu \geq 1$ . Seules les dérivées partielles

d'ordre 3 en une seule direction interviennent ainsi dans la mesure de fairness locale au nœud  $(u_k, v_l)$ , que l'on peut donc définir de la manière suivante:

$$L_{k,l} := \|\Delta_{uuu}(u_k, v_l)\|^2 + \|\Delta_{vvv}(u_k, v_l)\|^2, \quad (k, l) \in I, \quad (20)$$

où les termes  $\Delta_{uuu} = X_{uuu}^+ - X_{uuu}^-$  et  $\Delta_{vvv} = X_{vvv}^+ - X_{vvv}^-$  peuvent être exprimés en fonction de 21 points du réseau de contrôle. Ils sont illustrés schématiquement en fig. 30. Leur formulation exacte est précisée dans [39], qui se trouve en page 181 de ce mémoire.

La mesure globale de fairness consiste ainsi en la somme des mesures locales sur toute la surface

$$G_S := \sum_{(k,l) \in I} L_{k,l} \quad (21)$$

ce qui correspond au principe de fairness évoqué au début de ce paragraphe.

## 8.2 Une étape de lissage - l'algorithme "fairC3"

L'objectif d'une étape de lissage est double :

- mettre la valeur local  $L_{k,l} = 0$  au nœud qui contribue le plus à la mesure globale  $G_S$ ,
- déformer la surface le moins possible.

L'étape de fairing que nous proposons va concilier ces deux objectifs de la manière suivante.  $L_{k,l} = 0$  correspond aux deux conditions

$$\Delta_{uuu} = 0 \quad \text{et} \quad \Delta_{vvv} = 0. \quad (22)$$

Il a été évoqué dans la section précédente que 21 points de contrôle de la surface (voir fig. 30) interviennent dans le calcul de ces deux quantités. Il y a donc beaucoup plus de degrés de liberté que de conditions à satisfaire. Pour justement satisfaire le deuxième objectif de déformation minimale, une minimisation au sens des moindres carrés des distances entre les nouveaux et les anciens points de contrôle correspondants est utilisée, sous la contrainte d'obtenir  $L_{k,l} = 0$ . Les 12 points de contrôle extérieurs, marqué avec 0 en fig. 30 restent fixés, i.e. ne seront pas modifiés, ce qui rendra l'étape de lissage encore plus locale.

L'étape de fairing peut ainsi être décrite par

$$\text{Minimiser} \quad F(\hat{\mathbf{d}}_{ij}) = \sum_{i=k-3}^{k-1} \sum_{j=l-3}^{l-1} \|\mathbf{d}_{ij} - \hat{\mathbf{d}}_{ij}\|^2 \quad (23)$$

sous les deux contraintes (22) ,

où  $\hat{\mathbf{d}}_{ij}$  désignent les 9 points de contrôle modifiés de la surface.

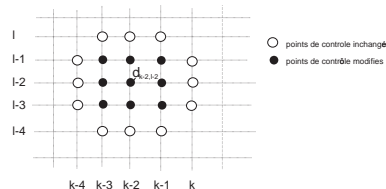


figure 30: Les points de contrôle impliqués dans un pas de lissage.

*Remarque 1: Aspects pratiques*

La technique des multiplicateurs de Lagrange permet de résoudre ce problème d'optimisation par un système linéaire ( $11 \times 11$ ), dont la matrice ne dépend que des nœuds. Il n'y a donc pas de matrices à inverser quand il s'agit de lisser une surface B-spline uniforme. Il suffit dans ce cas de calculer l'inverse une fois pour toute et de n'effectuer que des multiplications matrice-vecteur à chaque itération.

*Remarque 2: Comparaison avec une étape de KRR surfacique*

Les 9 points de contrôle sont identiques à ceux utilisés dans une étape de KRR surfacique de la méthode de lissage du chapitre 7 précédent, mais cette fois la surface devient  $C^3$  continue au nœud en lequel l'étape de lissage s'applique.

*Remarque 3: L'algorithme FairC3*

L'algorithme local et itératif correspondant à l'étape de lissage décrite plus haut, peut être schématisé par le squelette suivant:

1. Calculer la valeur initiale de la mesure globale  $G^0$ : (21)
  2. WHILE (condition d'arrêt = false) DO
    - (a) choisir  $(u_k, v_l)$ , tel que  $L_{kl} = \max_{(ij) \in I} L_{ij}$ : (20)
    - (b) exécuter une étape locale de fairing au nœud  $(u_k, v_l)$ : (23)
    - (c) calculer la nouvelle mesure globale  $G^j$
- END.

