



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

Bases de fonctions sur les variétés

THÈSE

présentée et soutenue publiquement le 10 juillet 2008

pour l'obtention du

Doctorat de l'Institut National Polytechnique de Lorraine
(spécialité informatique)

par

Bruno Vallet

Composition du jury

<i>Président :</i>	Karl Tombre	INRIA Nancy - Grand Est
<i>Rapporteurs :</i>	Claude Puech Leif Kobbelt	Université Paris-Sud RWTH Aachen
<i>Examineur :</i>	Jean-Daniel Boissonat	INRIA Sophia Antipolis - Méditerranée
<i>Directeurs de thèse :</i>	Jean Claude Paul Bruno Lévy	Tsinghua University INRIA Nancy - Grand Est

Mis en page avec la classe thloria.

Résumé

Les bases de fonctions sont des outils indispensables de la géométrie numérique puisqu'ils permettent de représenter des fonctions comme des vecteurs, c'est à dire d'appliquer les outils de l'algèbre linéaire à l'analyse fonctionnelle.

Dans cette thèse, nous présentons plusieurs constructions de bases de fonctions sur des surfaces pour la géométrie numérique. Nous commençons par présenter les bases de fonctions usuelles des éléments finis et du calcul extérieur discret, leur théorie et leurs limites. Nous étudions ensuite le Laplacien et sa discrétisation, ce qui nous permettra de construire une base de fonctions particulière : les fonctions propres de l'opérateur de Laplace-Beltrami, ou harmoniques variétés. Celles-ci permettent de généraliser la transformée de Fourier et le filtrage spectral aux fonctions définies sur des surfaces. Nous présentons ensuite des applications de cette base de fonction à la géométrie numérique. En particulier, nous montrons qu'une fois calculée, cette base de fonction permet de filtrer la géométrie en temps interactif.

Pour pouvoir définir des bases de fonctions de façon plus indépendante du maillage de la surface, nous nous intéressons ensuite aux paramétrisations globales, et en particulier aux champs de directions à symétries qui permettent de les définir. Ainsi, dans la dernière partie, nous étudions ces champs de directions à symétries, et en particulier leur géométrie et leur topologie. Nous donnons alors des outils pour les construire, les manipuler et les visualiser.

Mots-clés: Variétés, géométrie différentielle, espaces de Hilbert, éléments finis, calcul extérieur, champs de directions, paramétrisation globale.

Abstract

Function bases are fundamental objects in geometry processing as they allow to represent functions as vectors, that is to apply tools from linear algebra to functional analysis.

In this thesis, we present various constructions of useful functions bases for geometry processing. We start by presenting usual function bases, their theory and limits. We then study the Laplacian operator and its discretization, and use it to define a particular function basis : Laplacian eigenfunctions or Manifold harmonics. The Manifold Harmonics form a function basis that allows to generalize the Fourier transform and spectral filtering on a surface. We present some applications and extensions of this basis for geometry processing.

To define function bases in a mesh-independent manner, we need to build a global parameterization, and especially the direction fields required to define them. Thus, in the last part of this thesis we study N -symmetry direction fields on surfaces, and in particular their geometry and topology. We then give tools to build, edit, control and visualize them.

Keywords: Manifold, differential geometry, Hilbert space, finite elements, exterior calculus, direction field, global parameterization.

Remerciements

Je commencerai évidemment par remercier Bruno Lévy pour m'avoir convaincu de venir faire ma thèse au sein de son équipe-projet Alice, et pour m'avoir encadré en faisant preuve d'une grande originalité dans les choix des sujets qu'il m'a fait aborder.

Je souhaite aussi remercier tous les membres du jury pour avoir pris du temps pour lire et commenter cette thèse, et pour venir, souvent de loin, assister à ma soutenance.

Je tiens à remercier aussi mon directeur de thèse Jean Claude Paul, pour son immense gentillesse tout d'abord, mais aussi pour avoir su orienter mes choix professionnels de façon judicieuse à un moment clef.

Cette thèse doit énormément à Nicolas Ray qui m'a soutenu et supporté pendant ces trois années. Je le remercie d'avoir fait preuve d'autant de créativité et de pragmatisme dans nos discussions, et de m'avoir aiguillé dans des directions de recherche toujours passionnantes.

Les conseils avisés et la qualité scientifique de Laurent Alonso m'ont été d'un grand secours tout au long de ma thèse.

Tous mes collègues du projet ALICE ont à leur façon contribué à cette thèse, que ce soit par leur support scientifique, technique ou moral.

Je remercie en particulier Wan-Chiu Li et Grégory Lecot pour leur bonne humeur communicative, Rodrigo Toledo, Laurent Castanié, Luc Buatois, Cedric Borghese, Nicolas Stoibert, Olivier Genevaux, Mathieu Chavent et Aurélien Martinet pour leur disponibilité et leur gentillesse.

Je remercie aussi Vincent, Ahmed, Jérôme, Cécile et Nicolas pour avoir amené leur gentillesse et leur dynamisme au service du projet ALICE.

Un grand merci à Isabelle Herlich et Nadine Beurné dont les compétences ont allégé mon fardeau administratif.

Enfin, et par dessus tout, mon soutien le plus précieux pendant ces trois années a été l'amour de ceux que je chéris le plus et qui se reconnaîtront si ils lisent ces lignes.

*Je dédie cette thèse à mon père,
qui m'a appris son amour pour la Science.*

Table des matières

Notations	xiii
Introduction générale	1
Partie I Calcul géométrique	7
Chapitre 1	
Le problème de la représentation géométrique	
1.1 De la géométrie	10
1.1.1 Les objets géométriques	10
1.1.2 Les variétés	11
1.1.3 Les représentations géométriques	12
1.2 Variétés affines	13
1.2.1 Géométrie projective	13
1.2.2 Le plan	14
1.2.3 Définition implicite	14
1.2.4 Indépendance géométrique	14
1.2.5 Coordonnées barycentriques	14
1.3 Les complexes simpliciaux	15
Chapitre 2	
Méthode des Éléments finis	
2.1 Théorie des éléments finis	18
2.1.1 Espaces de Hilbert	18
2.1.2 Espaces de Hilbert fonctionnels	19
2.1.3 Théorie de Lax-Milgram	21
2.1.4 Approximation interne	21
2.1.5 Galerkin et les éléments finis	22

2.2	Bases de fonctions interpolantes	23
2.2.1	Éléments P_n	24
2.2.2	Interpolation bilinéaire	28
2.3	Bases mobiles	29
2.3.1	Problème	30
2.3.2	Interpolation P_1 des fonctions convexes en 1D	30
2.3.3	Approximation par fonctions génériques en 1D	31
2.3.4	Conclusion	31

Chapitre 3

Le calcul extérieur continu et son équivalent discret

3.1	Calcul extérieur dans le cadre continu	34
3.1.1	Tenseurs	34
3.1.2	k -chaines et k -formes alternées	35
3.1.3	Produit extérieur	36
3.1.4	k -chaines et k -formes différentielles	36
3.1.5	Dérivée extérieure	37
3.1.6	Intégration et dualité	39
3.1.7	Étoile de Hodge	41
3.1.8	Produit interne	41
3.1.9	Conclusion	42
3.2	Le calcul extérieur discret	42
3.2.1	Formes, chaines et intégration discrètes	42
3.2.2	Orientation	43
3.2.3	Dérivée extérieure et opérateur de bord	44
3.2.4	(Co)homologie	44
3.2.5	Complexe dual	45
3.2.6	Étoile de Hodge	46
3.2.7	Produit interne	47
3.2.8	Applications de De Rham et Whitney	48
3.2.9	Calcul extérieur et dualités	48

Partie II Harmoniques variétés

Chapitre 4

Le Laplacien

4.1	Le Laplacien continu	56
4.1.1	Laplacien dans \mathbb{R}^d	56
4.1.2	Laplacien sur une variété	57
4.1.3	Laplacien d'une k -forme	57
4.1.4	Conclusion	58
4.2	Etat de l'art	59
4.3	Laplaciens discrets et discrétisés sur un maillage	60
4.3.1	Forme générale	60
4.3.2	Formulation en éléments finis	61
4.3.3	Retrouver la symétrie	62
4.3.4	Laplacien d'interpolation	62
4.3.5	Conditions aux bords	66
4.3.6	Laplacien du calcul extérieur discret	66
4.3.7	Localisation du Laplacien P_1 par le graphe mixte	67
4.3.8	Limites d'un Laplacien discret	70
4.4	Conclusion	71

Chapitre 5

Harmoniques variétés

5.1	Previous Work	75
5.2	Spectral Analysis on Manifolds	76
5.2.1	Fourier Analysis	76
5.2.2	Intern approximation for Laplacian eigenfunctions	77
5.2.3	Manifold Harmonics Basis (MHB)	77
5.2.4	Interpretation of the MHB	78
5.3	The Manifold Harmonic Transform (MHT)	79
5.3.1	MHT and inverse MHT	79
5.3.2	Spectral filtering	80
5.4	Numerical Solution Mechanism	81
5.4.1	Band-by-band computation of the MHB	82
5.4.2	Limited-memory spectral filtering	83
5.5	Results	83
5.6	Conclusion	85
5.6.1	Discussion	85
5.6.2	Toward more localized basis functions	87

Partie III Champs de directions

91

Chapitre 6

Les champs de directions

6.1 Applications	98
6.1.1 Rendu non photoréaliste	98
6.1.2 Synthèse de texture	98
6.1.3 Paramétrisation et remaillage quad	99
6.1.4 Conclusion	100
6.2 Caractéristiques des outils de manipulation de champs de directions	100
6.3 Etat de l'art	102
6.3.1 Lissage géométrique des champs de directions (GDS)	104
6.3.2 Création de champs de vecteurs tangents (TVFD)	105
6.3.3 N-rosy	106
6.4 Conclusion	108

Chapitre 7

Les champs de directions à symétries et leur discrétisation

7.1 N -symmetry direction fields on surfaces with boundary	111
7.1.1 Surfaces with boundary	112
7.1.2 Curves and Cycles on \mathcal{S}	112
7.1.3 Direction field	114
7.1.4 Curvature	115
7.2 Turning numbers and their fundamental properties	115
7.2.1 Definition	115
7.2.2 Boundary property	116
7.2.3 Topological equivalence	119
7.3 Singularities	119
7.3.1 2D vector field Singularity Classification	120
7.3.2 2D N -symmetry direction field Singularities	121
7.3.3 N -symmetry direction field Singularities on surfaces	122
7.3.4 Poincaré Hopf theorem	122
7.4 Discretization of N -symmetry direction fields	123
7.4.1 N -symmetry directions on facets	124
7.4.2 Curvature and period jumps	125
7.4.3 Turning numbers	126
7.4.4 Singularity index	127

7.4.5	Converting vector fields into our representation	128
7.4.6	Conclusion	129

Chapitre 8

Visualisation

8.1	Previous work	132
8.1.1	Representing Higher-order Singularities in Vector Fields	132
8.1.2	Visualization of Vector Fields on Surfaces	136
8.2	Interpolation	136
8.2.1	Step 1 : 0D	136
8.2.2	Step 2 : 1D	137
8.2.3	Step 3 : 2D	138
8.3	Corrected interpolation	139
8.4	The GPU-based Visualization Algorithm	141
8.4.1	Pass 1 : Depth Value for Geometric Discontinuities	141
8.4.2	Pass 2 : Object to Image Space Projection	141
8.4.3	Pass 3 : LIC in the Image Space	142
8.5	Results, Discussions and Conclusions	143

Chapitre 9

Manipulation de champs de directions avec contrôle topologique

9.1	Zippping algorithm : Constraining the topology	148
9.1.1	Problem setting	148
9.1.2	Edge classification	149
9.1.3	Topological issues	150
9.2	Constrained direction field smoothing	152
9.2.1	Problem formulation	152
9.2.2	Algorithm	152
9.3	Results and conclusion	153
9.3.1	Results	153
9.3.2	Conclusion	154

Chapitre 10

Manipulation de champs de directions avec contrôle de l'influence géométrique

10.1	Problem statement	159
10.2	Geometric smoothing algorithm	160
10.3	Canceling the influence of the geometry	160

10.3.1	Topological disks	161
10.3.2	Extension to surfaces of arbitrary genus	162
10.4	Filtering the geometric influence	162
10.4.1	Filtering K	163
10.4.2	Computing singularity indexes	163
10.4.3	Editing the direction field topology	163
10.5	Results and Applications	164
10.6	Conclusion	166

Conclusion générale **169**

Annexe A

Algèbre linéaire pour la géométrie

A.1	Algèbre linéaire géométrique	173
A.1.1	Inverse d'une matrice	174
A.1.2	Application aux calculs d'intersection	174
A.1.3	Pseudo-inverse	175
A.1.4	Factorisations de matrices	175
A.1.5	Orthonormalisation de Gram-Schmidt	176
A.2	Systèmes linéaires contraints	176
A.2.1	Minimisation de formes quadratiques en variables réelles	177
A.2.2	Minimisation de formes quadratiques en variables entières	177
A.2.3	Minimisation de formes quadratiques en variables réelles et entières	178
A.3	Minimisation contrainte	178
A.3.1	Blocage de variable	178
A.3.2	Contraintes linéaires	178
A.3.3	Contraintes quadratiques	179

Annexes

Publications **181**

Bibliographie **183**

Notations

Nous fournissons ici quelques tables des notations utilisées dans cette thèse :

- La Table 1 présente les notations générales utilisées dans toute la thèse.
- La Table 2 présente les notations utilisées pour travailler sur les complexes simpliciaux.
- La Table 3 instancie ces notations pour les surfaces triangulées.
- La Table 4 présente les notations utilisées en modélisation par éléments finis (Chapitre 2) et en calcul extérieur discret (Chapitre 3). La plupart seront utilisées dans toute la thèse puisqu'elle se base sur ces formalismes.
- La Table 5 présente des notations pour les champs de directions. Elles seront utilisées dans tous les chapitres de la Partie III.

Nous espérons que cette présentation des notations aidera le lecteur à s'y retrouver. Les notations qui n'apparaissent pas dans ces tables sont seulement les notations utilisées de façon très locale, auquel cas la définition de la notation sera proche de là où elle sera utilisée. Enfin, nous résumons ici les conventions utilisées dans la thèse :

- Les vecteurs et les points dans \mathbb{R}^d sont écrits en gras (\mathbf{x}, \mathbf{y}).
- Par extension, les vecteurs d'inconnues dans les systèmes linéaires (\mathbf{x}), les lignes et colonnes de matrices ($\mathbf{a}_i, \mathbf{a}^j$) seront aussi en gras. La notation \vec{d} est réservée pour les directions (assimilables à des vecteurs unitaires) éventuellement à N -symétries.
- Nous supposons toujours que nous disposons d'une numérotation ordonnée pour les sommets, et nous assimilons les sommets avec leur index : $i, j, k, l, v_1, \dots, v_{n_v}$ réfèrent à la fois à des indices entiers et à des sommets. i, j, k, j sont utilisés pour nommer les sommets lorsqu'ils sont peu nombreux (quand on travaille sur un triangle ou une paire de triangles par exemple), alors que la notation v_0, \dots, v_k

E, \mathbb{R}^d	Espace Euclidien, espace vectoriel de d
$\mathcal{S} \subset \mathbb{R}^d$	Surface/variété plongée dans \mathbb{R}^d
$T\mathcal{S}$	Fibré tangent de \mathcal{S}
\mathbf{e}_i	i -ème vecteur de base de \mathbb{R}^d
$\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$	vecteurs
$\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{d+1}$	vecteurs projectifs
$x_i, y_i \in \mathbb{R}$	coefficients des vecteurs
$\mathbf{x} \cdot \mathbf{y} = \mathbf{x}^t \mathbf{y}$	produit scalaire
$S^d = \{\mathbf{x} \in \mathbb{R}^{d+1} \mathbf{x}^2 = 1\}$	sphère unité dans \mathbb{R}^{d+1}
A, A_i^j	matrice et ses coefficients
$\mathbf{a}_i, \mathbf{a}^j$	i -ème ligne, j -ème colonne de A
T_n^m	tenseur (n, m)

TAB. 1 – Notations générales

M	surface triangulée
$M = \{\mathcal{V}, \mathcal{E}, \mathcal{F}\}$	ensemble des sommets, arêtes et faces de M
$n_v = \mathcal{V} , n_e = \mathcal{E} , n_f = \mathcal{F} $	nombre de sommets, arêtes et faces de M
$[M] \subset \mathbb{R}^d$	plongement de M dans \mathbb{R}^d
$i, j, k, l, v_0, \dots, v_k$	sommets
$s = i, ij, ijk$	simplexes (sommets, arêtes et faces)
$dim(s)$	dimension de s
$s_k \prec s_l$	s_k est une face de s_l
$[s] \subset \mathbb{R}^d$	plongement d'un simplexe orienté dans \mathbb{R}^d
$ s \in \wedge^k(\mathbb{R}^d)$	chaîne correspondant à un k -simplexe
$ s \in \mathbb{R}$	mesure d'un simplexe
$\mathbf{g}(s)$	barycentre de $[s]$
$St_l(s)$	l -étoile de s

TAB. 2 – Notations pour les complexes simpliciaux

$s =$	i	ij	ijk
$dim(s)$	0	1	2
$[s]$	point	arête	triangle
$ s $	point	vecteur	élément d'aire orienté
$ s $	± 1	longueur	aire
$\mathbf{g}(s)$	point	milieu	barycentre
$St_1(s)$	arêtes iv	ij	\emptyset
$St_2(s)$	triangles iv_1v_2	triangles ijv	ijk

TAB. 3 – Instantiation des notations pour les surfaces triangulées

est plus générique et sert pour itérer sur les sommets d'une surface triangulée ou d'un ensemble de cardinal non fixé. Notez que les sommets sont des objets topologiques impliqués dans la définition des complexes simpliciaux (comme les sommets d'un graphe), alors que leurs plongements $[i], [j], \dots$ sont des points de l'espace \mathbb{R}^d , ce qui réfère à de la géométrie. Par exemple, deux sommets différents d'une surface triangulée peuvent avoir la même position dans l'espace, auquel cas on aura $i \neq j$ mais $[i] = [j]$.

- Les quantités définies sur les sommets i , arêtes ij et triangles ijk sont indicées par i, ij et ijk . Par convention elles seront nulles si $i, ij, ijk \notin M$.
- Une quantité relative à un sommet i dans le triangle ijk (ex : angle) sera indexée par i, jk (ex : $\beta_{i,jk} = \angle(|ij|, |ik|)$).