**8.3) Extensions de l'étape de fairing: utilisation de masques de lissage**

La technique de lissage qui rend la surface localement  $C^3$  continue au nœud traité, introduite dans la section précédente, s'avère être exploitable dans une optique plus large. Il est en effet possible d'imposer la continuité  $C^3$  non seulement en un nœud, mais aussi en un groupe de nœuds correspondants par exemple à 4 points, ou à 1 segment complet. Les détails de ces généralisations ainsi que d'autres extensions sont donnés dans [39]. Une étape d'itération correspond alors à la modification de 16 points pour le schéma "4-points", ou de 20 points pour le schéma "1-segment", comme illustré en fig. 31 et 32.

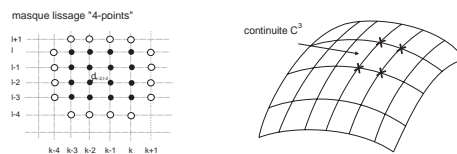


figure 31: Les points de contrôle impliqués dans le pas de lissage 4-point.

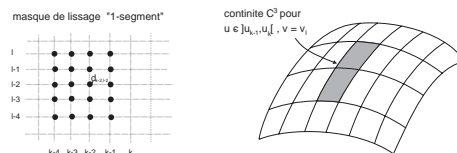


figure 32: Les points de contrôle impliqués dans le pas de lissage 1-segment.

## Résultats

Les résultats expérimentaux que nous avons obtenus sont satisfaisants. Premièrement les temps d'exécution sont courts, et deuxièmement les modifications des points de contrôle de la surface sont minimales. Ce dernier point est très important car un algorithme de lissage ne doit pas "trop" déformer la surface. Nous mesurons après chaque étape de lissage l'erreur maximale entre les points de contrôle de la nouvelle surface et ceux de la surface initiale. Quand on souhaite ne pas dépasser une certaine tolérance prédéfinie, alors on peut interdire les étapes de lissage qui font dépasser cette tolérance. L'erreur que nous mesurons est de plus une erreur de sécurité, car elle ne mesure que les distances entre points de contrôle. Or, les distances réelles entre les surfaces sont toujours inférieures ou égales à celles-ci grâce à la propriété "enveloppe convexe" des surfaces B-spline [27].

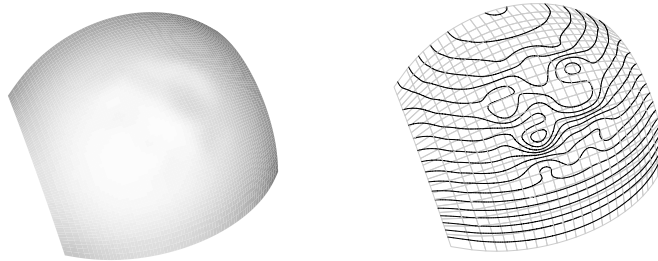


figure 33: Surface avec 225 points de contrôle. Les imperfections sont visualisées par des isophotes tracées sur la surface.

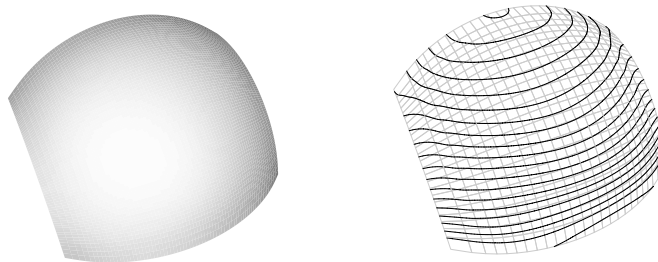


figure 34: Surface issue de l'algorithme de base 1-point après 500 itérations.

Le premier exemple que nous montrons est extrait de l'article [39]. La surface possède  $15 \times 15$  points de contrôle et a une paramétrisation uniforme. Cette surface a été obtenue à partir de points de contrôle appartenant à la sphère unité, et en déplaçant les points intérieurs légèrement pour introduire des imperfections dans la surface. En fig. 33 est illustrée la surface initiale, à gauche par une image ombrée, à droite avec des isophotes (courbes pour lesquelles l'angle entre la normale et un vecteur donné est fixe) témoignant des imperfections sévères que nous avons artificiellement créées. Au bout de 500 itérations, l'algorithme de base (1-point) résulte en la surface illustré en fig. 34. Les lignes d'isophotes se sont nettement améliorées. La mesure globale de fairness a diminuée de  $G_{X_{init}} = 0.78$  à  $G_{X_{fair}} = 0.0001$ . L'erreur relative maximale entre les points de contrôle  $\tau = \max \|\mathbf{d}_{ij} - \hat{\mathbf{d}}_{ij}\| / \max \|\mathbf{d}_{ij} - \mathbf{d}_{kl}\|$  s'élève à 0.01, la valeur moyenne étant de 0.001. Le temps d'exécution est très rapide avec 0.3 s sur une SGI Octane.

Le deuxième exemple que je souhaite présenter n'a pas été publié. Il a seulement été présenté lors d'un congrès SIAM on Geometric Design à Nashville. C'est un exemple "real world", car la surface provient d'une conception CAO\* réelle, représentant une partie du capot d'une voiture de marque Daimler-Benz, (voir fig. 35). Les lignes d'isophotes dans la deuxième figure (en partant de la gauche) visualisent une imperfection qui doit être lissée sans trop déformer la surface, i.e. sans que la forme globale ne change. C'est ce type de surfaces auquel les algorithmes de lissage doivent réellement faire face. Il est donc important de pouvoir valider un algorithme sur de tels exemples réels.

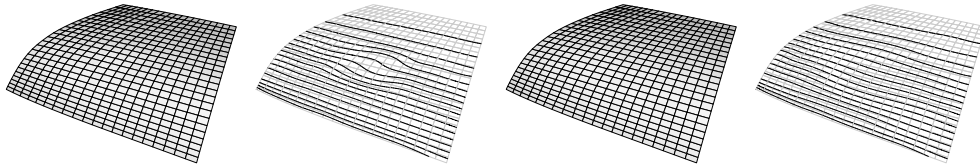


figure 35: A gauche: une petite bosse dans la surface d'un capot d'une voiture visualisé par des isophotes tracées sur la surface. A droite: la surface obtenue après lissage.

---

\* Courtesy P. Kaklis, NTU Athènes, TMR Fairshape



---

## Bilan et perspectives

Les travaux de recherche qui ont fait l'objet de ce mémoire traitent de deux aspects de la modélisation géométrique de surfaces lisses. Le premier thème (chapitres 6-8) concerne le lissage de surfaces spline déjà existantes. Les algorithmes de lissage heuristiques (chap. 7) ainsi que l'algorithme 'fairC3' (chap. 8) sont itératifs et locaux. Ils opèrent sur un critère de lissage qui, contrairement à ce qui se fait classiquement, n'est pas proportionnel à une énergie. Ce critère prend en compte la somme des sauts de la dérivée troisième. Le but est de modifier la surface afin de minimiser ce critère, et ainsi de lisser la surface. L'inconvénient que peut représenter la dépendance de la paramétrisation est compensé par des propriétés telles que la localité et l'invariance affine, au moins pour l'algorithme 'fairC3'. Les algorithmes ont été validés par un ensemble de cas tests comprenant un cas réel de carrosserie de voiture.

Une ouverture possible des méthodes développées aurait pu être le lissage de données discrètes, comme p.ex. les triangulations représentant des objets 3D captés par un scanneur 3D. Ce pas vers cette autre application n'aurait pas été très grand, car les algorithmes que nous avons développé agissent déjà comme des "filtres" sur les points de contrôle de la surface. Ils sont d'ailleurs proches des techniques introduites en 1995 [98] pour les triangulations. Plutôt que d'aborder ce thème, j'ai préféré privilégier une autre direction de recherche tout en restant fidèle aux surfaces lisses: les méthodes multirésolution (MR) en modélisation géométrique. Le succès que connaissent aujourd'hui ces méthodes me donne finalement raison.

Le deuxième thème est l'interpolation lisse de triangulations polygonales. Nos travaux sur ce sujet (chapitres 1-5) qui ont débuté en 1998, et en particulier les récents travaux sur l'interpolant hiérarchique, font aujourd'hui partie d'un des 'grands' thèmes d'actualité en modélisation géométrique: la multirésolution. Les différents interpolants que nous avons développé reposent tous sur la décomposition du type 4-split, qui consiste à subdiviser régulièrement en 4 les triangles. Ce type de subdivision de triangulations polygonales est particulièrement bien adapté aux méthodes multirésolution, puisqu'elle conserve les angles des triangles, et peut ainsi être hiérarchisée de manière stable. Les surfaces interpolantes sont polynomiales par morceau de degré 5, quatre

morceaux de surface correspondent à un triangle donné en entrée.