$\Omega = [M], \mathcal{S}, \mathbb{R}^d \dots$ $\mathbf{x}, \mathbf{y} \in \Omega$ $f(\mathbf{x}), g(\mathbf{x})$ V V_n $L^2(\Omega)$ $C_c^k(\Omega)$ $H^1, H_0^1, H(\Omega)$ $\langle f, g \rangle_V$ $\ f\ _V$ $D(f)$ $L(g), a(f, g)$ $r_n(f)$ $\lambda_{i,(jk)}(\mathbf{x})$ $\Phi^i(\mathbf{x})$ $\bar{\Phi}^i(\mathbf{x})$ $\Phi = (\Phi^1, \dots, \Phi^n), \bar{\Phi}$ $P_n^i(\mathbf{x})$ $B_{ij} : [0, 1]^2 \rightarrow \mathbb{R}$ $H^k(\mathbf{x}) = \Phi(\mathbf{x})\mathbf{h}^k$	espace de base générique points de Ω fonctions génériques ($\Omega \rightarrow \mathbb{R}$) espace de Hilbert générique suite d'espaces de Hilbert (de dimension n) fonctions de carré intégrable sur Ω fonctions k fois dérivables à support compact sur Ω espaces de Sobolev produit interne pour les fonctions dans V norme pour les fonctions dans V opérateur différentiel générique dans un espace de Sobolev forme linéaire, bilinéaire projection de $f \in V$ sur V_n coordonnées barycentriques de \mathbf{x} relatives à i (dans ijk) i^{th} base de fonction générique i^{th} base de fonction normalisée vecteurs lignes correspondants i -ème fonction de la base P_n fonction de la base bilinéaire relative au point $[i, j] \in \{0, 1\}^2$ k^{th} harmonique variété
$\alpha_k, \beta_k \in \wedge_k(\Omega)$ $\alpha^k, \beta^k \in \wedge^k(\Omega)$ $\omega_k, \eta_k \in \wedge_k(M)$ $\omega^k, \eta^k \in \wedge^k(M)$ $R^M : \wedge^k([M]) \rightarrow \wedge^k(M)$ $W^M : \wedge^k(M) \rightarrow \wedge^k([M])$	k -chaines différentielles sur Ω k -formes différentielles sur Ω k -chaines discrètes sur M k -formes discrètes sur M application de <i>de Rham</i> application de <i>Whitney</i>
\wedge $b : \wedge^k(\Omega) \rightarrow \wedge_k(\Omega)$ $\sharp : \wedge_k(\Omega) \rightarrow \wedge^k(\Omega)$ $d_k : \wedge^k(\Omega) \rightarrow \wedge^{k+1}(\Omega)$ $\partial_k : \wedge_{k+1}(\Omega) \rightarrow \wedge_k(\Omega)$ $\star_k : \wedge^k(\Omega) \rightarrow \wedge^{d-k}(\Omega)$ $\Delta_k : \wedge^k(\Omega) \rightarrow \wedge^k(\Omega)$	produit extérieur opérateur bémol opérateur dièse dérivée extérieure (d'une k -forme) bord (d'une k -chaîne) étoile de Hodge (d'une k -forme) Laplacien (d'une k -forme)

TAB. 4 – Notations pour les éléments finis et le calcul extérieur discret

$g(\Omega), b(\Omega)$ $\chi(\Omega) = 2 - 2g(\Omega) - b(\Omega)$ \equiv_l, \equiv_l $\vec{d} \in \mathcal{D}_N(\mathcal{S})$ \mathbf{n} $\gamma \in \mathcal{C}(\mathcal{S})$ $\mathbf{t}_\gamma, \mathbf{n}_\gamma,$ ∂ $S \subset \mathcal{S}$ $H(\mathcal{S}) = \{\gamma_i^H\}$ $R(\vec{d}, \theta)$ $\angle(\mathbf{x}, \mathbf{y})$ $\kappa_{\mathcal{S}}(\mathbf{t}_\gamma)$ κ_γ $\kappa_{\vec{d}}(\mathbf{t}_\gamma)$ $T_{\vec{d}}(\gamma)$ $\mathbf{p}_i \in \mathcal{S}$ $I_{\vec{d}}(\mathbf{p}_i)$ $\Omega(\mathbf{p}_i) \subset \mathcal{S}$	<p>genre, nombre de bords de $\Omega = \mathcal{S}, [M]$ caractéristique d'Euler de Ω équivalence homotopique, homologique champs de direction à N symétries sur \mathcal{S} normale de la surface cycle sur \mathcal{S} cycle tangent, conormale opérateur de bord sous variété de \mathcal{S} de dimension 2 base d'homologie de \mathcal{S} rotation d'angle θ dans $T\mathcal{S}$ angle entre \mathbf{x} et \mathbf{y} courbure de \mathcal{S} dans la direction \mathbf{t}_γ courbure de γ dans $T\mathcal{S}$ courbure de \vec{d} dans la direction \mathbf{t}_γ nombre de tours de \vec{d} le long de γ singularité index de la singularité du champs \vec{d} petit voisinage d'une singularité</p>
$\vec{d} \in \mathcal{D}_N([M])$ $\mathbf{r}(f)$ $\kappa_1(e), \kappa_1^r(e)$ $\theta_0(f)$ $p_1(e)$ $I_2(v)$ $I_2^c(v), I_2^r(v)$ $K_2(v)$ $K_2^*(v) = K_2/ v^* $	<p>champs de direction à N symétries sur $[M]$ vecteur de référence sur la face f 1-formes de courbure de \vec{d} et \mathbf{r} angle entre $\vec{d}(f)$ et $\mathbf{r}(f)$ saut de période index au sommet v index contraint, index de \mathbf{r} courbure Gaussienne intégrée (défaut angulaire) courbure Gaussienne moyenne</p>

TAB. 5 – Notations pour les champs de directions

Introduction générale

Contexte et objectifs

De nombreux domaines scientifiques nécessitent de manipuler des objets géométriques. Ces objets peuvent être des solides physiques en mécanique, des formes de molécules en biologie, des couches souterraines en géologie... Il existe de nombreuses façons de modéliser ces objets géométriques : les nuages de points, les surfaces triangulées, les *splines*, les surfaces implicites... Heureusement, la plupart de ces objets peuvent être formalisés par un concept mathématique appelé *variété*. Une variété est un objet à n dimensions plongé dans un espace à $d \geq n$ dimensions. Une variété ressemble à \mathbb{R}^n localement (cf Chapitre 1 pour une définition plus formelle). Comme nous nous intéressons à des objets de l'espace réel, nos variétés seront plongées dans \mathbb{R}^3 , et peuvent donc être de quatre types :

- les 0-variétés sont des points
- les 1-variétés sont des courbes
- les 2-variétés sont des surfaces
- les 3-variétés sont des volumes

Dans cette thèse, on s'intéressera surtout aux 2-variétés ou surfaces (qui ressemblent à \mathbb{R}^2 localement) plongées dans \mathbb{R}^3 . Cependant, on a choisi le terme *variété* dans le titre de cette thèse puisqu'il ne fait pas seulement référence à un objet abstrait mais aussi à tout un formalisme mathématique que nous utiliserons abondamment, et qui s'applique aux variétés quelle que soit leurs dimensions.

En informatique graphique, la communauté de la géométrie numérique se base fortement sur ce formalisme mathématique, et s'est attachée à définir des outils pour manipuler ces variétés :

- trouver des représentations pour les variétés : nuages de points, maillages, splines, surfaces de subdivisions...
- convertir entre les représentations : remaillage, reconstruction...
- modifier les variétés : filtrage, déformation, segmentation...
- attacher des objets aux variétés : plaquage de texture, paramétrisation...
- comprendre les variétés : classification, reconnaissance de formes...

La plupart de ces outils nécessitent de définir des fonctions sur les variétés. En conséquence, la résolution d'un problème de géométrie numérique se réduit souvent à trouver des fonctions ayant certaines propriétés, comme minimisant une énergie ou vérifiant une équation aux dérivées partielles. Par exemple, on peut lisser une surface en résolvant l'équation de la chaleur sur cette surface. La paramétrisation de surface s'appuie également sur la définition d'une fonction de la surface vers \mathbb{R}^2 qui vérifiera des équations harmoniques ou conformes par exemple. Cette thèse porte sur les bases fonctions sur les variétés, qui permettent de définir de telles fonctions de façon très générale.

Pourquoi étudions nous les bases de fonctions ? L'utilisation de bases de fonction est un moyen général, simple et rigoureux de définir des fonctions sur les variétés par l'intermédiaire d'un nombre fini de variables. Étant donné une base de fonctions $\{\Phi^i\}$ sur une surface \mathcal{S} , nous pouvons décrire toutes les

fonctions de la forme $\sum_i f_i \Phi^i$ par les seuls coefficients f_i . Par rapport à discrétiser une fonction par ses valeurs en un nombre fini de points de la variété, l'utilisation de bases de fonctions a l'avantage de définir la fonction partout sur la variété. En particulier, une fonction connue par ses valeurs sur les sommets d'une surface triangulée peut être interpolée linéairement en l'exprimant dans la base de fonctions d'interpolation P_1 (cela sera expliqué précisément au Chapitre 2). Ainsi les bases de fonctions nous permettent d'utiliser tous les outils de l'analyse fonctionnelle, et en particulier les méthodes d'éléments finis.

Comment va-t-on définir des bases de fonctions sur les variétés ? Nous allons commencer par présenter les notions théoriques nécessaires à la définition des bases de fonctions sur les variétés au Chapitre 1. On donnera alors les définitions et les outils pour travailler avec les bases de fonctions dans deux cadres populaires : les méthodes aux éléments finis (Chapitre 2), et le calcul extérieur discret (Chapitre 3). Dans ces deux chapitres, on décrira les bases de fonctions usuelles et on discutera leurs propriétés.

Pourquoi construire de nouvelles bases de fonctions ? La géométrie numérique utilise principalement les bases de fonctions d'interpolation pour lesquelles les coefficients dans la base correspondent à la valeur de la fonction sur des "points de contrôle". Ces bases de fonctions sont très dépendantes du maillage et très localisées. Nous présenterons dans la deuxième partie de cette thèse (Chapitre 4 et Chapitre 5) un cadre général pour construire de nouvelles bases de fonctions qui ne sont pas localisées dans l'espace géométrique, mais dans l'espace fréquentiel : les Harmoniques Variétés. En raison de cette localisation en fréquence, les Harmoniques Variétés nous permettront de faire du filtrage spectral de façon très directe. Toutefois, les Harmoniques Variétés sont obtenues par un simple changement de base dans l'espace usuel des fonctions interpolantes. Par conséquent, elles sont aussi dépendantes de la qualité du maillage de la surface. En outre, parce qu'elles ne sont pas localisées, elles ne font que saisir la forme globale de la surface, sans tenir compte des détails fins. Pour construire des bases de fonctions fondamentalement nouvelles et surmonter ces limites, nous avons besoin d'un domaine de base intermédiaire entre un simple triangle et la surface toute entière. Une façon de définir ces domaines de base est de trouver une paramétrisation quad de la surface. Pour orienter une telle paramétrisation, on va avoir besoin de construire un champ de directions sur la surface. Ainsi, la dernière partie de cette thèse porte sur ces champs de directions.

Nous allons maintenant détailler la structure et l'organisation de cette thèse.

Structure de la thèse

Cette thèse est organisée en trois parties. Dans la **première partie**, on présentera les concepts et outils théoriques dont nous nous servons dans les autres parties. Cette première partie est composée de trois chapitres :

- Le Chapitre 1 propose un survol rapide de notions de géométrie fondamentales. En particulier, on présentera les variétés et les surfaces triangulées dont on se servira abondamment dans cette thèse.
- Le Chapitre 2 présente le formalisme de la méthode par éléments finis, qui fournit un outil mathématique essentiel pour étudier les bases de fonctions et donne des garanties et limites importantes les concernant. Nous présenterons aussi une adaptation aux surfaces des bases de fonctions usuelles en éléments finis. On terminera ce chapitre en donnant deux résultats simples que nous avons obtenus en autorisant une certaine mobilité à ces bases de fonctions.
- Le Chapitre 3 explique le formalisme du calcul extérieur et de son équivalent discret. En plus d'une simple exposition, nous discuterons certaines relations importantes entre le calcul extérieur et son équivalent discret, et proposerons une réflexion sur leurs dualités.

La **deuxième partie** de cette thèse propose une étude des Harmoniques Variétés, définies comme les fonctions propres du Laplacien. Cette partie est composée de deux chapitres :

- Le Chapitre 4 étudie le Laplacien sur les surfaces, en commençant par sa définition continue. On compare alors des diverses discrétisations et équivalents discrets¹ du Laplacien. On construira ensuite les Laplaciens se basant sur les bases de fonctions présentées au Chapitre 2. On s'intéressera aussi à plusieurs aspects importants du Laplacien discret, comme ses conditions aux bord, sa symétrie et sa localisation. Ce chapitre est basé sur la partie théorique de notre article [Vallet and Levy 2008] que nous avons beaucoup développée, étoffée et généralisée.
- Le Chapitre 5 utilise les concepts du Chapitre 4 pour définir les Harmoniques Variétés de façon claire et rigoureuse. On s'intéresse ensuite à l'interprétation, au calcul et aux applications de ces Harmoniques Variétés. Ce chapitre correspond à la partie technique de notre article [Vallet and Levy 2008].

La **troisième et dernière partie** de cette thèse porte sur les champs de directions, et se décompose en cinq chapitres :

- Le Chapitre 6 propose un état de l'art sur la manipulation des directions et sur leurs applications. Nous exhibons les caractéristiques qu'un outil de manipulation de champs de directions doit proposer pour ces applications.
- Le Chapitre 7 introduit la notion de champs de directions à symétries, et étudie leur géométrie et leur topologie. Il propose ensuite une structure pour représenter de tels champs de directions de façon non ambiguë. Ce chapitre est une synthèse des parties théoriques de nos articles [Ray et al. 2008b, Li et al. 2006b, Ray et al. 2008a].
- Le Chapitre 8 fait un état de l'art des méthodes de visualisation de champs de directions, et en propose une basée sur la structure introduite au Chapitre 7 qui dépasse les limites actuelles sur le type de singularités qu'elle permet de visualiser. Ce chapitre correspond à notre article [Li et al. 2006b].
- Le Chapitre 9 introduit un algorithme de lissage de champs de directions à topologie contrainte en se basant sur les résultats et la représentation du Chapitre 7. Cet algorithme assure un contrôle exact sur la topologie (singularités) du champs lissé. Ce chapitre correspond à la partie technique de notre article [Ray et al. 2008b].
- Le Chapitre 10 introduit un autre algorithme qui permet de lisser des champs de directions, mais cette fois-ci avec un contrôle plus faible sur l'apparition des singularités en fonction d'un niveau de détail déterminé. Il démontre comment la théorie présentée au Chapitre 7 peut être utilisée pour améliorer les algorithmes classiques de lissage de champs de directions en introduisant un contrôle *faible* de la topologie. L'outil de manipulation de champs de directions en découlant réalise un compromis entre le contrôle *fort* proposé au Chapitre 9, et les algorithmes usuels qui font apparaître les singularités spontanément sans aucun contrôle. Ce chapitre correspond à notre article [Ray et al. 2008a] en cours de soumission.

Avant d'entrer dans le coeur du sujet, nous allons discuter quelques concepts importants qu'on retrouvera tout au long de la thèse, et en particulier sa méthodologie générale.

Concepts principaux

La modélisation géométrique

Nous percevons le monde qui nous entoure comme géométrique. Il est constitué de matière disposée dans l'espace dans une certaine configuration. Dans cette configuration, il y a des interfaces, des surfaces de contact entre des matières de types différents. Ce sont ces surfaces de contact que nous percevons, par

¹On expliquera dans la prochaine section la distinction entre discrétisation et équivalent discret

la vue et le toucher. Pour représenter une réalité dans un ordinateur, ce sont donc aussi ces surfaces que l'on va tenter de modéliser. La difficulté majeure est que ces surfaces sont des objets continus (du moins à notre échelle), alors qu'un ordinateur ne peut représenter que des objets discrets. Le premier objet de la modélisation géométrique est donc celui de la représentation, ou de la discrétisation des surfaces.

Géométrie et topologie

Dans cette thèse, je me suis intéressé à l'une de ces représentations : les surfaces triangulées. Les surfaces triangulées approximent une surface continue de façon linéaire par morceaux triangulaires. Elle est donc constituée de triangles (plats) dans l'espace ambiant (3D) partageant des arêtes. Cette représentation est à la fois :

- Géométrique : les positions des sommets des triangles sont définies par des réels qui peuvent évoluer de façon continue.
- Topologique : les agencements de triangle et la connectivité des sommets ne peut être modifiée que par une opération discrète consistant à couper et recoller, à détruire et reconstruire.

Cette dualité étant déjà présente à la base dans la représentation des surfaces, on va la retrouver dans tous les problèmes que se pose la modélisation géométrique, et on verra qu'elle est encore plus présente dans la structure des champs de directions. Pour résoudre un problème donné, la méthodologie employée au cours de cette thèse a été de trouver ses degrés de liberté continus et discrets, géométriques et topologiques, puis d'exhiber leurs relations, et de trouver les moyens de les contrôler de la façon la plus simple ou la plus judicieuse possible.

Les bases de fonctions

Les bases de fonctions peuvent être vues comme un outil mathématique pour implémenter cette méthodologie, puisqu'elles vont nous permettre de définir une fonction sur une surface par l'intermédiaire d'une suite finie de coefficients réels (degrés de libertés). La discrétisation d'une fonction continue est le fait de la transformer en une telle suite de coefficients entier ou réels. Les bases de fonctions sont donc un outil idéal pour discrétiser les fonctions continues puisqu'elles donnent une signification concrète à ces coefficients : ce sont les coordonnées de la fonction dans une base de fonctions. En d'autres termes, une fonction est assimilée à un point dans l'espace vectoriel engendré par les fonctions de la base. Le choix des fonctions de base est donc primordial puisque de ce choix vont dépendre tous les calculs que l'on pourra faire sur la fonction. L'objet premier de cette thèse est de trouver de nouvelles bases de fonctions sur les surfaces, avec des propriétés qui seront utiles pour résoudre des problèmes liés aux surfaces.

Outils discrets ou discrétisation ?