Les premiers travaux (en collaboration avec GP. Bonneau) qui incluent l'interpolant 4-split régulier et ses généralisations au niveau des paramètres de forme (thèse de R. Taleb) se sont pourtant avérées ne pas être exploitables pour la multirésolution (DEA de E. Gilbert) comme nous l'avions souhaité initialement. Seule la méthode suivante d'interpolation irrégulière est appropriée aux applications multirésolution, car elle possède la propriété d'invariance par subdivision. Plus précisément, l'interpolation appliquée à une certaine subdivision de la triangulation donnée en entrée donne le même résultat que l'interpolation appliquée à la triangulation non subdivisée.

Comme première application MR nous avons pu construire un modèle surfacique triangulaire hiérarchique (DEA A. Yvart) généralisant les splines produit tensoriel hiérarchiques introduites par Forsey et Bartels [29]. Contrairement à ces dernières, notre méthode permet de représenter des surfaces de topologie quelconque. Par la suite nous allons avec A. Yvart (en thèse BDI depuis sept. 2001) continuer le développement d'applications MR basées sur cet interpolant. Un de nos projets concerne la reconstruction MR d'objets 3D par une surface lisse. De nombreuses questions intéressantes, comme la paramétrisation d'un ensemble de points non-structurés (par rapport à une triangulation polygonale) vont se poser.

Le thème MR en modélisation géométrique est devenu de manière général un pôle de recherche actuel et futur autour duquel est né un certain nombre de projets de recherche et de collaborations:

- Il s'agit d'abord de la participation depuis Janvier 2000 au projet européen RTN MINGLE (Multiresolution IN Geometric modELing) en tant que copartenaire pour Grenoble. Avec 9 partenaires de 6 pays (SINTEF et Sim en Norvège, Max Planck Institut für Informatik et TU München en Allemagne, Cambridge en Angleterre, Univ. Gène en Italy, Univ. Tel Aviv et Technion en Israël, UJF en France) nous développons et animons des recherches dans divers domaines de la MR en modélisation géométrique. Le but principal, la formation de pré- et post-doctorants européens sur ce sujet, est de notre côté quasiment atteint avec l'encadrement du post-doc M. Bastian (TU Dresden, Allemagne, 7 mois au LMC), du pré-doc J. Mikkelsen (SINTEF, Norvège, 6 mois) et du post-doc M. Hoffmann (Uni Eger, Hongrie, 12 mois) d'avril 2000 jusqu'à août 2002.
- Une thèse CIFRE (Loïc Lefevre, ex-ENSIMAG) en collaboration avec Dassault Systèmes, vient de débiter. Le sujet porte sur les surfaces de subdivision et leurs applications en CAO.

Les deux projets suivants s'articulent autour d'un sujet de recherche plus spécifique: les schémas MR de courbes et surfaces *sous contraintes*:

- Le premier est un projet PLATON (PAI) qui est mené depuis un an avec P. Kaklis de la NTU Athènes sur les outils de design intuitifs en CAGD.
- Le second est le projet IMAG AMOA sous la responsabilité de V. Perrier auquel je participe. Le stage de B. Sauvage (été 2001) sur les courbes MR

sous contraintes (aire, longueur constantes) a apporté des premiers résultats encourageants, incitant à poursuivre ce thème en sujet de DEA ou de thèse en collaboration avec GP. Bonneau et MP. Cani du laboratoire GRAVIR.

Au niveau international un autre projet de recherche et de validation de la recherche vient de naître:

- Il s'agit d'un projet européen IST-RTD (Recherche et Développement), intitulé MUST.USE.IT (Multi Sense Tensor User Interface), déposé avec 5 partenaires (AVS Danemark, Linköping University Suède, University of Kaiserslautern Allemagne, INTECS Italy et INPG France) dans le 6ème programme cadre de la CE. Ce projet comporte trois grands volets: visualisation de champs de tenseurs, visualisation de qualité de surfaces (sous ma responsabilité) et réalité virtuelle. La coordination est prise en charge par la compagnie AVS (Advanced Visual Systems A/S) qui fabrique le logiciel connu sous le même nom.

Les techniques MR proviennent du traitement du signal. Elles ont été développées et appliquées depuis leur introduction en informatique graphique dans des domaines aussi divers que le calcul de radiosité pour le rendu réaliste, la visualisation de données volumiques, la représentation de squelettes pour l'animation de surfaces implicites, ou encore la modélisation et la simplification d'objets 3D polygonaux.