Un autre thème récurrent de cette thèse sera les représentations discrètes. L'informatique graphique cherche à représenter un monde visuel continu à l'aide d'un ordinateur qui manipule par construction des objets discrets. A partir d'une représentation discrète, nous pouvons construire deux types d'outils :

1. Des outils *discrets* agissant directement sur la représentation discrète en tant qu'objet discret.
2. Des outils *discrétisés* qui consistent à se baser sur des concepts continus que l'on cherche à approximer de façon discrète.

Ces deux approches ne sont pas antagonistes. En fait, on s'aperçoit souvent que les outils issus des deux approches amènent au même résultat. On étudiera plus tard les points communs et les différences entre un outil discret nommé le calcul extérieur discret, et la discrétisation d'opérateurs par une méthode d'éléments finis. En revanche, on peut noter qu'en général, la classe des outils discrets est plus vaste que celle des outils discrétisés puisqu'elle ne se pose aucune contrainte, alors que les outils discrétisés sont plus spécifiques, mais du coup mieux compris et bien justifiés en tant qu'approximation du continu.

Première partie

Calcul géométrique

1

Le problème de la représentation géométrique

Sommaire

1.1 De la géométrie	10
1.1.1 Les objets géométriques	10
1.1.2 Les variétés	11
1.1.3 Les représentations géométriques	12
1.2 Variétés affines	13
1.2.1 Géométrie projective	13
1.2.2 Le plan	14
1.2.3 Définition implicite	14
1.2.4 Indépendance géométrique	14
1.2.5 Coordonnées barycentriques	14
1.3 Les complexes simpliciaux	15

Introduction

Cette thèse se classe dans le domaine de la géométrie numérique. Ainsi ce chapitre va présenter les diverses notions théoriques et pratiques impliquées dans la définition et la manipulation numérique de la géométrie.

Plan du chapitre

Ce chapitre est organisé en trois sections. La Section 1.1 présentera les différentes façons de représenter la géométrie de façon numérique. La Section 1.2 se concentrera sur la représentation géométrique la plus simple : les variétés affines qui regroupent les points, droites et plans. Ce sera l'occasion de définir certains concepts géométriques de base comme la géométrie projective, l'indépendance géométrique et les coordonnées barycentriques. Enfin, la Section 1.3 présentera la représentation géométrique sur laquelle on travaille dans cette thèse : les surfaces triangulées qui sont formalisées mathématiquement par la notion de complexe simplicial.

Contributions

- Nous passons en revue les façons de définir de la géométrie de façon numérique ;
- Nous introduisons la formalisation mathématique des notions et objets géométriques que nous utilisons dans tout le reste de cette thèse.

1.1 De la géométrie

Nous allons commencer par définir ce que sont les objets géométriques, et en particulier d'où ils viennent, comment ils sont modélisés de façon mathématique par des *variétés*, et comment ils sont représentés numériquement.

1.1.1 Les objets géométriques

Les objets géométriques sur lesquels on peut travailler ont principalement deux sources :

- La réalité : on essaye de représenter un objet existant par diverses méthodes d'acquisition :
 - Scanner 3D : un faisceau laser balaye la surface, pendant que des caméras enregistrent la scène. Les images acquises permettent de retrouver les positions 3D des points balayés, ce qui permet d'approximer la surface réelle par un nuage de points.
 - Enveloppe visuelle : De nombreuses vues d'un objet sont prises sous des angles variés. De chaque image, on peut extraire la silhouette de l'objet qui correspondra à une enveloppe en 3D. En intersectant ces enveloppes, on obtient un objet 3D qui a les mêmes silhouettes que l'objet acquis dans toutes les images, et qui l'approxime d'autant mieux que les vues sont nombreuses et variées.
 - Vision multiscopique : notre objet 3D est de nouveau vu par plusieurs caméras (au moins 2), mais cette fois-ci on trouve des points caractéristiques sur l'objet qu'on cherche à apparier entre les différentes images. Ces appariements permettent de retrouver la position 3D des points caractéristiques. Ce problème est appelé *photogrammétrie* si les positions des caméras sont connues, et *vision par ordinateur* si elles ne le sont pas.
 - Imagerie sonique : des charges explosives sont déclenchées en divers points de la surface terrestre, et les ondes réfléchies sont enregistrées en divers points. Un problème inverse est alors résolu pour reconstruire la géométrie des couches géologiques à partir de ces données.
 - Échographie : le principe est le même que précédemment, à part que les ondes sont maintenant ultrasoniques (non audibles), ce qui permet de capturer des détails plus fins, mais sur des objets beaucoup plus petits (corps humain).
 - IRM, rayons X,... : les ondes sonores peuvent être remplacées par des ondes électromagnétiques pour capturer des détails encore plus fins, ou d'autres propriétés des objets.
- L'imagination : Un opérateur (artiste, designer, ingénieur...) conçoit un objet géométrique qu'il imagine en 3D. Il utilisera alors un logiciel nommé *modeleur* qui lui fournit des outils pour interagir avec la forme des structures géométriques qu'il crée.

Dans tous les cas, on considèrera que le résultat de ces processus est une approximation de la géométrie réelle ou imaginaire. Il nous faut maintenant formaliser ce qu'est cette géométrie acquise, et comment elle peut être représentée et manipulée numériquement. Pour faire cela, on va définir un objet mathématique qui généralise la plupart des objets géométriques que l'on va vouloir manipuler, puis on présentera les façons usuelles de les représenter.

1.1.2 Les variétés

De par son étymologie, la géométrie est la science de la mesure (*metry*) de la terre (*geo*), ou plus généralement de l'espace ambiant. Dans une approximation non-relativiste, nous assimilons cet espace ambiant à l'espace vectoriel \mathbb{R}^3 , d'autant plus volontiers que le temps restera souvent absent de nos considérations. Dans une approche plus générale, nous définirons la géométrie comme l'étude des sous-ensembles de \mathbb{R}^3 . Parmi ces sous ensembles, nous nous intéresserons en particulier à ceux pour lesquels on peut définir une dimension entière et que l'on appelle *variétés* (plongées dans \mathbb{R}^d).

Définition 1 (Variété plongée) $\mathcal{S} \subset \mathbb{R}^d$ est une k -variété plongée dans \mathbb{R}^d si pour tout $\mathbf{x} \in \mathcal{S}$ il existe des voisinages ouverts $N_{\mathbf{x}}$ et N_0 de \mathbf{x} et 0 (l'origine de \mathbb{R}^d) et un homomorphisme² $f : N_{\mathbf{x}} \rightarrow N_0$ tel que :

$$f(N_{\mathbf{x}} \cap M) = N_0 \cap (\mathbb{R}^k \times \{0\}^{d-k})$$

Intuitivement, une k -variété est un sous espace de \mathbb{R}^d localement isomorphe à \mathbb{R}^k , ou une collection de morceaux qui peuvent être aplatis en ouverts de \mathbb{R}^k . $\mathbb{R}^k \times \{0\}^{d-k}$ (le vecteur nul de dimension $d - k$) peut être vu comme la k -variété canonique dans \mathbb{R}^d . L'existence d'un homomorphisme nous dit que toute k -variété ressemble à $\mathbb{R}^k \times \{0\}^{d-k}$ localement. L'homomorphisme f s'appelle une *paramétrisation locale*, dont les paramètres sont les k coordonnées non nulles de $N_0 \cap (\mathbb{R}^k \times \{0\}^{d-k})$. k s'appelle la dimension de la variété et $d - k$ sa codimension :

- Les 0-variétés sont des points ;
- Les 1-variétés sont des courbes ;
- Les 2-variétés sont des surfaces ;
- Les 3-variétés sont des volumes.

Une variété peut avoir plusieurs composantes connexes, donc le terme "unions de points/courbes/..." serait plus approprié (mais plus lourd). Une variété \mathcal{S} est un ouvert, de sorte qu'il peut différer de son adhérence : $\bar{\mathcal{S}} = \{\mathbf{x} \in \mathbb{R}^d \mid \exists \{\mathbf{x}_n \in \mathcal{S}\} \text{ tel que } \lim_{n \rightarrow \infty} \mathbf{x}_n = \mathbf{x}\}$. Dans ce cas, on appelle $\partial \mathcal{S} = \bar{\mathcal{S}} / \mathcal{S}$ le bord de \mathcal{S} . Si $\partial \mathcal{S} = \emptyset$, on dit que \mathcal{S} est *fermée*, ou *sans bord*. Si $\partial \mathcal{S} \neq \emptyset$, on dit que \mathcal{S} est *ouverte*, ou *avec bord*.

On voudra souvent définir un *espace tangent* aux variétés. La définition formelle d'un tel espace sur une variété quelconque est très abstraite, mais elle devient simple quand la variété est plongée dans \mathbb{R}^d .

Définition 2 (Espace tangent d'une variété plongée) Soit \mathcal{S} une k -variété plongée dans \mathbb{R}^d , et $\mathbf{x} \in \mathcal{S}$ un point de cette variété. Alors, l'espace tangent $T_{\mathbf{x}}\mathcal{S}$ de \mathcal{S} au point \mathbf{x} est l'ensemble des vecteurs $\nabla \gamma(0)$ pour toutes les courbes $\gamma : [-1, 1] \rightarrow \mathcal{S}$ telles que $\gamma(0) = \mathbf{x}$. $T_{\mathbf{x}}\mathcal{S}$ est un espace vectoriel de dimension k .

L'union de tous les espaces vectoriels tangents d'une variété \mathcal{S} est appelé son *fibré tangent* et noté $T\mathcal{S}$. On utilise aussi souvent la notion d'espace cotangent $T_{\mathbf{x}}^*\mathcal{S}$ qui est simplement l'espace des applications linéaires de $T_{\mathbf{x}}\mathcal{S}$ vers \mathbb{R} , et on appelle $T^*\mathcal{S}$ le *fibré cotangent* correspondant.

Pour mesurer des longueurs, aires, angles,... sur une variété, on l'équipe d'une *métrique*.

Définition 3 (Métrique (Riemannienne)) Une *métrique (Riemannienne)* est une forme bilinéaire symétrique non dégénérée de $T\mathcal{S} \times T\mathcal{S}$ dans \mathbb{R} .

Une métrique peut être vue simplement comme un produit scalaire dans $T\mathcal{S}$. Le plongement d'une variété dans \mathbb{R}^d induit une métrique canonique qui est donnée par le produit scalaire dans \mathbb{R}^d . La paramétri-

²Homomorphisme : application continue bijective dont l'inverse est aussi continu

sation d'une variété induit une autre métrique donnée par le produit scalaire dans l'espace paramétrique $\mathbb{R}^k \times \{0\}^{d-k}$. Une variété munie d'une métrique Riemannienne est appelée *variété Riemannienne*.

Remarque 1.1.1 *A une variété \mathcal{S} correspond une fonction caractéristique $1_{\mathcal{S}}(\mathbf{x}) = 1$ si $\mathbf{x} \in \mathcal{S}$ et 0 sinon. De la même façon, on peut associer à \mathcal{S} l'opérateur $I_{\mathcal{S}}f = \int_{\mathcal{S}} f(\mathbf{x})d\mathcal{S}(\mathbf{x}) = \int_{\mathbb{R}^3} 1_{\mathcal{S}}(\mathbf{x})f(\mathbf{x})d\mathbf{x}$. Comme l'analyse fonctionnelle est un domaine très bien compris, on essaiera souvent de représenter et de manipuler la géométrie par l'intermédiaire de fonctions. Cette analogie entre géométrie et fonctions est loin d'être triviale. C'est l'un des points centraux de ce chapitre.*

1.1.3 Les représentations géométriques

Le problème de la discrétisation

La géométrie numérique se base sur des représentations numériques de la géométrie et donc des variétés. On appellera *variété numérique* une variété qui peut se définir par une suite finie de symboles, et *discrétisation* le fait d'approximer une variété par une variété numérique. Le processus de discrétisation se heurte à deux problèmes :

- On aura en général besoin de réels pour définir une variété de façon numérique, que ce soient des coefficients de polynômes ou des coordonnées de points. L'informatique graphique utilise généralement des `double` pour représenter les réels. Cette "discrétisation" de \mathbb{R} est particulièrement simpliste puisque \mathbb{R} est de cardinal infini non dénombrable, incommensurable avec \mathbb{N} qui est de cardinal infini dénombrable, lui même incommensurable avec l'ensemble des `double` qui est de cardinal 2^{32} . Cependant, l'imprécision apportée par les `double` aura en général un impact limité dans nos applications, même si il faudra garder cela en tête lorsque l'on divisera par des quantités proches de 0 ou que l'on inversera des matrices mal conditionnées.
- Un problème beaucoup plus important en revanche est qu'il faudrait a priori un nombre infini de `double` pour représenter une variété (trois pour chaque point de la variété), ou un nombre infini de variables binaires (une pour chaque point de \mathbb{R}^3). On ne pourra donc pas approximer chaque point de la surface \mathcal{S} , mais seulement les regrouper par blocs formés de sous ensembles de \mathcal{S} . Ainsi, pour chaque bloc on ne disposera que d'un nombre fini de variables pour représenter un nombre infini de points, conduisant nécessairement à des approximations, beaucoup plus grossières que celle d'approximer des réels par des `double`.

Évidemment, plus on disposera de variables, plus on repoussera ces limites, c'est à dire moins la perte d'information due à la discrétisation sera importante. Nous pouvons maintenant nous intéresser à la façon de représenter des variétés en pratique.

Représentations numériques

Il va donc nous falloir choisir une représentation numérique de la géométrie, c'est à dire des variétés. De façon générale, toutes les constructions géométriques usuelles se basent sur deux types de primitives géométrique numérique :

1. $V_{>}(f) = \{\mathbf{x} \in \mathbb{R}^n | f(\mathbf{x}) > 0\}$: ce sont des n -variétés avec bord.
2. $V_{=}(f) = \{\mathbf{x} \in \mathbb{R}^n | f(\mathbf{x}) = 0\}$: ce sont des $n - 1$ variétés sans bord. On remarquera que $V_{=}(f)$ est le bord de $V_{>}(f)$, ce qu'on notera : $V_{=}(f) = \partial V_{>}(f)$

A partir de ces primitives de base, on va définir des variétés numériques :

- Par transformation $T : \mathbb{R}^m \rightarrow \mathbb{R}^n$. La transformée par T d'une primitive $V_=(f)$ est alors : $T(V_=(f)) = \{\mathbf{x} = T(\mathbf{y}) \in \mathbb{R}^n \mid \mathbf{y} \in V_=(f)\}$. On dira alors que la primitive est paramétrée ou paramétrique, puisque les coordonnées dans l'espace de départ (\mathbb{R}^m) paramètrent la variété résultant de la transformation. En général, ces transformations préservent la dimension de la variété, mais pas celle de l'espace dans laquelle elle est plongée.
- Par union ou intersection de primitives, éventuellement paramétrique.

Cette façon de définir des variétés numérique est assez générale pour englober la plupart des représentations utilisées actuellement. On notera cependant que les surfaces de subdivision n'entrent pas complètement dans cette catégorie puisqu'elles sont définies comme une limite de telles représentations. Pour les autres, on peut citer :

- Nuages de points : chaque point est une intersection de 3 primitives de la forme $V_=(f(\mathbf{x}) = x_i - p_i)$, et le nuage est une union de points.
- Surfaces polyédrales : On peut définir un polyèdre à n sommets dans \mathbb{R}^2 comme intersection de n primitives $V_>(f(\mathbf{x}) = \mathbf{ax} - \mathbf{b})$, puis le transformer dans \mathbb{R}^3 .
- Splines : Un carré est défini dans \mathbb{R}^2 comme intersection de 4 primitives $V_>$, et transformé dans \mathbb{R}^3 par une fraction rationnelle.
- Les quadriques, quartiques... sont des primitives de la forme $V_=(P)$ où P est un polynôme.

Ces constructions ont le mérite de représenter des variétés par des fonctions analytiques de $\mathbb{R}^n \rightarrow \mathbb{R}^m$, et représentent donc de la géométrie par un nombre fini de symboles, ce qui va permettre de les manipuler de façon informatique. Nous allons maintenant voir à partir de quoi sont créées ces structures de façon plus concrète.

Nous allons maintenant présenter les variétés les plus simples, c'est à dire les variétés affines. Elles sont particulièrement adaptées à un traitement numérique puisque les outils qu'on utilisera pour les manipuler seront ceux de l'algèbre linéaire.

1.2 Variétés affines

1.2.1 Géométrie projective

L'espace ambiant géométrique étant \mathbb{R}^3 , on note $\mathbf{x} = (x, y, z)$ ses point. La géométrie projective se base sur l'espace vectoriel \mathbb{R}^4 des point $\mathbf{X} = (x, y, z, s)$. Dans les cas particuliers :

- $s = 1$: $\mathbf{X} = (x, y, z, 1)$ est un point géométrique.
- $s = 0$: $\mathbf{X} = (x, y, z, 0)$ est un vecteur.

La géométrie projective propose un typage des données (x, y, z) en terme de points ou de vecteur, puisqu'elle donne automatiquement le type du résultat d'une opération :

- La somme ou la différence de vecteurs est un vecteur.
- La somme d'un point et d'un vecteur est un point.
- Une somme de point divisée par le nombre de points est un point (barycentre).

Si le résultat d'une opération n'a pas $s = 0$ ou $s = 1$, il ne correspond pas à un objet de la géométrie usuelle, mais on peut le voir comme un point pondéré, ou point à l'échelle s , et il faut alors diviser ses coordonnées par s pour retrouver un point au sens usuel.

1.2.2 Le plan

Le plan est le plus simple des variétés affines puisqu'il est défini par une seule relation affine. Nous allons donner sa définition ainsi que différentes équations implicites et explicites (paramétriques).

1.2.3 Définition implicite

Définition 4 (Plan) *Un plan est un ensemble de points $\mathbf{x} \in \mathbb{R}^3 = (x, y, z)$ qui vérifient $f(\mathbf{x}) = 0$ où f est une fonction affine $\mathbb{R}^3 \rightarrow \mathbb{R}$.*

L'ensemble des plans est donc le sous-ensemble de $V_=(f)$ correspondant aux fonctions f affines. Une fonction affine $\mathbb{R}^3 \rightarrow \mathbb{R}$ s'écrit nécessairement sous la forme $f(x, y, z) = ax + by + cz + d = \mathbf{A} \cdot \mathbf{X}$ avec $a, b, c, d \in \mathbb{R}$, ce qui nous donne une représentation d'un plan par quatre réels. Cette définition est redondante puisque n'importe quelle fonction λf , $\lambda \in \mathbb{R}^*$ correspond au même plan. L'ensemble des plans est donc isomorphe à $\mathbb{R}^4 / \mathbb{R}^* = \mathcal{S}^3 \times \mathbb{R}^+$ où \mathcal{S}^3 est la sphère unité. Cela veut dire que la représentation la plus concise d'un plan n'est pas par 4 réels mais par un point de la sphère et un réel positif ou nul. Par exemple, on peut choisir de représenter le plan par sa normale (vecteur unitaire $\in \mathcal{S}^3$) et sa distance à l'origine. En termes plus mathématiques, le plan est défini par l'équation $\mathbf{x} \cdot \mathbf{n} - d = 0$, $\mathbf{x} \in \mathbb{R}^3$, $\mathbf{n} \in \mathcal{S}^3$, $d \in \mathbb{R}^+$. Imposer $d \geq 0$ rend le choix du sens de la normale non ambigu, sauf pour $d = 0$. En pratique, on contourne ce problème en autorisant $d \in \mathbb{R}$, ce qui rend la représentation légèrement redondante, mais plus pratique.

1.2.4 Indépendance géométrique

Définition 5 (Indépendance géométrique) *En géométrie projective, n points $\mathbf{X}_1, \dots, \mathbf{X}_n$ sont indépendants si il n'existe pas de poids $\lambda_1, \dots, \lambda_n \in \mathbb{R}^3 = (x, y, z)$ qui vérifient $\sum \lambda_i \mathbf{X}_i = 0$, ou de façon équivalente si $\det(\mathbf{X}_1, \dots, \mathbf{X}_n) \neq 0$.*

L'outil privilégié de l'indépendance linéaire est le déterminant. Notons encore une fois que la géométrie projective permet de traiter à la fois l'indépendance linéaire des vecteurs, et l'indépendance géométrique des points (3 points sont indépendants si ils ne sont pas alignés, 4 points si ils ne sont pas coplanaires). L'indépendance linéaire va nous donner une deuxième équation pour les plans sous la forme $\det(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}) = 0$. $\mathbf{X}_1, \mathbf{X}_2$ et \mathbf{X}_3 sont des points ou des vecteurs, mais l'un au moins doit être un point. Cette définition est plus pratique pour obtenir l'équation d'un plan dont on connaît 3 points, 2 points et un vecteur ou 1 point et 2 vecteurs. Cette définition est beaucoup plus redondante puisqu'elle nécessite de stocker 9 réels au lieu de 3 ou 4. En fait cette redondance peut être utile puisqu'elle définit une paramétrisation. En effet, on peut toujours privilégier un point (mettons \mathbf{X}_1) et définir le plan comme l'ensemble des points : $\mathbf{X}(\alpha, \beta) = \mathbf{X}_1 + \alpha \mathbf{X}_2 + \beta \mathbf{X}_3$, $\alpha, \beta \in \mathbb{R}$ étant les paramètres.

1.2.5 Coordonnées barycentriques

Comme on l'a vu, on peut définir une somme de points en géométrie projective, et en particulier une somme de points pondérés. Si la somme des pondérations est 1, le résultat est bien un point qui est leur barycentre pondéré : $\mathbf{X}(\lambda_1, \dots, \lambda_n) = \sum_i \lambda_i \mathbf{X}_i$, $\sum \lambda_i = 1$.

Ceci nous donne une troisième définition du plan. Le plan passant par trois points X_1, X_2 , et X_3 est l'ensemble des points $X(\lambda_1, \lambda_2, \lambda_3) = \lambda_1 \mathbf{X}_1 + \lambda_2 \mathbf{X}_2 + \lambda_3 \mathbf{X}_3$ tel que $\sum \lambda_i = 1$. Cela amène à la même

représentation paramétrique en choisissant l'un des points pour éliminer l'une des variable λ_i par la relation $\sum_i \lambda_i = 0$.

1.3 Les complexes simpliciaux

Une façon très simple et particulièrement adaptée à l'informatique de décrire la géométrie est par des surfaces triangulées. Une définition mathématique des surfaces triangulées utilisées en informatique graphique passe par les notions de simplexes et de complexes simpliciaux.

Définition 6 (Simplexe) Dans \mathbb{R}^n , un k -simplexe est l'enveloppe convexe de $k + 1$ points géométriquement indépendants :

$$s_k = [v_0 \dots v_k] = \left\{ \mathbf{x} = \sum_j \lambda_j [v_j] \in \mathbb{R}^d \mid \lambda_0, \dots, \lambda_k \in \mathbb{R}^+ \text{ and } \sum_i \lambda_i = 1 \right\}$$

- Sommets et dimension : On appelle v_0, \dots, v_k les *sommets* du simplexe et k sa *dimension*. Ainsi un 0-simplexe est un point, un 1-simplexe est une arête, un 2-simplexe est un triangle, et un 3-simplexe est un tétraèdre. A cause de la condition d'indépendance, la dimension d'un simplexe est au plus celle de l'espace dans lequel il est plongé ($k \leq d$).
- Concaténation : $s_k = [v_0 \dots v_k]$ est indexé par une concaténation d'indices, de sorte qu'en appliquant cette concaténation récursivement, on a $\dim(s_1 \dots s_d) = \sum_i \dim(s_i)$.
- Face : si $k < l$ et $\{v_0, \dots, v_k\} \subset \{v'_0, \dots, v'_l\}$, $s_k = [v_0 \dots v_k]$ est appelé une *face* de $s_l = [v'_0, \dots, v'_l]$, ce qu'on note $s_k \prec s_l$.
- l -étoile : on note $St_l(s_k) = \{s_l \mid s_k \prec s_l\}$ la (l -)étoile d'un simplexe. C'est l'ensemble des l -simplexes s_l de degrés $l > k$ et qui contiennent s_k .

Définition 7 (Complexe simplicial) Un complexe simplicial M dans \mathbb{R}^n est une collection de k -simplexes vérifiant :

- $s_k \in M$ et $s_l \prec s_k \Rightarrow s_l \in M$: chaque face d'un simplexe de M est dans M .
- $s_k, s_{k'} \in M$ et $s_l = s_k \cap s_{k'} \Rightarrow s_l \prec s_k$ et $s_l \prec s_{k'}$: l'intersection de deux simplexes de M est une face de chacun d'eux.

Si tous les simplexes d'un complexe M sont de dimension au plus m , on dira que M est de dimension m . Un complexe simplicial permet de décrire une variété numérique de \mathbb{R}^n de façon très concise puisqu'il suffit de connaître ses points (0-simplexes) et sa connectivité (quels sont les sommets de chaque m -simplexe). Ainsi, un complexe simplicial M de dimension m plongé dans \mathbb{R}^n , avec n_s sommets et n_f m -simplexes (faces maximales) sera représenté par :

- Une matrice $S_{n_s, n}$ à coefficients réels des coordonnées dans \mathbb{R}^n de chacun des n_s sommets de M
- Une matrice $C_{n_f, m}$ à coefficients entiers contenant les indices de points (dans S) composant chacun des n_f m -simplexes.

D'un point de vue informatique, cela veut dire que M est représenté par $n_s n$ doubles et $n_f(m + 1)$ ints. On voit dès leur définition que ces surfaces ont à la fois :

- une composante continue ou géométrique correspondant aux positions des sommets dans \mathbb{R}^n , stockées dans des double.
- une composante discrète ou topologique correspondant à la connectivité des m -simplexes, stockée dans des int.

Dans cette thèse, nous nous intéresserons en priorité aux complexes simpliciaux M de \mathbb{R}^3 de dimension 2, c'est à dire aux surfaces de l'espace géométrique. De nombreux algorithmes de géométrie numérique

nécessitent de définir des fonctions sur ces complexes qui vérifient certaines propriétés. En particulier, on s'intéressera aux solutions d'équations différentielles ou aux fonctions minimisant des d'énergies. Les deux prochains chapitres proposent des outils pour déterminer ces fonctions : la méthode des éléments finis et le calcul extérieur discret. Ces deux formalismes sont bien étudiés et justifiés, et fournissent des garanties solides sur la qualité des solutions qu'ils proposent. Le point commun entre ces deux approches est qu'elles cherchent à représenter des fonctions par des vecteurs, c'est à dire par un nombre fini de valeurs. La qualité de cette définition et l'efficacité des méthodes en résultant dépendra ainsi directement de la qualité des *bases de fonctions* utilisées pour faire le lien entre espaces fonctionnels et espaces vectoriels. C'est pourquoi nous allons nous y intéresser de près dans cette partie.

2

Méthode des Éléments finis

Sommaire

2.1	Théorie des éléments finis	18
2.1.1	Espaces de Hilbert	18
2.1.2	Espaces de Hilbert fonctionnels	19
2.1.3	Théorie de Lax-Milgram	21
2.1.4	Approximation interne	21
2.1.5	Galerkin et les éléments finis	22
2.2	Bases de fonctions interpolantes	23
2.2.1	Éléments P_n	24
2.2.2	Interpolation bilinéaire	28
2.3	Bases mobiles	29
2.3.1	Problème	30
2.3.2	Interpolation P_1 des fonctions convexes en 1D	30
2.3.3	Approximation par fonctions génériques en 1D	31
2.3.4	Conclusion	31

Introduction

La méthode des Éléments Finis (EF) est l'une des méthodes de discrétisation des Équations aux Dérivées partielles (EDP) les plus répandues. L'idée de base est de remplacer un espace de Hilbert V (de dimension infinie) dans lequel on recherche la solution de l'EDP par un espace V_n de dimension finie. Comme la dimension de V_n est finie, ses fonctions peuvent être représentées par des vecteurs, et l'opérateur différentiel peut être représenté par une matrice. Ainsi, les EF permettent de convertir des EDP en systèmes linéaires. Les mathématiques appliquées fournissent de nombreux résultats sur la convergence des EF, donnant en particulier des conditions auxquelles la solution dans V_n converge vers la solution dans V quand on raffine V_n . Un aspect important des EF, en particulier dans le cadre de cette thèse, est que V_n doit être une bonne approximation de V pour que la solution soit elle-même bien approximée, et que raffiner V_n doit amener à mieux approximer V .

Organisation du chapitre

Nous commencerons par présenter dans la Section 2.1 la théorie des éléments finis ainsi que ses résultats importants. Dans la Section 2.2, nous décrirons les bases de fonctions interpolantes qui sont très utilisées dans les méthodes par éléments finis. Enfin, nous introduirons dans la Section 2.3 le problème très ouvert des bases de fonctions mobiles qui introduisent plus de degrés de liberté que ce que la méthode des éléments finis n'autorise normalement. Nous présenterons alors deux résultats préliminaires sur ce sujet obtenus pendant cette thèse.

Contributions

- Nous synthétisons les concepts et résultats principaux des méthodes par éléments finis.
- Nous présentons les bases de fonctions interpolantes que nous utiliserons abondamment pendant cette thèse.
- Nous introduisons le concept de bases de fonctions mobiles ainsi que deux résultats préliminaires les concernant.

2.1 Théorie des éléments finis

Cette section présente des résultats théoriques importants pour les EF. Elle est basée en particulier sur les cours de Grégoire Allaire [Allaire 2002]. Comme nous utiliserons beaucoup le formalisme des EF dans cette thèse, il nous paraît important de présenter ses fondements. En particulier, nous donnerons certaines garanties que proposent les EF, mais aussi leurs limites.

2.1.1 Espaces de Hilbert

La méthode des éléments finis repose sur les espaces de Hilbert. Ces espaces sont non seulement un objet mathématique, mais aussi un concept : celui d'identifier des espaces de fonctions à des espaces vectoriels. En d'autres termes, il y a derrière la notion d'espace de Hilbert l'idée de construire des espaces de fonctions par combinaisons linéaires d'un certain nombre de fonctions de base, et d'approximer n'importe quelle fonction de cette façon. Les espaces de Hilbert permettent alors d'étendre plusieurs concepts géométriques fondamentaux (produit scalaire, norme, projection,...) aux bases de fonctions, ce qui permet de construire des outils à la fois intuitifs et efficaces. Enfin c'est un outil élégant de discrétisation de fonctions, où la fonction n'est pas interpolée mais approximée au sens de la norme. Nous allons maintenant donner une définition (en deux étapes) des espaces de Hilbert, et essayer de véhiculer les intuitions sous-jacentes.

Définition 8 (Espace à produit interne réel) *Un espace à produit interne réel est un espace vectoriel E défini sur \mathbb{R} , avec une forme bilinéaire symétrique définie positive non dégénérée appelée produit interne.*

Le produit interne est une application :

$$\langle \cdot, \cdot \rangle : V \times V \rightarrow \mathbb{F}$$

qui satisfait les axiomes suivants : pour tout $\mathbf{x}, \mathbf{y}, \mathbf{z} \in E, a, b \in \mathbb{R}$:

- Symétrie : $\langle x, y \rangle = \langle y, x \rangle$.

- Linéarité en la première variable : $\langle ax, y \rangle = a\langle x, y \rangle$, $\langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle$. Par symétrie, cela implique la linéarité en la seconde variable.
- Positivité stricte : $\langle x, x \rangle \geq 0$.
- Non dégénérescence : $\langle x, x \rangle = 0 \rightarrow x = 0$.

Intuitivement, il est utile de voir ce produit interne comme un "produit scalaire" dans un espace vectoriel. Cependant, comme nous le verrons plus tard, il ne faut pas confondre le produit interne avec le produit scalaire usuel car il est en général différent. Il nous faut encore une condition pour construire un espace de Hilbert.

Définition 9 (Espace de Hilbert) *Un espace de Hilbert est un espace réel à produit interne qui est un espace complet et normé (espace de Banach) pour la norme définie par le produit interne.*

La condition de complétude de l'espace (auss appelé condition de Cauchy) signifie que toute suite de Cauchy dans un espace de Hilbert V a une limite qui est encore dans V . Intuitivement, un espace est complet si il n'a pas de trous ponctuels.

A part ces quelques conditions techniques qui évitent certains cas dégénérés, un espace de Hilbert peut être simplement vu comme un espace vectoriel (éventuellement de dimension infinie) muni d'un produit interne. Dans le cas où sa dimension est finie, un espace de Hilbert a une base $\{\Phi_i\}_{i=1..n}$, et son produit interne est complètement déterminé par la matrice symétrique M_Φ d'élément $m_{ij} = \langle \Phi_i, \Phi_j \rangle$ que l'on appelle la *matrice du produit interne* ou plus simplement *matrice de masse*. Les espaces de Hilbert sont des espaces très généraux et abstraits. Nous allons maintenant en construire des plus spécifiques dont les éléments sont des fonctions à valeurs dans \mathbb{R} .

2.1.2 Espaces de Hilbert fonctionnels

Beaucoup d'espaces fonctionnels ont une structure d'Espace de Hilbert particulièrement bien adaptée pour résoudre des EDP. Nous allons maintenant décrire les plus répandus et les plus utilisés.

Soit un ensemble ouvert $\Omega \subset \mathbb{R}^d$. Nous définissons :

- L'espace $L^2(\Omega)$ des fonctions de carré intégrable sur Ω , muni d'un produit interne :

$$\langle f, g \rangle = \int_{\Omega} f(x)g(x)dx$$

$L^2(\Omega)$ est un espace de Hilbert avec la norme :

$$\|f\|_{L^2(\Omega)}^2 = \int_{\Omega} f^2(x)dx$$

- L'espace $C_c^\infty(\Omega)$ des fonctions de classe C^k à support compact dans Ω .
- L'espace $C_c^\infty(\Omega)$ des fonctions de classe C^∞ à support compact dans Ω . Il n'est pas évident que $C_c^\infty(\Omega)$ ne contient pas que la fonction nulle puisque toutes les dérivées sont nulles lorsque l'on rentre dans le support. Cependant, non seulement ce n'est pas le cas, mais en plus C_c^∞ approxime bien L^2 .

Nous avons $C_c^\infty(\Omega) \subset C_c^k(\Omega) \subset C_c^{k-1}(\Omega) \subset C_c^1(\Omega) \subset C_c^0(\Omega) \subset L^2(\Omega)$ et donc $C_c^k(\Omega)$ et $C_c^\infty(\Omega)$ héritent du même produit interne que $L^2(\Omega)$. $L^2(\Omega)$ est le plus grand espace de Hilbert dans lequel on peut espérer travailler en analyse fonctionnelle puisque la seule condition que l'on pose est que la norme des fonctions soit finie. Les espaces $C_c^k(\Omega)$ contiennent des fonctions sur lesquelles on pose des conditions de régularité, et ils sont donc beaucoup plus petit. $C_c^\infty(\Omega)$ impose encore plus de régularité, et il est donc encore plus petit. Cependant, un théorème prouve que même $C_c^\infty(\Omega)$ approxime bien $L^2(\Omega)$:

Théorème 2.1.1 (Densité de $C_c^\infty(\Omega)$) $C_c^\infty(\Omega)$ est dense dans $L^2(\Omega)$. Cela veut dire que pour toute fonction $f \in L^2(\Omega)$, il existe une séquence $f_n \in C_c^\infty(\Omega)$ telle que :

$$\lim_{n \rightarrow +\infty} \|f - f_n\|_{L^2(\Omega)} = 0$$

Et parce que $C_c^\infty(\Omega) \subset C_c^k(\Omega)$, $C_c^k(\Omega)$ est aussi dense dans $L^2(\Omega)$. En d'autres termes, quelque soit la fonction $f \in L^2(\Omega)$, on peut trouver une fonction de $C_c^\infty(\Omega)$ (et donc de $C_c^k(\Omega)$) arbitrairement proche de f (au sens de la norme L^2). Dans notre contexte, nous devons souvent travailler avec des fonctions de $C_c^0(\Omega)$ avec des mauvaises conditions de régularité, et il est intéressant de savoir qu'il existe toujours une fonction infiniment régulière arbitrairement proche de la notre. En particulier, même si notre fonction n'est pas dérivable, elle est arbitrairement proche d'une fonction qui l'est. Cela va nous permettre par exemple de définir la dérivation dans un sens plus faible.

Dérivation faible

La dérivation faible généralise la dérivée usuelle en lui donnant un cadre d'application plus large.

Définition 10 (Dérivation faible) $f \in L^2(\Omega)$ est dite faiblement dérivable si il existe des fonctions $\{w_i \in L^2(\Omega)\}_{i=1..d}$ telles que pour tout $f_\infty \in C_c^\infty(\Omega)$, on a :

$$\int_{\Omega} f(x) \frac{\partial f_\infty}{\partial x_i}(x) dx = - \int_{\Omega} w_i(x) f_\infty(x) dx$$

On appelle w_i la i ème dérivée partielle faible de f , ce qu'on note $\frac{\partial f}{\partial x_i}$.