Les méthodes MR pour les surfaces lisses sont en grande majorité basées sur les schémas de subdivision et s'appliquent aux surfaces de topologie arbitraire. Les méthodes interpolantes sont rares, car elles produisent souvent des surfaces souffrant d'imperfections de forme. La plupart des méthodes ne font donc qu'approximer un maillage polygonal agissant comme un polyèdre contrôlant la forme de la surface finale. Elles permettent alors d'obtenir des formes lisses. Par contre, concernant les surfaces paramétriques, il n'existe que des schémas MR produit tensoriel comme les splines hiérarchiques ou les ondelettes B-spline.

Les travaux présentés dans ce mémoire et qui ont abouti à une méthode d'interpolation polynomiale hiérarchique de triangulations polygonales, répondent à ce manque en surfaces interpolantes polynomiales, lisses, de topologie arbitraire, et permettant une modélisation hiérarchique. Nous avons ainsi jeté les bases théoriques et algorithmiques qui vont maintenant constituer l'outil principal pour le développement de différentes applications, comme la reconstruction 3D, ou la modélisation et l'édition de surfaces polynomiales lisses à différents niveaux de détail, à l'aide de techniques MR.

---

## Références

1. Bajaj C., Smoothing polyhedra using implicit algebraic splines, *Computer Graphics* **26** (2) (1992), 79–88.
2. Bajaj C., Visualization techniques for surface analysis, *Advanced Visualization Techniques*, John Wiley, (1999).
3. Böhm W., The de Boor algorithm for triangular splines, in Barnhill R., Böhm W. (eds.): *Surfaces in Computer Aided Geometric Design*, North-Holland (1983), 109–120.
4. Böhm W. and Farin G. and Kahmann J., A survey of curve and surface methods in CAGD, *Computer Aided Geometric Design* **1**, (1986), 1–60.
5. Bonneau G.P., Hahmann S., Nielson G.M., BLaC-Wavelets: a multiresolution analysis with non-nested spaces, *IEEE Proceedings Visualization'96*, (1996), pp. 43–48.
6. Bonneau G.P., Multiresolution analysis on irregular surface meshes, *IEEE Transactions on Visualization and Computer Graphics* **4** (4), (1998), 365–378.
7. Bonneau G.P. and Hagen H., Variational design of rational Bézier curves and surfaces, in *Curves and Surfaces II*, Laurent, Le Méhauté, Schumaker (eds.), (1994), 51–58.
8. Bonneau G.-P., Hahmann St., Polyhedral modeling, *IEEE Vis'00 proceedings*, (2000), 381–387.
9. Bohl H. and Reif U., Degenerate Bézier patches with continuous curvature, *Computer Aided Geometric Design* **14**, (1997), 749–761.
10. Brunet P., Increasing the smoothness of bicubic spline surfaces, In R. Barnhill and W. Böhm (eds.) *Surfaces in CAGD'84*, North-Holland (1985).
11. Brunet G., Hagen H. and Santarelli P., Variational design of curves and surfaces, *Surv. Math. Ind.* **3**, (1993), 1–27.
12. Catmull E., Clark J., Recursively generated B-spline surfaces on arbitrary topological meshes, *Computer Aided Design* **10**, (1978), 350–355.
13. Celniker G., Gossard D., Deformable curve and surface elements for free-form shape design, *ACM Computer Graphics* **25**, (1991), 257–265.