Espaces de Sobolev

Pour finir notre rapide survol des espaces de Hilbert utiles, nous allons présenter quelques espaces de Sobolev, qui sont les espaces dans lesquels se trouvent les solutions des EDPs. Soit $\Omega \subset \mathbb{R}^d$ un ouvert, nous définissons :

– L'espace de Sobolev :

$$H^1(\Omega) = \left\{ f \in L_2(\Omega) \text{ tel que } \forall i = 1..d \quad \frac{\partial f}{\partial x_i} \in L_2(\Omega) \right\}$$

– L'espace de Sobolev $H_0^1(\Omega)$ est la fermeture de $C_c^\infty(\Omega)$ dans $H^1(\Omega)$. En d'autres termes, $H_0^1(\Omega)$ est le sous espace de $H^1(\Omega)$ des fonctions dont les valeurs et les dérivées s'annulent au bord $\partial\Omega$.

Ces deux espaces sont des espaces de Hilbert pour le produit interne :

$$\langle f, g \rangle = \int_{\Omega} (f(x)g(x) + \nabla f(x) \nabla g(x)) dx$$

et la norme :

$$\|f\|_{H^1(\Omega)}^2 = \int_{\Omega} (f^2(x) + |\nabla f(x)|^2) dx$$

Comme les EDPs font intervenir des opérateurs différentiels, nous devons définir de tels espaces de Sobolev sur lesquels ces opérateurs sont bien définis. De plus, les EDP incluent généralement des conditions aux bords qui peuvent être incluses directement dans la définition de l'espace (de Sobolev) solution. Nous n'allons pas étudier les espaces de Sobolev en détail ici, donc quand nous étudierons une EDP, nous appellerons simplement $H(\Omega)$ l'espace de Sobolev dans lequel se trouve sa solution, c'est à dire l'espace dans lequel les opérateurs différentiels impliqués dans l'EDP sont bien définis, et dont les fonctions satisfont les conditions aux bords.

2.1.3 Théorie de Lax-Milgram

La théorie de Lax-Milgram dans un espace de Hilbert V donne les conditions sous lesquelles le problème :

$$\text{Trouver } f \in V \text{ tel que } a(f, g) = L(g) \forall g \in V \quad (2.1)$$

admet une solution unique.

Théorème 2.1.2 (Lax-Milgram) Si :

- $L : V \rightarrow \mathbb{R}$ est une forme linéaire.
- L est continue, c'est à dire qu'il existe une constante $C > 0$ telle que :

$$|L(v)| \leq C \|v\| \quad \forall v \in V$$

- $a : V^2 \rightarrow \mathbb{R}$ est une forme bilinéaire, c'est à dire que a est linéaire en ses deux variables.
- a est continue, c'est à dire qu'il existe une constante de continuité M telle que :

$$|a(f, g)| \leq M \|f\| \cdot \|g\| \quad \forall f, g \in V$$

- a est coercive, c'est à dire qu'il existe une constante de coercivité ν telle que :

$$a(f, f) \geq \nu \|f\|^2 \quad \forall f \in V$$

alors le problème (2.1) admet une solution unique $f \in V$, qui dépend continument de L .

2.1.4 Approximation interne

L'approximation interne est un moyen général de résoudre les EDP dans un espace de Sobolev $H(\Omega)$. Les EDP qui nous intéressent prennent la forme générale :

$$\text{Trouver } f \in H(\Omega) \text{ tel que } D(f) = r \quad (2.2)$$

où $D : H(\Omega) \rightarrow H(\Omega)$ est un opérateur différentiel, et $r \in H(\Omega)$ une fonction arbitraire. Ce problème admet la formulation variationnelle suivante :

$$\text{Trouver } f \in H(\Omega) \text{ tel que } \langle D(f), g \rangle = \langle r, g \rangle \quad \forall g \in H(\Omega) \quad (2.3)$$

La théorie de Lax-Milgram donne les conditions sur $a(f, g) = \langle D(f), g \rangle$ et $L(g) = \langle r, g \rangle$ pour que le problème (2.3) admette une solution unique. L'approximation interne de (2.3) est double :

1. On remplace l'espace de Sobolev (généralement de dimension infinie) $H(\Omega)$ dans lequel on recherche la solution par un espace *solution* V_n^s de dimension n finie.
2. On remplace l'espace de Sobolev (généralement de dimension infinie) $H(\Omega)$ sur lequel l'EDP est projetée par un espace *test* V_m^t de dimension m finie.

de sorte que l'approximation interne de (2.3) est :

$$\text{Trouver } f_n \in V_n^s \text{ tel que } a(f_n, g_m) = L(g_m) \quad \forall g_m \in V_m^t \quad (2.4)$$

qui admet une solution unique sous les conditions de Lax-Milgram. Soient $\{\Phi_s^i\}_{i=1..n}$ une base de V_n^s , et $\{\Phi_t^j\}_{j=1..m}$ une base de V_m^t , alors $f_n = \sum_i f_i \Phi_s^i$ et $g_m = \sum_j g_j \Phi_t^j$, et le problème (2.4) est équivalent à :

$$\sum_i f_i a(D(\Phi_s^i), \Phi_t^j) = L(\Phi_t^j) \quad \forall j = 1..m$$

En notant $\mathbf{f}_n = (f_1, \dots, f_n)^t$ et $\mathbf{g}_m = (g_1, \dots, g_m)^t$, on aboutit à la forme matricielle :

$$A\mathbf{f}_n = \mathbf{I}_m$$

où A est une matrice (n, m) de terme $A_i^j = a(\Phi_s^i, \Phi_t^j)$ et \mathbf{I}_m un vecteur de terme $(\mathbf{I}_m)_j = L(\Phi_t^j)$. La condition de coercivité sur $a(\cdot, \cdot)$ implique le caractère défini positif de A . La plupart du temps, pour des raisons de simplicité et de symétrie évidentes, on choisit $n = m$ et même souvent $V_n^h = V_m^t = V_n$. Dans ce cas A est carrée et si $a(\cdot, \cdot)$ est symétrique, A l'est aussi.

Supposons maintenant $V_n^h = V_m^t = V_n$. Remplacer $H(\Omega)$ par V_n est évidemment une approximation qui dépend de la façon dont V_n approxime $H(\Omega)$. Pour quantifier cette approximation, on dispose d'un lemme utile :

Lemme 2.1.3 (de Céa) *Soit f une solution au problème (2.3) et f_n une solution au problème (2.4) sous les conditions de Lax-Milgram. Alors on a :*

$$\|f - f_n\| \leq \frac{M}{\nu} \inf_{f'_n \in V_n} \|f - f'_n\|$$

De plus, si $a(\cdot, \cdot)$ est symétrique, on a :

$$\|f - f_n\| \leq \sqrt{\frac{M}{\nu}} \inf_{f'_n \in V_n} \|f - f'_n\|$$

Notons qu'à cause de leurs définitions, $M \geq \nu$. Ce lemme donne des indications sur la "qualité" que l'on désire pour la matrice A pour que cette approximation soit bonne.

Nous voulons maintenant caractériser le fait que quand n augmente, V_n approxime $H(\Omega)$ de plus en plus finement. Tout d'abord, nous avons besoin d'un opérateur de projection $r_n : H(\Omega) \rightarrow V_n$ qui vérifie $\|f - r_n(f)\| = \min_{f'_n \in V_n} \|f - f'_n\|$ (le minimum n'est pas nécessairement unique). Alors, l'idée que V_n approxime $H(\Omega)$ de plus en plus finement quand n augmente peut s'écrire :

$$\lim_{n \rightarrow \infty} \|f - r_n(f)\| = 0 \quad \forall f \in V \tag{2.5}$$

et nous avons le résultat de convergence suivant :

Théorème 2.1.4 (Convergence de l'approximation interne) *Sous les conditions de Lax-Milgram, et étant donnés des espaces de Hilbert V_n et des opérateurs de projection r_n vérifiant (2.5), l'approximation interne converge :*

$$\lim_{n \rightarrow \infty} \|f - f_n\| = 0$$

où f est solution de (2.3) et f_n est solution de (2.4).

Alors que les conditions de Lax-Milgram sont des conditions sur le problème lui-même, nous avons maintenant des conditions sur les espaces V_n sous lesquelles la solution approchée converge vers la solution réelle. En conséquence, pour approcher la solution d'une EDP, on doit construire de tels espaces de fonctions V_n avec ces propriétés.

2.1.5 Galerkin et les éléments finis

Si un espace de Hilbert V est séparable (ce qui sera toujours notre cas), alors il admet une base de Hilbert (généralement infinie) $(\Phi^i)_{i \geq 1}$. A partir de cette base, on peut construire une séquence d'espaces de

Hilbert $V_n = \text{span}(\Phi^1, \dots, \Phi^n)$, et r_n est la projection orthogonale sur cette base. Les résultats ci-dessus montrent qu'alors la solution du problème (2.4) converge vers celle de (2.3). Cette méthode appelée "méthode de Galerkin" est intéressante du point de vue théorique. Cependant, elle est difficile à mettre en pratique car les matrices A_n sont alors généralement pleines (pas de termes nuls qui simplifient les calculs) et mal conditionnées.

Le principe de la méthode des éléments finis est aussi de construire des espaces d'approximation interne V_n des espaces de Sobolev $H^1(\Omega)$ ou $H_0^1(\Omega)$. Cependant, la méthode est différente puisqu'elle s'appuie sur la notion de maillage M de l'espace Ω sur lequel les fonctions de bases sont définies. Les fonctions de base Φ^i sont alors définies par morceaux sur les éléments du maillage (généralement des triangles), et non nulles sur très peu de ces éléments. Ainsi, ces fonctions sont très localisées, de sorte que les matrices A_n sont creuses (beaucoup de termes nuls) ce qui rend les calculs numériques beaucoup plus rapides.

Dans notre cas, Ω est une surface triangulée. Ceci est à la fois un avantage et un inconvénient : c'est un avantage car on dispose déjà d'un maillage de notre domaine, et on a donc pas à le calculer. C'est un inconvénient car ce maillage peut avoir de mauvaises propriétés, entraînant une mauvaise approximation des solutions. La prochaine section décrit les bases de fonctions les plus utilisées sur ces maillages.

Dans la deuxième partie de cette thèse, nous décrirons comment construire une base différente de ces espaces de fonctions basés sur les maillages, et les applications de cette base. Dans la dernière partie, on s'attache à construire un maillage différent de notre surface triangulée afin d'avoir de meilleures propriétés pour les fonctions qu'on pourra y installer. Cette dernière approche essaiera en particulier de capturer les structures géométriques et topologiques globales de notre maillage pour y construire des bases de fonctions plus adéquates.

2.2 Bases de fonctions interpolantes

Les bases de fonctions interpolantes sont les bases de fonctions les plus répandues. Elles sont simples à définir et à interpréter.

Définition 11 (Bases de fonctions interpolantes) Soit $\{\mathbf{x}_j^c \in \Omega \subset \mathbb{R}^d\}_{j=1..m}$ un ensemble fini de points appelés points de contrôle. Une base de fonctions interpolantes $\{\Phi^i\}_{i=1..n}$ sur Ω avec les points de contrôle $\{\mathbf{x}_i^c\}$ est une base de fonction qui vérifie :

$$\Phi^i(\mathbf{x}_j^c) = \delta_{ij} = 1 \text{ si } i = j \text{ sinon } 0$$

Définition 12 (Interpolation) Soit f une fonction sur Ω et $\{\Phi^i\}_{i=1..n}$ une base de fonctions interpolantes. Alors l'interpolation de f est la fonction :

$$I_\Phi(f)(\mathbf{x}) = \sum_i f(\mathbf{x}_i^c) \Phi^i(\mathbf{x})$$

L'interpolation $I_\Phi(f)$ d'une fonction f prend les mêmes valeurs que f aux points de contrôle : $I_\Phi(f)(\mathbf{x}_i^c) = f(\mathbf{x}_i^c)$

Remarque 2.2.1 (Interpolation vs Extrapolation) L'interpolation $I_\Phi(f)$ n'est pas la projection orthogonale de f sur $\text{span}(\Phi^i)$, qu'on appelle extrapolation $E_\Phi(f)$ et qui minimise $\|f - E_\Phi(f)\|$. Cependant, l'extrapolation $E_\Phi(f)$ d'une fonction f ne coïncide en général pas avec f aux points de contrôles : $E_\Phi(f)(\mathbf{x}_i^c) \neq f(\mathbf{x}_i^c)$

Ce dernier point est important dans le contexte des éléments finis : les éléments finis cherchent à approcher la solution d'un problème au sens de la norme L_2 , c'est à dire à l'extrapoler. Choisir une base de fonctions interpolantes pour appliquer la méthode des éléments finis donne l'impression trompeuse que l'on cherche à interpoler la solution. En réalité, on ne cherche pas à approximer la valeur de la solution aux points de contrôle mais à définir ces valeurs de sorte à minimiser la norme de l'erreur, c'est à dire l'erreur d'approximation globale (au sens intégral) de la solution.

Nous allons maintenant définir les éléments P_n (aussi appelés polynômes d'interpolation de Lagrange) qui sont les bases de fonctions interpolantes les plus répandues en éléments finis.

2.2.1 Éléments P_n

Les éléments P_n sont les plus simples des bases de fonctions. Nous allons commencer par les étudier en une dimension et présenter leur propriétés, puis étendre cette construction aux surfaces triangulées.

Éléments P_n en 1D

En 1D, les éléments P_n interpolent une fonction f connue par ses valeurs $f_i = f(x_i)$ en $n + 1$ points de contrôle x_0, \dots, x_n . La fonction interpolée est $P_n f = \sum_{i=0}^n f_i P_n^i(x)$ où :

$$P_n^i = \prod_{k=0, k \neq i}^n \frac{x - x_k}{x_i - x_k}$$

est le seul polynôme d'ordre n qui vérifie $P_n^i(x_j) = 1$ si $i = j$ et 0 sinon. Ainsi, $P_n f$ est le seul polynôme d'ordre n qui vérifie $P_n f(x_i) = f_i \forall i = 0..n$.

Pour approximer une fonction $f : \Omega \subset \mathbb{R} \rightarrow \mathbb{R}$ donnée, on doit donc découper Ω en m intervalles I^m , sur lesquels f sera P_n -interpolée. Pour faire simple, on choisit de subdiviser $I_m = [x_0^m, x_n^m]$ de façon régulière en choisissant $x_i^m = x_0^m + i(x_n^m - x_0^m)/n$. On a alors un résultat important sur l'erreur d'interpolation :

Théorème 2.2.2 (Erreur d'interpolation P_n) Pour toute fonction $f \in C^{n+1}([x_0, x_n])$, on a :

$$\forall x \in [x_0, x_n] \quad \exists \chi \in [x_0, x_n] \text{ tel que } f(x) - P_n f(x) = \frac{f^{(n+1)}(\chi)}{n+1!} \prod_{k=0}^n (x - x_k)$$

Ce théorème nous permet de donner une borne supérieure sur l'erreur d'interpolation :

$$\forall x \in [x_0, x_n] \quad |f(x) - P_n f(x)| < \frac{(x_n - x_0)^{n+1}}{4(n+1)!^2} \sup_{x \in [x_0, x_n]} f^{(n+1)}(x)$$

Cela implique entre autre que l'interpolation P_n est exacte sur les polynômes d'ordre n (pour lesquels $f^{(n+1)}(x) = 0$). Cela implique aussi que l'erreur d'approximation P_n d'une fonction f dépend de combien f diffère d'un polynôme d'ordre n .

Cette borne nous permet de comparer la subdivision avec l'interpolation d'ordre supérieur dans un cas simple. Pour une fonction f définie sur $[x_0, x_2]$ (avec $x_1 = (x_0 + x_2)/2$), comparons l'erreur entre :
 - l'interpolation P_1 sur $[x_0, x_1]$ et $[x_1, x_2]$. Sur les deux intervalles :

$$|f(x) - P_1 f(x)| < \frac{(x_1 - x_0)^2}{16} \sup_{x \in [x_0, x_n]} f^{(2)}(x) = Err_1$$

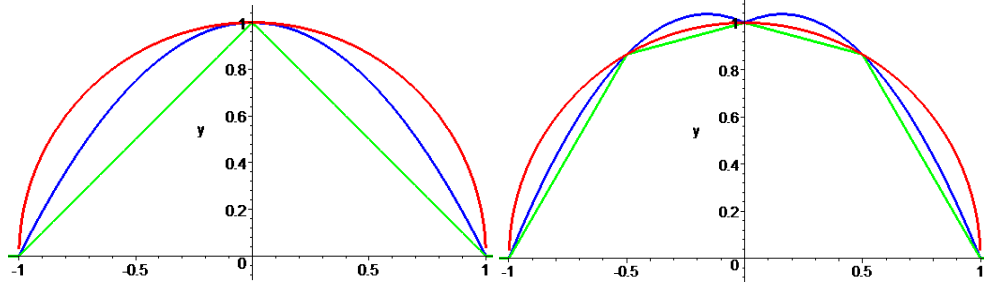


FIG. 2.1 – Comparaison entre l’interpolation P_2 (bleu) et P_1 (vert) sur un demi cercle (rouge) pour le même nombre de degrés de liberté. **Gauche** : avec une subdivision, les erreurs L_2 sont 0,0438 pour P_2 et 0,1917 pour P_1 . **Droite** : Avec deux subdivisions, les erreurs L_2 sont 0,0204 pour P_2 et 0,0397 for P_1 .

– l’interpolation P_2 sur $[x_0, x_2]$ avec les points de contrôle $x_i, i = 0, 1, 2$:

$$|f(x) - P_2 f(x)| < \frac{(x_2 - x_0)^3}{144} \sup_{x \in [x_0, x_n]} f^{(3)}(x) = Err_2$$

Le ratio des erreurs est :

$$\frac{Err_2}{Err_1} = \frac{4(x_1 - x_0) \sup_{x \in [x_0, x_n]} f^{(3)}(x)}{9 \sup_{x \in [x_0, x_n]} f^{(2)}(x)}$$

pour P_2 par rapport à P_1 . Regardons cela sur un sinus de longueur d’onde λ : $f(x) = \sin(2\pi x/\lambda)$. Dans ce cas, on a $Err_2/Err_1 = 8\pi(x_2 - x_0)/18\lambda$. Mais le théorème d’échantillonnage de Nyquist nous dit que l’on doit avoir $(x_2 - x_0) < \lambda/2$, auquel cas $Err_2/Err_1 < 8\pi/36 \approx 0,7$. Cela veut dire que même à la limite de Nyquist, l’interpolation P_2 est déjà meilleure que l’interpolation P_1 , pour le même nombre de degrés de liberté. De plus, elle s’améliore quand λ diminue. Comme n’importe quelle fonction peut être décomposée en somme de fonctions sinus par la transformée de Fourier, cette considération paraît assez générale pour justifier l’étude des interpolants d’ordre supérieurs à 1. La Figure 2.1 illustre cela sur un exemple simple (interpolation d’un demi cercle), où l’erreur L_2 est plus faible en P_2 pour le même nombre de degrés de liberté.

Éléments P_n sur un maillage

Sur un maillage $M = \langle \mathcal{V}, \mathcal{E}, \mathcal{F} \rangle$, nous étendons les éléments P_n en nous basant sur les coordonnées barycentriques et la notion de n -grille. Appelons $\lambda_i(\mathbf{x})$ les coordonnées barycentriques d’un point \mathbf{x} relatives au vertex i , de sorte que si \mathbf{x} est dans le triangle $[ijk]$:

$$\mathbf{x} = \lambda_i(\mathbf{x})\mathbf{x}_i + \lambda_j(\mathbf{x})\mathbf{x}_j + \lambda_k(\mathbf{x})\mathbf{x}_k$$

Comme dans le cas 1D, on définit les fonctions de base comme les polynômes (en coordonnées barycentriques) du plus petit ordre possible qui valent 1 sur un point de contrôle et 0 sur les autres. La seule différence est que nous subdivisons maintenant un triangle et plus un simple intervalle. On doit donc définir une grille régulière sur le triangle, qu’on appelle n -grille. La n -grille d’un triangle $[ijk]$ est l’ensemble des points de $[ijk]$ dont les coordonnées barycentriques sont multiples de $1/n$: $[n_i/n, n_j/n, n_k/n]$ avec $n_i + n_j + n_k = n$. Sur un maillage, les éléments P_n sont définis comme les polynômes d’interpolation sur la n -grille. $P_n(n_i, n_j, n_k)$ est le polynôme d’ordre n en $\lambda_i, \lambda_j, \lambda_k$ qui prend la valeur 1 en $[n_i/n, n_j/n, n_k/n]$

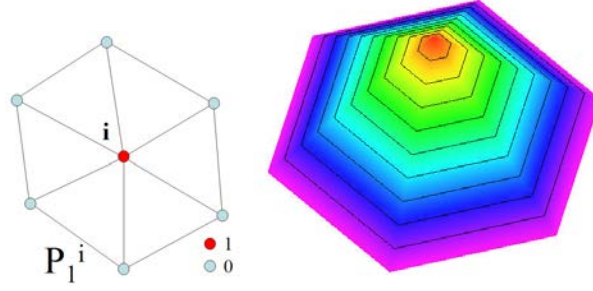


FIG. 2.2 – Les éléments P_1 (fonctions chapeau) sont non nulles sur l'étoile de leur point de contrôle.

et 0 sur les autres points de la n -grille :

$$P_n^{n_i, n_j, n_k} = \prod_{n'_i=0}^{n_i-1} \frac{n\lambda_i - n'_i}{n_i - n'_i} \prod_{n'_j=0}^{n_j-1} \frac{n\lambda_j - n'_j}{n_j - n'_j} \prod_{n'_k=0}^{n_k-1} \frac{n\lambda_k - n'_k}{n_k - n'_k}$$

qui est de degrés $n_i + n_j + n_k = n$. Ces fonctions ont pour support $St_2([i^{n_i} j^{n_j} k^{n_k}])$, où $i^{n_i} = \emptyset$ si $n_i = 0$ et i sinon. En d'autres termes, les support est réduit à $[ijk]$ si $n_i, n_j, n_k > 0$, il est composé des deux triangles se partageant ij si $n_k = 0$, et de toute l'étoile de i si $n_j = n_k = 0$. Nous allons maintenant présenter en détail les 3 premiers éléments P_n .

Les éléments P_1 sont simplement les coordonnées barycentriques λ_i dont le support est $St_2(i)$ (voir Figure 2.2). On les appelle aussi "fonctions chapeau" à cause de la forme qu'elles prennent sur $St_2(i)$. On peut donner une expression explicite pour $\lambda_i(\mathbf{x})$: dans le triangle $[ijk]$, $\lambda_i(\mathbf{x})$ est le ratio entre $\|i\mathbf{x}\|$ et $\|ijk\|$:

$$\lambda_{i,jk}(\mathbf{x}) = \frac{(|jk| \wedge [j]\mathbf{x}) \cdot \mathbf{n}}{2\|ijk\|} \quad (2.6)$$

où \mathbf{n} est la normale au triangle $[ijk]$. Ces fonctions sont affines par morceaux, donc leur gradient est constant par morceaux sur les triangles :

$$\nabla \lambda_{i,jk} = \frac{|jk|^\perp}{2\|ijk\|} \quad (2.7)$$

où $|jk|^\perp$ est le vecteur $|jk|$ tourné de $+\pi/2$ autour de la normale.

Les éléments P_2 sont les deux familles (cf Figure 2.3) :

$$P_2^i = (2\lambda_i - 1)\lambda_i \quad P_2^{ij} = 4\lambda_i\lambda_j \quad (2.8)$$

et ont pour gradients :

$$\nabla P_2^i = \frac{(4\lambda_i - 1)|jk|^\perp}{2\|ijk\|} \quad \nabla P_2^{ij} = \frac{2(\lambda_i|ki|^\perp + \lambda_j|jk|^\perp)}{\|ijk\|} \quad (2.9)$$

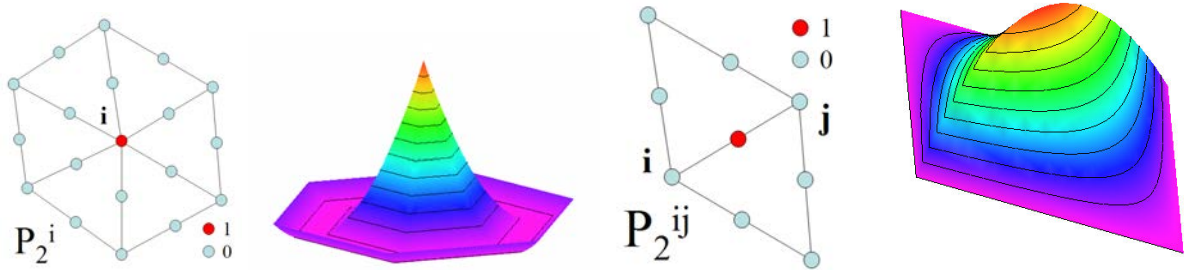


FIG. 2.3 – **Gauche** : Les éléments P_2^i valent 0 au milieu des arêtes. **Droite** : Les éléments P_2^{ij} sont non nuls sur une paire de triangles et valent 1 au milieu de l'arête partagée

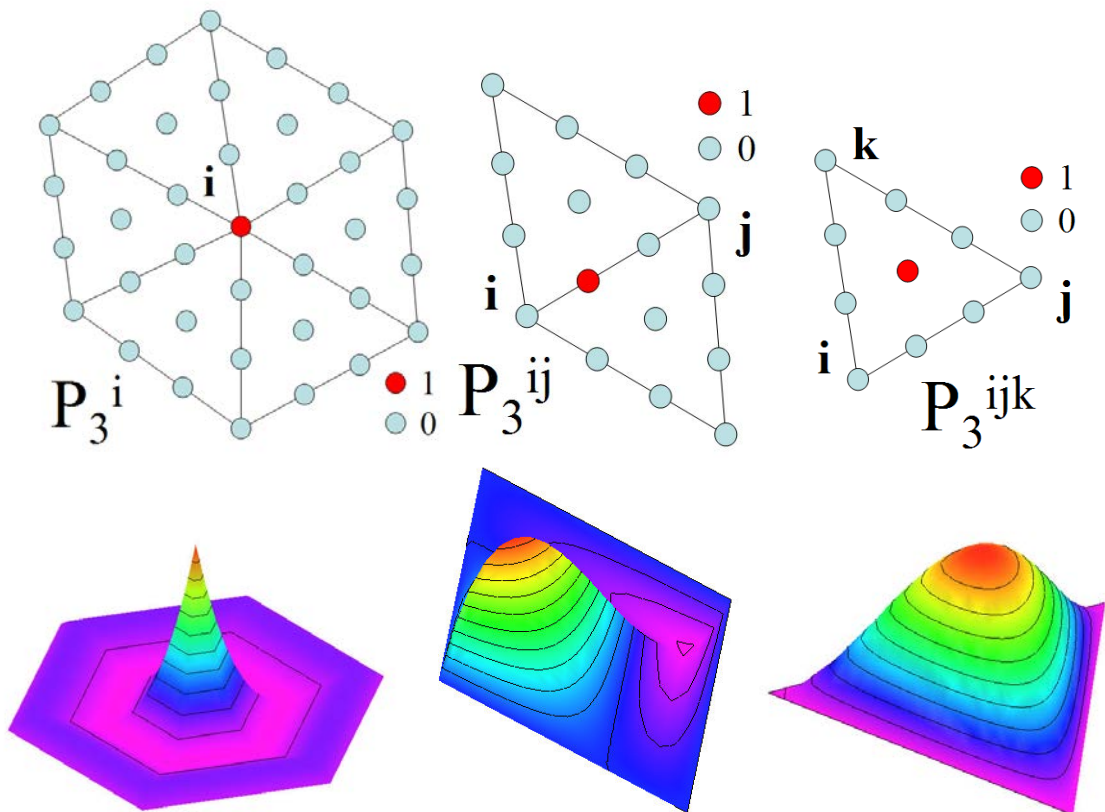


FIG. 2.4 – **Gauche** : Les éléments P_3^i sont nuls au tiers et aux deux tiers des arêtes. **Milieu** : Les éléments P_3^{ij} sont non nuls sur une paire de triangle seulement et valent 1 et 0 aux tiers et deux tiers de l'arête partagée. **Droite** : Les éléments P_3^{ijk} sont non nuls seulement sur le triangle ijk , et valent 1 au barycentre $g(ijk)$ du triangle.

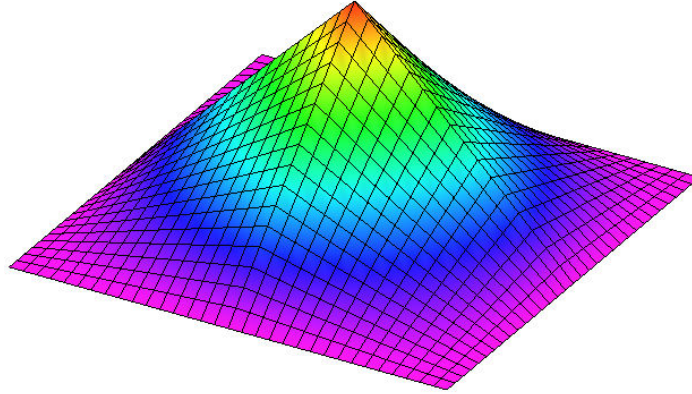


FIG. 2.5 – Une fonction de base bilinéaire

Les éléments P_3 sont les trois familles (cf Figure 2.4) :

$$P_3^i = (3\lambda_i - 2)(3\lambda_i - 1)\lambda_i/2 \quad P_3^{ij} = 9\lambda_i\lambda_j(3\lambda_i - 1)/2 \quad P_3^{ijk} = 27\lambda_i\lambda_j\lambda_k \quad (2.10)$$

et ont pour gradients :

$$\begin{aligned} \nabla P_3^i &= (12\lambda_i - 1)(3\lambda_i - 1) \frac{|jk|^\perp}{4||ijk||} & \nabla P_3^{ij} &= 9 \frac{\lambda_i(3\lambda_i - 1)|ki|^\perp + \lambda_j(6\lambda_i - 1)|jk|^\perp}{4||ijk||} \\ \nabla P_3^{ijk} &= 27 \frac{\lambda_i\lambda_j|ij|^\perp + \lambda_j\lambda_k|jk|^\perp + \lambda_k\lambda_i|ki|^\perp}{2||ijk||} \end{aligned} \quad (2.11)$$

Erreur d'interpolation Pour les bases de fonction P_n sur un maillage, on a les mêmes bornes d'erreur qu'en 1D, mais en remplaçant la longueur de l'intervalle $(x_n - x_0)$ par le diamètre du cercle circonscrit au triangle. Cela veut dire que la précision d'une méthode P_n sera limitée par le plus grand diamètre des cercles circonscrits aux triangles du maillage. Pour avoir plus de précision, il faut soit remailler/subdiviser le maillage, soit installer des bases d'ordre supérieur. La subdivision est plus simple mais moins précise que le passage à l'ordre supérieur. Le remaillage est un problème complexe qui sera abordé dans la troisième partie de la thèse.

2.2.2 Interpolation bilinéaire

Si l'on dispose d'un maillage quadrangulé, on peut le casser en triangles (de façon plus ou moins arbitraire) pour y installer des éléments P_n . Cependant, on peut mieux tirer parti de la structure quadrangulaire en installant une base de fonction adaptée : celle de l'interpolation bilinéaire.

Base bilinéaire

Nous appelons $B_{ij}(u, v) : [0, 1]^2 \rightarrow \mathbb{R}$, $i, j \in \{0, 1\}$ la base de fonctions interpolantes bilinéaires définies par :

$$B_{00} = (1 - u)(1 - v) \quad B_{10} = u(1 - v) \quad B_{01} = (1 - u)v \quad B_{11} = uv$$

Soient $\mathbf{x}_{ij} : i, j \in \{0, 1\}$ quatre points de \mathbb{R}^3 . On définit le morceau bilinéaire S par son équation paramétrique :

$$S = \{\mathbf{x}(u, v) = \sum_{i,j} \mathbf{x}_{ij} B_{ij}(u, v) \mid (u, v) \in [0, 1]^2\}$$

Une fonction bilinéaire f connue par ses valeurs f_{ij} en \mathbf{x}_{ij} peut alors être définie indirectement sur S par :

$$f(\mathbf{x}(u, v)) = f_{uv}(u, v) = \sum_{i,j} f_{ij} B_{ij}(u, v)$$

Un maillage quadrangulé est la réunion de tels morceaux bilinéaires, et une fonction de base bilinéaire basée sur un sommet du maillage est ainsi non nulles sur les quatre morceaux autour de ce sommet (cf Figure 2.5).

Gradient et Jacobien

Les gradients de ces fonctions sont :

$$\nabla B_{00} = \begin{pmatrix} v-1 \\ u-1 \end{pmatrix} \quad \nabla B_{10} = \begin{pmatrix} 1-v \\ -u \end{pmatrix} \quad \nabla B_{01} = \begin{pmatrix} -v \\ 1-u \end{pmatrix} (1-u)v \quad \nabla B_{11} = \begin{pmatrix} v \\ u \end{pmatrix}$$

Définissons $\mathbf{v}_{uj} = \mathbf{x}_{0j} \mathbf{x}_{1j}$ et $\mathbf{v}_{iv} = \mathbf{x}_{i0} \mathbf{x}_{i1}$. Le Jacobien de la paramétrisation est le gradient de $\mathbf{x}(u, v)$ in (u, v) :

$$J(u, v) = \nabla_{uv} \mathbf{x}(u, v) = (\partial_u \mathbf{x}, \partial_v \mathbf{x})^t = ((1-v)\mathbf{v}_{u0} + v\mathbf{v}_{u1}, (1-u)\mathbf{v}_{0v} + u\mathbf{v}_{1v})^t$$

On note J^{-1} l'inverse à droite de J qui s'écrit :

$$J^{-1} = \frac{1}{\text{Det}(J)} \left(-(\partial_v \mathbf{x})^\perp, (\partial_u \mathbf{x})^\perp \right) = \frac{1}{\text{Det}(J)} J^\perp$$

où \perp dénote la rotation de $\pi/2$ autour de la normale $\partial_u \mathbf{x} \times \partial_v \mathbf{x}$, et $\text{Det}(J) = -(\partial_v \mathbf{x})^\perp \cdot \partial_u \mathbf{x} = (\partial_u \mathbf{x})^\perp \cdot \partial_v \mathbf{x}$. Cela permet d'écrire le gradient des fonctions de base sur S sous la forme $\nabla_S B_{ij} = J^{-1} \nabla B_{ij}$

2.3 Bases mobiles

Les fonctions que l'on cherche à décrire en informatique graphique présentent parfois des variations importantes dans certaines zones, tout en étant très lisses dans d'autres. Par exemple, l'illumination d'une scène peut être constante dans de larges zones, et très être très détaillées au bord des ombres. Pour capturer ce comportement, il est alors souhaitable d'adapter la triangulation au niveau de détail. Ainsi, l'idée d'autoriser une triangulation de Ω d'évoluer dynamiquement pour s'adapter à la fonction à approximer mérite d'être étudiée. Cela permettra de réduire de façon significative le nombre de triangles requis pour une précision donnée.

Jusqu'ici, nous avons supposé que nos points de contrôle étaient donnés *a priori*, et nous installions des bases de fonctions sur un maillage fixé. On peut se demander ce qu'il se passe si on introduit les positions des points de contrôle comme nouvelles variables. Cela pourrait autoriser la densité de points de contrôle à s'adapter à la précision requise. Nous nous proposons de formaliser ce problème, et donnons deux résultats dans des cas très simples.

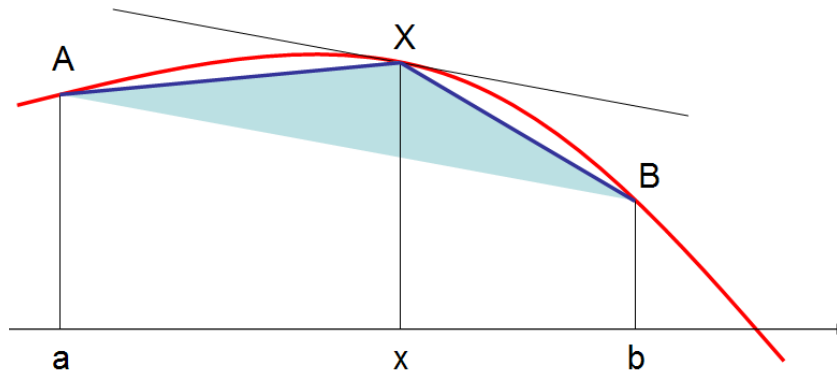


FIG. 2.6 – L'aire entre les courbes rouges et bleues est une constante moins l'aire du triangle AXB. Elles est donc minimale quand l'aire de AXB est maximale.