14. Chiyokura H. and Kimura F., Design of solids with free-form surfaces, *Computer Graphics* **17** (13), (1983), 289–298.
15. Davis P., *Circulant Matrices*, Wiley, (1979).
16. Desbrun M., Meyer M., Schröder P., Barr A., Implicit fairing of irregular meshes using diffusion and curvature flow, in proceedings SIGGRAPH'99, ACM, (1999), 317–324.
17. Doo D., Sabin M., Analysis of the behaviour of recursive division surfaces near extraordinary points, *Computer Aided Design* **10**, (1978), 356–360.
18. Du W. H., Etude sur la représentation de surfaces complexes : application à la reconstruction de surface échantillonnées, Thèse, Department Image, TELECOM, Paris, (1988).
19. Du H.W., Schmitt F., On the  $G^1$  continuity of piecewise Bézier surfaces: a review with new results, *CAD* **22** (9), (1990), 556–573.
20. Dyn N., Levin D., Gregory J., A butterfly subdivision scheme for surface interpolation with tension control, *ACM Transactions on Graphics* **9** (2), (1990), 160–169.
21. Eck M., Jaspert R., Automatic fairing point sets, in *Designing Fair Curves and Surfaces*, N. Sapidis (ed.), SIAM (1994), 45–60.
22. Eck M., DeRose T., Duchamp T., Hoppe H., Lounsbery M., Stuetzle W., Multiresolution analysis of arbitrary meshes, in Proceedings of SIGGRAPH'95, ACM New York, (1995), 173–182.
23. Farin G., A construction for visual  $C^1$  continuity of polynomial surface patches, *Computer Graphics and Image Processing* **20** (1982), 272–282.
24. Farin G., Smooth interpolation to scattered 3D data, in *Surfaces in CAGD*, R. Barnhill, W. Boehm (eds.), North Holland Publishing Company, 1983, 43–63.
25. Farin, G., Rein G., Sapidis N., and Worsey A.J., Fairing cubic B-spline curves, *Computer Aided Geometric Design* **4** (1987), 91–103.
26. Farin G., Sapidis N., Curvature and the fairness of curves and surfaces, *IEEE Computer Graphics & Applications* **9** (1989), 53–57.
27. Farin G., *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, New York, 4th edition, (1996).
28. Farin G., Hansford D., *The Essentials of CAGD*, AK Peters, (2000).
29. Forsey D., Bartels R., Hierarchical B-spline refinement, Proceedings of SIGGRAPH'88, ACM New York, (1988), 205–212.
30. Gregory J.A., N-sided surface patches, in *The Mathematics of Surfaces*, J. Gregory (ed.), Clarendon Press, Oxford (1986), 217–232.
31. Greiner G., Variational design and fairing of spline surfaces, Proc. Eurographics 1994, 143–154.
32. Hadenfeld J., Local energy fairing of B-spline surfaces, in *Mathematical Methods for Curves and Surfaces*, Morten Dæhlen, Tom Lyche, Larry L. Schumaker (eds.), Vanderbilt University Press, Nashville & London, (1995), 203–212.

33. Hagen H., Geometric surface patches without twist constraints, *Computer Aided Geometric Design* **3** (1986), 179–184.
34. Hagen H., Schulze G., Automatic smoothing with geometric surface patches, *Computer Aided Geometric Design* **4** (1987), 231–236.
35. Hagen H., Pottmann H., Curvature continuous triangular interpolants, in *Mathematical Methods in Computer Aided Geometric Design* Tom Lyche, Larry L. Schumaker (eds.), Academic Press, New York (1989), 373–384.
36. Hagen H., Santarelli P., Variational design of smooth B-spline surfaces, in *Topics in geometric modeling*, H. Hagen (ed.), SIAM Philadelphia, (1992), 85–94.
37. Hahmann S., Konz S., Fairing bicubic B-spline surfaces using simulated annealing, in *Curves and Surfaces with Applications in CAGD*, A. Le Méhauté, L. L. Schumaker, C. Rabut (eds.), Vanderbilt University Press, (1997), 159–168.
38. Hahmann S., Konz S., Knot-removal surface fairing using search strategies, *Journal of CAD* **30** (2), (1998), 131–138.
39. Hahmann S., Shape improvement of surfaces, *Computing Suppl.* **13**, (1998), 135–152.
40. Hahmann S., Visualization techniques for surface analysis, Chapter 4 in C. Bajaj (ed.): *Advanced Visualization Techniques*, John Wiley, (1999), 49–74.
41. Hahmann S., Bonneau G-P., Triangular  $G^1$  interpolation by 4-splitting domain triangles, *Computer Aided Geometric Design* **17** (2000), 731–757.
42. Hahmann S., Bonneau G-P., Taleb R., Smooth irregular mesh interpolation, in *Curve and Surface Fitting: Saint-Malo 1999* Albert Cohen, Christophe Rabut, and Larry L. Schumaker (eds.), Vanderbilt University Press, Nashville, (2000), 237–246.
43. Hahmann S., Bonneau G-P., Taleb R., Localizing the 4-split method for  $G^1$  free-form surface fitting, *Computing Suppl.* **14**, (2001), 185–198.
44. Hahmann S., Bonneau G.-P., Parametric surfaces over arbitrary triangulations, *IEEE Transactions on Visualization and Computer Graphics*, à paraître (2002).
45. Hahmann S., Bonneau G.-P., Yvart A., Hierarchical triangular surfaces, *Rapport de recherche LMC-IMAG*, en préparartion.
46. Hamann B., A data reduction scheme for triangulated surfaces, *Computer Aided Geometric Design* **11**, (1994), 197–214.
47. Herron G., Smooth closed surfaces with discrete triangular interpolants, *Computer Aided Geometric Design* **2** (1985), 297–306.
48. Hoschek J. and Lasser D., *Fundamentals of Computer Aided Geometric Design*, A.K. Peters, (1993).
49. Hoppe H., DeRose T., Duchamp T., McDonald J., Stuetzle W., Surface reconstruction from unorganized points, in *proceedings SIGGRAPH'92*, ACM New York, (1992), 71–78.