2.3.1 Problème

Nous voulons toujours résoudre notre EDP générique :

$$\text{Trouver } f \in V \text{ tel que } D(f) = r$$

mais la discrétisation de l'espace de base Ω de f peut maintenant évoluer. Les coordonnées des points de contrôle sont maintenant des variables du problème. Plusieurs difficultés apparaissent quand on essaie de résoudre ce problème, en particulier parce que l'espace de fonction dans lequel nous travaillons n'est plus un espace de Hilbert. On ne peut donc pas s'appuyer sur les résultats habituels de la théorie des éléments finis. Nous allons donc seulement nous attaquer au problème de l'approximation de fonction (D est l'identité), dans deux cas simples.

2.3.2 Interpolation P_1 des fonctions convexes en 1D

Nous supposons que nous interpolons une fonction convexe $f : [a, b] \rightarrow \mathbb{R}$ avec la base des éléments P_1 :

$$P_1 f(x) = \sum_i f(x_i) P_1^i(x)$$

Comme f est convexe, on a nécessairement $P_1 f(x) \leq f(x)$, et on peut donc évaluer l'erreur d'interpolation comme l'aire entre les graphes de f et $P_1 f$:

$$Err = \int_{[a,b]} f(x) - P_1 f(x) dx$$

Sans perte de généralité, on peut calculer l'erreur minimale en supposant qu'il n'y a qu'un point de contrôle x entre a et b . Un simple argument géométrique (cf Figure 2.6) nous dit que cette erreur est minimale quand le triangle de sommets $A = (a, f(a))$, $X = (x, f(x))$ and $B = (b, f(b))$ est d'aire maximale. Analytiquement, cela correspond à $f'(x) = (f(b) - f(a))/(b - a)$, qui a une seule solution quand f est convexe (puisque f' est alors monotone). C'est ce choix qui est fait par la "méthode de la corde" qui polygonalise une courbe de façon récursive en choisissant les points les plus éloignés des segments existants.

2.3.3 Approximation par fonctions génériques en 1D

Nous supposons maintenant que nous avons une fonction générique $T(x)$. Une version de ce générique peut être instanciée au point x_i par $T(x - x_i)$. On veut alors approximer une fonction $f : \mathbb{R} \rightarrow \mathbb{R}$ par une somme de n instances :

$$T_n f = \sum_i a_i T(x - x_i)$$

de sorte que l'erreur :

$$Err = \int_{\mathbb{R}} (f(x) - \sum_i a_i T(x - x_i))^2 dx$$

soit minimale. Si l'on note C la matrice (symétrique) de corrélation de terme $C_i^j = \int_{\mathbb{R}} T(x - x_i) T(x - x_j) dx$, et \mathbf{f} le vecteur de terme $f_i = \int_{\mathbb{R}} f(x) T(x - x_i) dx$, l'erreur se réécrit :

$$Err = \mathbf{a}^t C \mathbf{a} - 2 \mathbf{a}^t \mathbf{f} + c_{nst}$$

où \mathbf{a} est le vecteur des a_i . Cette erreur est minimale pour $\mathbf{a} = C^{-1} \mathbf{f}$, et vaut alors :

$$Err = -\mathbf{f}^t C^{-1} \mathbf{f} + c_{nst}$$

qui a pour dérivée :

$$\frac{\partial Err}{\partial x_i} = \mathbf{f}^t C^{-1} \frac{\partial C}{\partial x_i} C^{-1} \mathbf{f} - 2 \mathbf{f}^t C^{-1} \frac{\partial \mathbf{f}}{\partial x_i}$$

Les dérivées de C et \mathbf{f} sont alors :

$$\frac{\partial C_i^j}{\partial x_i} = - \int_{\mathbb{R}} T'(x - x_i) T(x - x_j) dx \quad \frac{\partial f_i}{\partial x_i} = - \int_{\mathbb{R}} f(x) T'(x - x_i) dx$$

Cela permet de minimiser l'erreur par un gradient conjugué par exemple. Nous n'avons pas pu explorer cette approche plus en profondeur, mais nous pensons qu'elle peut être aisément étendue aux dimensions supérieures, en utilisant plus d'une fonction générique, et/ou avec des génériques paramétrées (par exemple la largeur d'une ondelette, l'orientation d'une bandelette,...)

2.3.4 Conclusion

Le problème général des bases de fonctions mobiles est à nos yeux un problème très ouvert qui mérite d'être étudié. Les deux résultats que nous proposons ici donnent une idée du type d'outils que l'on peut utiliser pour le résoudre. Nous espérons que cela pourra servir de bases à des développements plus poussés.

3

Le calcul extérieur continu et son équivalent discret

Sommaire

3.1	Calcul extérieur dans le cadre continu	34
3.1.1	Tenseurs	34
3.1.2	<i>k</i> -chaines et <i>k</i> -formes alternées	35
3.1.3	Produit extérieur	36
3.1.4	<i>k</i> -chaines et <i>k</i> -formes différentielles	36
3.1.5	Dérivée extérieure	37
3.1.6	Intégration et dualité	39
3.1.7	Étoile de Hodge	41
3.1.8	Produit interne	41
3.1.9	Conclusion	42
3.2	Le calcul extérieur discret	42
3.2.1	Formes, chaines et intégration discrètes	42
3.2.2	Orientation	43
3.2.3	Dérivée extérieure et opérateur de bord	44
3.2.4	(Co)homologie	44
3.2.5	Complexe dual	45
3.2.6	Étoile de Hodge	46
3.2.7	Produit interne	47
3.2.8	Applications de De Rham et Whitney	48
3.2.9	Calcul extérieur et dualités	48

Introduction

Ce chapitre présente le formalisme du calcul extérieur dans le cadre continu et discret. Le calcul extérieur peut être vu comme une théorie permettant d'appréhender les notions de différentiation et d'intégration d'objets de dimension quelconque sur des variétés de dimension quelconque. Comme on le verra, le calcul extérieur continu et son équivalent discret présentent des difficultés différentes et sont très complémentaires, contrairement au cas de la méthode des éléments finis qui est une pure discrétisation des concepts du continu.

Organisation du chapitre

Nous commencerons par présenter la théorie du calcul extérieur dans la Section 3.1. Nous expliquerons ensuite comment construire son équivalent discret dans la Section 3.2.

Contributions

- Nous donnons une présentation concise mais complète de la théorie du calcul extérieur.
- Nous discutons la notion d'intégration sur une chaîne.
- Nous expliquons comment se construit le calcul extérieur discret.
- Nous discutons des aspects importants des relations subtiles qu'entretiennent le calcul extérieur et son équivalent discret, et en particulier leurs dualités.

3.1 Calcul extérieur dans le cadre continu

3.1.1 Tenseurs

Soient V et W deux espaces vectoriels sur \mathbb{R} avec pour bases $\{v_i\}_{i=1..n_V}$ et $\{w_j\}_{j=1..n_W}$ (n_V et n_W sont les dimensions de V et W). Leur produit tensoriel :

$$V \otimes W$$

est un espace vectoriel de dimension $n_V \cdot n_W$ sur \mathbb{R} avec une application bilinéaire :

$$\otimes : V \times W \rightarrow V \otimes W$$

qui est universelle (i.e., la plus petite possible sans perdre d'information). Le produit tensoriel $V \otimes W$ peut être construit comme l'espace vectoriel généré par une base

$$\{v_i \otimes w_j\}$$

Pour $v, v' \in V$, $w, w' \in W$, $r \in \mathbb{R}$, le produit tensoriel vérifie les propriétés de linéarité suivantes :

$$(v + v') \otimes w = v \otimes w + v' \otimes w$$

$$v \otimes (w + w') = v \otimes w + v \otimes w'$$

$$r(v \otimes w) = (rv) \otimes w = v \otimes (rw)$$

Ce produit tensoriel permet de définir les tenseurs dans \mathbb{R}^d . Soit $\{e_i\}_{i=1..d}$ la base canonique de \mathbb{R}^d . L'espace dual \mathbb{R}^{d*} de \mathbb{R}^d est défini comme l'espace vectoriel des applications linéaires de $\mathbb{R}^d \rightarrow \mathbb{R}$. Il a une base (duale) naturelle $\{e^j\}_{j=1..d}$ qui vérifie :

$$e^j(e_i) = \delta_{ij} = 1 \text{ si } i = j \text{ sinon } 0 \tag{3.1}$$

Un tenseur (n, m) est alors défini comme un élément (i.e., un vecteur) de l'espace vectoriel :

$$T_n^m(\mathbb{R}^d) = \underbrace{V \otimes \dots \otimes V}_m \otimes \underbrace{V^* \otimes \dots \otimes V^*}_n$$

$T_n^m(\mathbb{R}^d)$ est un espace vectoriel de dimension d^{n+m} avec comme vecteurs de base :

$$\{\mathbf{e}_{i_1} \otimes \dots \otimes \mathbf{e}_{i_n} \otimes \mathbf{e}^{j_1} \otimes \dots \otimes \mathbf{e}^{j_m}\} = \{\mathbf{e}_{i_1, \dots, i_n}^{j_1, \dots, j_m}\}$$

Un tenseur (n, m) α est indexé par n coordonnées *covariantes* (en indice) et m coordonnées *contravariantes* (en exposant) :

$$\alpha = \sum \alpha_{i_1, \dots, i_n}^{j_1, \dots, j_m} \mathbf{e}_{i_1, \dots, i_n}^{j_1, \dots, j_m}$$

Les exemples les plus simples sont :

- $T_0^0(V)$: scalaires
- $T_0^1(V)$: vecteurs
- $T_1^0(V)$: applications linéaires
- $T_1^1(V)$: matrices
- $T_2^0(V)$: produits scalaires

Les tenseurs vont maintenant nous servir à définir des objets plus "géométriques" : les formes et chaînes alternées.

3.1.2 k -chaînes et k -formes alternées

Les k -chaînes et k -formes alternées sont les éléments de base du calcul extérieur, qui généralisent les fonctions en des objets mathématiques intégrables.

Définition 13 (k -chaînes et k -formes) Les k -chaînes et les k -formes sont les éléments de $T_k^0(V)$ et $T_0^k(V)$ (respectivement).

Dans ce qui suit, on définira plusieurs concepts sur les chaînes, mais ils seront implicitement définis sur les formes par dualité. Pour une k -chaîne $\alpha \in \wedge_k$ nous appelons $k = \text{deg}(\alpha)$ le degrés de la forme. En se basant sur les k -chaînes, nous allons maintenant définir un objet plus géométrique. On veut généraliser l'idée qu'un vecteur tangent à une courbe représente un élément orienté infinitésimal de cette courbe. En deux dimensions par exemple, on veut construire un élément de surface orienté, par exemple à partir de deux vecteurs tangents. Si l'on change l'ordre des deux vecteurs, on inverse l'orientation de l'élément de surface. Les k -chaînes alternées vont généraliser ce concept à une dimension arbitraire, en se basant sur la notion de signature d'une permutation.

Définition 14 (Permutation) Une permutation σ de $\mathbb{N}_k = \{1, \dots, k\}$ est une application bijective de \mathbb{N}_k vers \mathbb{N}_k

Définition 15 (Signature) La signature ε d'une permutation σ de \mathbb{N}_k est définie par :

$$\varepsilon(\sigma) = \prod_{l < l'} \frac{\sigma(l) - \sigma(l')}{l - l'} \quad (3.2)$$

Si l'on décompose σ en transpositions élémentaire (échange de 2 éléments), la signature est +1 (resp. -1) si le nombre de transpositions est pair (resp. impair). Cette parité ne dépend pas de la façon dont on décompose.

La signature d'une permutation permet de définir la notion d'antisymétrie.

Définition 16 (k -chaînes alternées) Une k -chaîne $\alpha = \sum \alpha_{i_1, \dots, i_k} \mathbf{e}_{i_1, \dots, i_k}$ est dite alternée si et seulement si :

$$\alpha_{\sigma(i_1), \dots, \sigma(i_k)} = \varepsilon(\sigma) \alpha_{i_1, \dots, i_k}$$

pour n'importe quelle permutation σ de \mathbb{N}_k .

Les k -chaines représentent des éléments de volume orientés de dimension k . Une 0-chaine est un scalaire, une 1-chaine est un vecteur, une 2-chaine est un élément de surfaces orienté, une 3-chaine une élément de volume orienté. Pour bien comprendre cette construction, on va construire des k -formes alternées à partir de formes de degrés inférieur, par une opération appelée "produit extérieur".

3.1.3 Produit extérieur

Le produit tensoriel de deux formes alternées n'est pas nécessairement alterné, donc il nous manque un opérateur de base pour combiner les k -formes et k -chaines alternées. Une k -chaine peu être rendue alternée (ou antisymétrique) de la façon suivante :

$$Alt_k(\alpha) = \frac{1}{k!} \sum_{\sigma} \varepsilon(\sigma) \sum \alpha_{\sigma(i_1), \dots, \sigma(i_k)} \mathbf{e}_{i_1, \dots, i_k}$$

Cela va nous permettre de construire le *produit extérieur* qui combine deux chaines alternées en une chaine alternée :

$$\alpha \wedge \beta = Alt_{k+l}(\alpha \otimes \beta)$$

Le produit extérieur \wedge est :

- Anticommutatif : $\alpha \wedge \beta = (-1)^{deg(\alpha)deg(\beta)} \beta \wedge \alpha$
- Associatif : $(\alpha \wedge \beta) \wedge u = \alpha \wedge (\beta \wedge u)$
- Bilinéaire :

$$(c_1 \alpha_1 + c_2 \alpha_2) \wedge \beta = c_1(\alpha_1 \wedge \beta) + c_2(\alpha_2 \wedge \beta)$$

$$\alpha \wedge (c_1 \beta_1 + c_2 \beta_2) = c_1(\alpha \wedge \beta_1) + c_2(\alpha \wedge \beta_2)$$

Ces propriétés sont suffisantes pour le définir sur les coordonnées. De par sa définition, on a $deg(\alpha \wedge \beta) = deg(\alpha) + deg(\beta)$ donc le produit extérieur permet de construire des chaines alternées à partir de chaines de degrés inférieur. Par récurrence, toutes les chaines peuvent être construites à partir de chaines d'ordre 1 (vecteurs).

A cause de cela, on appelle \wedge_k et \wedge^k les espaces des k -chaines et k -formes alternées. \wedge_k est un espace vectoriel de dimension $\binom{d}{k}$, et de base :

$$\{\mathbf{e}_I = \mathbf{e}_{i_1} \wedge \dots \wedge \mathbf{e}_{i_k} | 0 \leq i_1 < \dots < i_k \leq d\}$$

où $I = \{i_1, \dots, i_k\}$ est l'ensemble des indices des vecteurs d'une base de \mathbb{R}^d . L'antisymétrie des chaines alternées est assurée par l'anticommutativité du produit extérieur. Ainsi, si une chaine $\alpha = \alpha_1 \wedge \alpha_2$, l'inversion de α_1 et α_2 inverse l'orientation de α . Cette condition est suffisante pour que la notion de chaine corresponde intuitivement à un élément de volume orienté. Géométriquement, $\alpha_1 \wedge \alpha_2$ est la chaine obtenue en extrudant α_1 le long de α_2 . Ainsi, si α_1 et α_2 sont proportionnels (colinéaires pour des vecteurs) α sera nulle, ce qui est attendu pour un objet antisymétrique.

Maintenant que nous avons défini les chaines ponctuellement, nous allons pouvoir les définir sur tout un ensemble de \mathbb{R}^d ou sur un variété, et nous intéresser à leurs propriétés différentielles.

3.1.4 k -chaines et k -formes différentielles

L'utilité majeure des k -formes et des k -chaines est leur application au calcul différentiel.

Définition 17 (k -chaines et k -formes différentielles) Les k -chaines et k -formes différentielles sur Ω (Ω est une variété ou un sous ensemble de \mathbb{R}^d) sont définies comme les applications lisses de Ω vers \wedge_k et \wedge^k

On appelle $\wedge_k(\Omega)$ et $\wedge^k(\Omega)$ les espaces des chaines différentielles sur Ω . Le calcul extérieur se base sur la correspondance suivante entre les concepts d’algèbre linéaire et de calcul différentiel :

$$\mathbf{e}^i \propto dx_i \quad \mathbf{e}_i \propto \nabla x_i$$

La différentielle des fonctions $df : \mathbb{R}^d \rightarrow \mathbb{R}$ est une 1-forme qui prend un vecteur \mathbf{v} en argument et renvoie la dérivée de f le long de \mathbf{v} . De cette façon, la relation fondamentale (3.1) est préservée :

$$\mathbf{e}^i(\mathbf{e}_j) = dx_i(\mathbf{e}_j) = \delta_{i,j}$$

Nous allons maintenant présenter la généralisation de cette différentielle aux k -formes. Celle-ci s’appelle la dérivée extérieure, et nous allons la présenter dans \mathbb{R}^d , puis l’étendre aux variétés.

3.1.5 Dérivée extérieure

Définition dans \mathbb{R}^d

La dérivée extérieure d’une k -forme a une définition abstraite par un ensemble de propriétés qui la rendent unique :

1. $deg(d\alpha) = deg(\alpha) + 1$
2. si $deg(\alpha) = 0$ alors d est la différentielle des fonctions
3. $d(\alpha \wedge \beta) = d\alpha \wedge \beta + (-1)^{deg(\alpha)} \alpha \wedge d\beta$
4. $d \circ d = 0$

La définition sur les coordonnées la rend plus compréhensible pour le lecteur : soit $\alpha = \sum_I \alpha_I dv_I$, alors sa dérivée extérieure $d\alpha$ est la $k + 1$ -forme :

$$d\alpha = \sum_I \frac{\partial \alpha_I}{\partial v_I} dv_I \wedge dv_I \quad (3.3)$$

A partir de cette définition abstraite, on peut vérifier que toutes les propriétés usuelles de la dérivée sont vérifiées :

- $d(\alpha_k + \beta_k) = d\alpha_k + d\beta_k$
- $d(\alpha_k \cdot \beta_k) = d\alpha_k \beta_k + \alpha_k d\beta_k$
- $d(\alpha_k \circ \beta_k) = (d\alpha_k \circ \beta_k) \cdot d\beta_k$

Nous allons maintenant étendre cette définition de la dérivée extérieure aux variétés.