50. Hoppe H., DeRose T., Duchamp T., McDonald J., Stuetzle W., Mesh optimization, in proceedings SIGGRAPH'93, ACM New York, (1993), 19–26.
51. Hoppe H., Progressive meshes, in proceedings SIGGRAPH'96, ACM New York, (1996), 99-108.
52. Jensen T., Assembling triangular and rectangular patches and multivariate splines, in *Geometric Modeling: Algorithms and new Trends*, G. Farin (ed.), SIAM (1987), 203–220.
53. Kirkpatrick S., Gelatt C.D., Vecchi M.P., *Science* **220** (1983), 671.
54. Kirkpatrick S., Optimization by simulated annealing, *J. Stat. Phys.* **34** (5/6), (1984), 975-987.
55. Kjellander, J. A., Smoothing of cubic parametric splines, *Computer Aided Design* **15** (1983), 175–179.
56. Kobbelt L., Discrete fairing, in proceedings of the Seventh IMA Conference on the Mathematics of Surfaces, (1997), 101–131.
57. Kjellander, J. A., Smoothing of bicubic parametric surfaces, *Computer Aided Design* **15** (1983), 289–293.
58. Levin A., Combined subdivision schemes for the design of surfaces satisfying boundary conditions, *Computer Aided Geometric Design* **16** (1999), 345-354.
59. Litke N., Levin A., Schröder P., Trimming for subdivision surfaces, *Computer Aided Geometric Design* **18** (2001), 463–481.
60. Loop C., Smooth subdivision surfaces based on triangle meshes, Master's Thesis, University of Utah, Department of Mathematics, (1987).
61. Loop C., De Rose T., A multisided generalization of Bézier surfaces, *ACM Transactions on Graphics* **8**, (1989), 204-234.
62. Loop C., A  $G^1$  triangular spline surface of arbitrary topological type, *Computer Aided Geometric Design* **11** (1994), 303–330.
63. Lott N.J., Pullin D.I., Method for fairing B-spline surfaces, *Computer Aided Design* **20** (1988), 597–604.
64. Lounsbery M., Multiresolution analysis for surfaces of arbitrary topological type, Ph.D. thesis, University of Washington, (1994).
65. Lyche T., Morken K., Knot removal for parametric B-spline curves and surfaces, *Computer Aided Geometric Design* **4** (1987), 217–230.
66. Mallat S.G., A theory for multiresolution signal decomposition: the wavelet representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **11** (7), (1989), 674-693.
67. Mann S., Surface approximation using geometric Hermite patches, Ph.D. dissertation, University of Washington, (1992).
68. Mann S., Loop C., Lounsbery M., Meyers D., Painter J., DeRose T., Sloan K., A survey of parametric scattered data fitting using triangular interpolants, in *Curve and Surface Design*, H. Hagen (ed.), SIAM (1992), 145–172.

69. Moreton H.P. and Séquin C.H., Functional optimisation for fair surface design, *Computer Graphics* **26** (2), (1992), 167–176.
70. Neamtu M., Pluger P., Degenerate polynomial patches of degree 4 and 5 used for geometrically smooth interpolation in  $\mathbb{R}^3$ , *Computer Aided Geometric Design* **11** (1994), 451–474.
71. Nielson G., Some Piecewise Polynomial Alternatives to Splines Under Tension, in *Computer Aided Geometric Design*, R. Barnhill, Riesenfeld (eds.), (1974).
72. Nielson G., A method for interpolating Scattered Data Based Upon a Minimum Norm Network, *Mathematics of Computation* **40**, (1983), 253–271.
73. Nielson G., A transfinite, visually continuous, triangular interpolant, in *Geometric Modeling: Algorithms and new Trends*, G. Farin (ed.), SIAM (1987), 235–246.
74. Nowacki H., Reese D., Design and fairing of ship surfaces, in *Computer Aided Geometric Design*, R. Barnhill and W. Böhm (eds.), North-Holland, (1983), 121–134.
75. Peters J., Local smooth surface interpolation: a classification, *Computer Aided Geometric Design* **7**, (1990), 191–195.
76. Peters J., Smooth interpolation of a mesh of curves, *Constructive Approximation* **7** (1991), 221–246.
77. Peters J., Parametrizing singularly to enclose vertices by a smooth parametric surface, in *Proceedings of Graphics Interface '91*, S. MacKay, E.M. Kidd (eds.), Canadian ManComputer Communications Society, Kurt Akeley, (1991), 1–7.
78. Piegl L., On NURBS: a Survey, *Computer Graphics and Applications* **11** (1), (1991), 55–71.
79. Pilcher D., Smooth Parametric Surfaces, in *Computer Aided Geometric Design*, R. Barnhill, Riesenfeld (eds.), (1974).
80. Piper B.R., Visually smooth interpolation with triangular Bézier patches, in *Geometric Modeling: Algorithms and new Trends*, G. Farin (ed.), SIAM (1987), 221–233.
81. Rando T., Roulier A., Designing faired parametric surfaces, *Computer Aided Design* **23** (1991), 492–497.
82. Richter M.M., *Prinzipien der künstlichen Intelligenz*, B. G. Teubner, Stuttgart 1989.
83. Roulier J. and Rando T., Measures of Fairness for Curves and Surfaces, in *Designing Fair Curves and Surfaces*, S. Sapidis.(eds), SIAM (1994).
84. Sabin M., Non-rectangular surface patches suitable for inclusion in a B-spline surface, in *tex Hagen P., (ed.), Proceedings Eurographics'83*, North-Holland (1983), 57–69.
85. Sapidis, N., Farin G., Automatic fairing algorithm for B-spline curves, *Computer Aided Design* **22** (1990), 121–129.



86. Sarraga R.,  $G^1$  interpolation of generally unrestricted cubic Bézier curves, *Computer Aided Geometric Design* **4** (1987), 23–39.
87. Schröder P., Sweldens W., Spherical Wavelets: Efficiently Representing Functions on the Sphere, in proceedings SIGGRAPH'95, ACM, (1995), 161-172.
88. Schroeder W.J., Zarge Z.A., Lorensen W.E., Decimation of triangle meshes, in proceedings SIGGRAPH'92, ACM, (1992), 65–70.
89. Schumaker L.L., Computing optimal triangulations using simulated annealing, *Computer Aided Geometric Design* **10** (1993), 329–345.
90. Schweikert D., An interpolation curve using a spline in tension, *Journal of Math. and Physics* **45**, (1988), 312–317.
91. Selesnick S., Local invariants and twist vectors in CAGD, *Computer Graphics and Image Processing* **17**, (1981), 145–160.
92. Shirman L.A., Séquin C.H., Local surface interpolation with Bézier patches, *Computer Aided Geometric Design* **4** (1987), 279–295.
93. Shirman L.A., Séquin C.H., Local surface interpolation with Bézier patches: errata and improvements, *Computer Aided Geometric Design* **8** (1991), 217–221.
94. Stam J., Exact evaluation of Cutmull-Clark subdivision surfaces at arbitrary parameter values, in proceedings SIGGRAPH'98, ACM, (1998), 395–404.
95. Stollnitz E., DeRose T., Salesin D., *Wavelets for Computer Graphics: a primer, Part 1*, IEEE Computer Graphics and Applications **15**, (1995).
96. Stollnitz E., DeRose T., Salesin D., *Wavelets for Computer Graphics: Theory and Applications*, Morgan Kaufmann Publishers, San Francisco, (1996).
97. Taleb R., Design géométrique de surfaces de topologie arbitraire, Thèse de doctorat, Université Joseph Fourier, Grenoble, 2001.
98. Taubin G, A signal processing approach to fair surface design, in proceedings SIGGRAPH'95, ACM, (1995), 351–358.
99. Turk G., Re-tiling polygonal surfaces, in proceedings SIGGRAPH'92, ACM, (1992), 55–64.
100. Weimer H., Warren J., Subdivision schemes for thin plate splines, *Computer Graphics Forum* **17**, (1998), 303–314.
101. Welch W., Witkin A., Free-form shape design using triangulated surfaces, in proceedings SIGGRAPH'94, ACM, (1994), 247–256.
102. Van Wijk J.J., Bicubic patches for approximating non-rectangular control meshes, *Computer Aided Geometric Design* **3** (1986), 1–13.
103. Yvart A., Modélisation hiérarchique de surfaces  $G^1$ , Mémoire de DEA Mathématiques Appliquées, Grenoble, Juin (2001).
104. Zorin D., Schröder P., Sweldens W., Interpolating subdivision for meshes with arbitrary topology, in proceedings SIGGRAPH'96, ACM, (1996), 189–102.