Dérivée extérieure sur les variétés

La définition de d est d’une certaine façon récursive. Pour l’étendre aux variétés, il nous suffit donc d’étendre la différentielle des fonctions aux variétés. Dans ce cas, df doit être un élément du fibré cotangent $T^*\mathcal{S}$, i.e. l’ensemble des applications linéaires des espaces tangents $T\mathcal{S}$ (fibré tangent) vers \mathbb{R} .

Définition 18 (Différentielle sur les variétés) Il y a deux façons de définir df en un point $P \in \mathcal{S}$:

1. Soit $\mathbf{x} : U_2 \subset \mathbb{R}^2 \rightarrow U_3$ une carte non dégénérée d'un sous-ensemble $U_3 \subset \mathcal{S}$ et $O = \mathbf{x}^{-1}(P)$. df_P est l'élément de $T_P^* \mathcal{S}$ qui vérifie la loi de composition :

$$d(f \circ \mathbf{x})_O = df_P d\mathbf{x}_O$$

2. Soit $c : I \subset \mathbb{R} \rightarrow \mathcal{S}$ une courbe sur \mathcal{S} telle que $c(0) = P$, $c'(0) = \mathbf{v}$. Alors :

$$df_P(\mathbf{v}) = \frac{\partial(f \circ c)}{\partial t} \Big|_{t=0} \quad (3.4)$$

Lemme 3.1.1 (Expression) La première définition est équivalente à :

$$df_P = d(f \circ \mathbf{x}) G_x^{-1} d\mathbf{x}^T \quad (3.5)$$

où $G_x = d\mathbf{x}^T d\mathbf{x}$ est la première forme fondamentale.

PREUVE. Soit (O, u_1, u_2) une base canonique de U_2 , et $\mathbf{t}_i = \partial \mathbf{x} / \partial u_i$. $d\mathbf{x} = (\mathbf{t}_1, \mathbf{t}_2)$ est une base de $T\mathcal{S}$ (c'est aussi la matrice Jacobienne), $d\mathbf{x}^T$ est une base de $T^*\mathcal{S}$ et df peut s'écrire : $df = (\phi_1, \phi_2) d\mathbf{x}^T = \phi d\mathbf{x}^T$. En insérant ceci dans la loi de composition on obtient :

$$d(f \circ \mathbf{x}) = \phi d\mathbf{x}^T d\mathbf{x} = \phi G_x$$

\mathbf{x} n'est pas dégénéré donc G_x est inversible et :

$$\phi = d(f \circ \mathbf{x}) G_x^{-1} \quad df = \phi d\mathbf{x}^T = d(f \circ \mathbf{x}) G_x^{-1} d\mathbf{x}^T$$

□

Lemme 3.1.2 (Géométriquement intrinsèque) La définition 1 est géométriquement intrinsèque (elle est indépendante du choix de la carte $\mathbf{x}(u_1, u_2)$).

PREUVE. Soit $\mathbf{y}(v_1, v_2)$ une autre carte de U_3 , et s l'application bijective $\mathbf{x}^{-1} \circ \mathbf{y}$. Alors on a $\mathbf{y} = \mathbf{x} \circ s$, et donc $d\mathbf{y} = d\mathbf{x} ds$, et :

$$\begin{aligned} d(f \circ \mathbf{y}) G_y^{-1} d\mathbf{y}^T &= d((f \circ \mathbf{x}) \circ s) (ds^T d\mathbf{x}^T d\mathbf{x} ds)^{-1} ds^T d\mathbf{x}^T \\ &= d(f \circ \mathbf{x}) ds ds^{-1} G_x^{-1} ds^T ds^T d\mathbf{x}^T = d(f \circ \mathbf{x}) G_x^{-1} d\mathbf{x}^T \end{aligned}$$

ce qui prouve l'indépendance pour 1, et donc pour 2 puisqu'elles sont équivalentes. □

Théorème 3.1.3 (Equivalence) Les deux définitions sont équivalentes

PREUVE. (2 \Rightarrow 1) : Comme $(\mathbf{t}_1, \mathbf{t}_2)$ est une base, tout vecteur $\mathbf{v} \in T\mathcal{S}$ peut être écrit $\mathbf{v} = v_1 \mathbf{t}_1 + v_2 \mathbf{t}_2$. $\mathbf{v} = d\mathbf{x}(v_1, v_2)^T$, donc on a $G_x^{-1} d\mathbf{x}^T(\mathbf{v}) = (d\mathbf{x}^T d\mathbf{x})^{-1} d\mathbf{x}^T d\mathbf{x}(v_1, v_2)^T = (v_1, v_2)^T$. Soit $c(t) = \mathbf{x}(tv_1, tv_2)$. On a $c(0) = O$ et $c'(0) = \mathbf{v}$. Alors par (3.4) on a $\frac{\partial(f \circ c)}{\partial t} \Big|_{t=0} = d(f \circ \mathbf{x})(v_1, v_2)^T$ ce qui prouve l'implication.

(1 \Rightarrow 2) : Soit \mathbf{x}_c une carte de U_3 telle que $\mathbf{x}_c(v_1 t, v_2 t) = c(t)$. Alors :

$$\frac{\partial(f \circ c)}{\partial t} \Big|_{t=0} = \frac{\partial(f \circ \mathbf{x}_c(v_1 t, v_2 t))}{\partial t} \Big|_{t=0} = d(f \circ \mathbf{x})_O(v_1, v_2)^T = d(f \circ \mathbf{x})_O G_x^{-1} d\mathbf{x}^T \mathbf{v}$$

□

Donc les définitions 1. et 2. peuvent être utilisées indépendamment, de même que l'équation (3.5), et ces définitions sont géométriquement intrinsèques. La différentielle des fonctions étant définie, elle permet de définir une dérivée extérieure sur les variétés de la même façon (récursive) que précédemment.

La notion de dérivation étant maintenant bien définie pour les k -formes, y compris sur les variétés, nous allons maintenant définir l'intégration des formes, ce qui nous plongera dans la dualité chaînes/formes.

3.1.6 Intégration et dualité

Intégration dans \mathbb{R}^d

On dit souvent que les k -formes sont faites pour être intégrées sur des ensembles de dimension k . Ainsi, la notation usuelle :

$$\int_0^1 f(x)dx = \int_{[0,1]} f(x)dx$$

fait référence à l'intégrale d'une 1-forme $f(x)dx$ sur l'ensemble (de dimension 1) $[0, 1]$. Pour comprendre la signification de ceci, nous devons retourner à la définition originale de l'intégrale. Les définitions usuelles de l'intégrale sont basées sur la notion de "mesure", et nécessitent de définir une suite de subdivisions de l'ensemble sur lequel on intègre dont la "mesure" des cellules tend vers 0. Ainsi l'intégrale d'une fonction $f : \Omega \rightarrow \mathbb{R}$ sur un ensemble $U \subset \Omega$ nécessite de subdiviser U en n cellules U_k^n mesurables, et qui vérifient :

$$\lim_{n \rightarrow \infty} \sup_k |U_k^n| = 0$$

où $|U_k^n|$ est la mesure de U_k^n . On a alors la définition suivante :

$$\int_U f dU = \lim_{n \rightarrow \infty} \sum_{k=1}^n f_k^n |U_k^n|$$

où l'on voit que le dU dans l'intégrale correspond au $|U_k^n|$ dans la somme. Dans la définition de Riemann, U_k^n correspond à une subdivision régulière de U en n cellules. Dans la définition de Lebesgue, $U_k^n = f^{-1}(I_k^n)$ où I_k^n est une subdivision de $f(U)$ en n cellules.

Pour expliquer l'intégration des k -chaines, il est plus facile de commencer par l'intégration des d -chaines dans \mathbb{R}^d . Dans \mathbb{R}^d , une d -chaîne s'écrit $\alpha_d = f_d V_d$ où f_d est une fonction scalaire (à valeur dans \mathbb{R}), et $V_d = \mathbf{e}_1 \wedge \dots \wedge \mathbf{e}_d$ peut être vu comme un (hyper)cube infinitésimal. Une d -forme s'écrit $\alpha_d = f^d dV_d$ où f^d est une fonction scalaire et $dV_d = dx_1 \wedge \dots \wedge dx_d$ peut être vu comme une mesure d'un volume infinitésimal V_d . Ainsi, on intègre une d -forme sur une d -chaîne en faisant une somme (infinie) de ces mesures infinitésimales. Cette intégrale généralisée est souvent notée $\langle \alpha^d | \alpha_d \rangle$, et on a :

$$\langle \alpha^d | \alpha_d \rangle = \int_{\mathbb{R}^d} f_d f^d dV_d$$

au sens usuel de l'intégrale dans \mathbb{R}^d . On voit qu'une d -chaîne généralise ainsi la notion d'ensemble sur lequel on peut intégrer, au sens où un ensemble $\Omega \subset \mathbb{R}^d$ peut être vu comme un cas particulier de chaîne pour laquelle $f_d = 1_\Omega$ où $1_\Omega(\mathbf{x}) = 1$ si $\mathbf{x} \in \Omega$ et 0 sinon est la fonction caractéristique de Ω . Ainsi on a :

$$\langle \alpha^d | 1_\Omega V_d \rangle = \int_\Omega \alpha^d$$

Cette façon généralisée de concevoir l'intégrale en calcul extérieur rend complètement duales les d -formes qu'on intègre, et les d -chaines sur lesquelles elles sont intégrées. Voyons maintenant comment cette intégrale se généralise aux formes d'ordre arbitraire.

Intégration sur une variété

On a expliqué comment intégrer une d -forme sur une d -chaîne dans \mathbb{R}^d . Que se passe-t-il maintenant si on veut intégrer une k -forme sur une k -chaîne avec $k < d$? Une k -chaîne dans \mathbb{R}^d peut être vue comme la

donnée d'un élément infinitésimal orienté de dimension k , comme par exemple un élément infinitésimal d'une k -variété plongée dans \mathbb{R}^d . Cependant, comme la k -chaîne est définie dans \mathbb{R}^d , cette variété emplirait tout \mathbb{R}^d , ce qui paraît impossible vu qu'elle est de dimension $k < d$. En fait une k -chaîne dans \mathbb{R}^d ne définit pas une unique variété, mais une famille infinie de variétés appelée *foliation* [Mullen et al. 2006] et qui remplit \mathbb{R}^d . Ainsi, cette construction assimile une 1-chaîne (champs de vecteur) avec l'ensemble de ses lignes de champs (qui remplit bien l'espace), et une 2-chaîne (champs d'éléments de surface orientés) avec l'ensemble des surfaces tangentes à ces éléments de surface, tel l'ensemble des isosurfaces d'une fonction scalaire. L'intégrale d'une k -forme sur une telle foliation est alors obtenue comme avant en somme infinie des mesures infinitésimales locales $\alpha^k(\alpha_k)$. L'intégration sur une variété est alors définie au sens des distributions :

$$\langle \alpha^k | \alpha_k^{\mathcal{S}} \rangle = \int_{\mathcal{S}} \alpha^k$$

où l'intégrale sur \mathcal{S} peut être définie sur une paramétrisation de \mathcal{S} dans \mathbb{R}^k en tenant compte de la métrique. La k -variété \mathcal{S} peut alors être vue comme une k -chaîne $\alpha_k^{\mathcal{S}}$ au sens des distributions (elle est "infinie" sur \mathcal{S}_k et 0 ailleurs). Pour définir l'intégrale sans paramétrisation, on a besoin d'une subdivision de \mathcal{S} . Alors qu'une subdivision de \mathbb{R}^d est triviale à expliciter, cela ne l'est plus quand on est sur une k -variété. Sur un maillage triangulé en revanche, on a une subdivision naturelle donnée par les triangles. Cette approche, qui conduit au calcul extérieur discret, sera présentée dans la Section 3.2.

Dualité et opérateur de bord

De par cette généralisation de l'intégrale, on a maintenant une dualité $\langle \cdot | \cdot \rangle$ entre les k -formes et les k chaînes. En se basant sur cette dualité, on peut définir l'opérateur de bord sur les chaînes comme l'adjoint de d par $\langle \cdot | \cdot \rangle$:

$$\langle \alpha_{k+1} | d_k \alpha^k \rangle = \langle \partial_{k+1} \alpha_{k+1} | \alpha^k \rangle$$

On peut vérifier sur des cas simples que cet opérateur coïncide avec notre intuition de ce qu'est un "bord". Pour un ensemble $\Omega \subset \mathbb{R}^d$, cela veut dire que la dérivée extérieure vérifie :

$$\int_{\Omega} d_k \alpha^k = \langle 1_{\Omega} V_d | d_{d-1} \alpha^{d-1} \rangle = \langle \partial_d 1_{\Omega} V_d | \alpha^k \rangle = \int_{\partial \Omega} \alpha^k \quad (3.6)$$

qu'on appelle souvent le théorème de Stokes et qui est une propriété fondamentale du calcul extérieur et résume à elle seule :

– Le Théorème fondamental de l'intégration :

$$\int_a^b f'(x) dx = f(b) - f(a)$$

– La formule de Green :

$$\int_D \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} dx dy = \int_{\partial D} P dx + Q dy$$

– Le théorème d'Ampère :

$$\oint_{\partial S} \vec{v} \cdot d\vec{l} = \iint_S \text{rot} \vec{v} \cdot d\vec{S}$$

– La formule d'Ostrogradsky :

$$\oint_{\partial U} \mathbf{v} d\vec{S} = \int_U (\text{div} \mathbf{v}) dx dy dz$$

Enfin, cette dualité naturelle entre formes et chaînes est explicitée par un isomorphisme noté \sharp qui associe les k -chaînes aux k -formes (il fait monter les indices en exposants) et un isomorphisme appelé \flat qui associe les k -formes aux k -chaînes (il fait descendre les exposants en indices). Dans \mathbb{R}^d , ces isomorphismes changent simplement les $\mathbf{e}^i = dx_i$ en \mathbf{e}_i et réciproquement, ce qui est suffisant pour les définir. Sur une variété, les opérateurs \sharp et \flat doivent en plus prendre la métrique en compte, ce qui ne sera pas développé ici.

En plus de la dualité entre formes et chaînes, le calcul extérieur propose une autre dualité utile basée sur l'étoile de Hodge.

3.1.7 Étoile de Hodge

L'étoile de Hodge est un autre opérateur de dualité qui se base sur le fait que les espaces \wedge^k et \wedge^{d-k} sont des espaces vectoriels de même dimension $\binom{d}{k} = \binom{d}{d-k}$, et sont donc isomorphes. L'étoile de Hodge construit donc un isomorphisme entre ces espaces de la façon suivante :

$$\alpha(\mathbf{e}_1 \wedge \dots \wedge \mathbf{e}_d) = (\star_k \alpha)(\mathbf{e}_{k+1} \wedge \dots \wedge \mathbf{e}_d)$$

La définition sur les chaînes est équivalente. L'étoile de Hodge associe à une k -forme la k -forme "orthogonale" de dimension maximale. Par exemple, sur les 2-chaînes en dimension 3, l'étoile de Hodge associe sa normale à un élément de surface orientée. En dimension 2, elle associe les 0-formes aux 2-formes, c'est à dire les fonctions scalaires (fonctions définies de façon ponctuelle) aux fonctions définies de façon intégrale. L'index k est souvent omis car il est implicite en fonction de l'ordre de la forme à laquelle s'applique l'étoile de Hodge. Nous le garderons pour enlever l'ambiguïté, et pour rappeler au lecteur que l'étoile de Hodge est un opérateur différent quand il agit sur des formes d'ordre différents.

3.1.8 Produit interne

L'étoile de Hodge va finalement nous permettre de doter les espaces $\wedge^k(\Omega)$ d'un produit interne, et donc d'une norme et d'une structure d'espace de Hilbert. Ce produit interne se définit de la façon suivante :

$$\langle \alpha, \beta \rangle_{\wedge^k(\Omega)} = \int_{\Omega} \alpha \wedge \star_k \beta$$

On peut vérifier que $\alpha \wedge \star_k \beta$ est une forme d'ordre maximale sur Ω et qu'elle peut donc bien être intégrée sur Ω .

Ce produit interne permet de définir une norme sur $\wedge^k(\Omega)$:

$$\|\alpha\|_{\wedge^k(\Omega)}^2 = \langle \alpha, \alpha \rangle_{\wedge^k(\Omega)}$$

Pour cette nouvelle dualité, l'adjoint de d s'appelle maintenant la *codifférentielle* δ . En particulier, $\delta_k : \wedge^k(\Omega) \rightarrow \wedge^{k-1}(\Omega)$ est l'adjoint de d_{k-1} :

$$\langle d_{k-1} \alpha, \beta \rangle = \langle \alpha, \delta_k \beta \rangle$$

ce qui donne $\delta_k = (-1)^{nk+n+1} \star_{n-k+1} d_{n-k} \star_k$. Sur les surfaces ($n = 2$), les différentielles s'écrivent $\delta_k = -\star_{3-k} d_{2-k} \star_k$, et ne sont définies que pour $k = 1$ et 2 .

3.1.9 Conclusion

On a défini dans cette section les notions fondamentales de dérivation et d'intégration des formes, et expliqué les deux dualités présentes dans le calcul extérieur. La définition de l'intégration a fait apparaître le besoin d'une subdivision de l'espace sur lequel on intègre. Sur une surface triangulée, une telle subdivision est naturellement donnée par la structure de complexe simplicial (cf Chapitre 1). Cette approche conduit à la définition d'un calcul extérieur discret, que nous allons maintenant présenter.

3.2 Le calcul extérieur discret

Pour une introduction plus complète au calcul extérieur discret, on réfèrera le lecteur à [Desbrun et al. 2005a], [Hirani 2003] et à [Arnold et al. 2006] pour les preuves de convergence. On introduit ici brièvement les notions et notations de calcul extérieur discret qu'on utilisera abondamment dans le reste de cette thèse. On discutera aussi les dualités de calcul extérieur discret et ses relations particulières avec la théorie continue.

3.2.1 Formes, chaînes et intégration discrètes

Le calcul extérieur s'adapte de façon très naturelle au cadre discret par la définition des chaînes et formes discrètes basée sur la notion de complexe simplicial.

Comme on l'a expliqué au Chapitre 1, un k -simplexe s_k est l'étendue géométrique de $k + 1$ points, de sorte que les 0, 1 et 2-simplices sont des points, des arêtes et des triangles respectivement. Un complexe simplicial M de dimension n est alors formé des $n + 1$ collections de n_k k -simplices ($k = 0, \dots, n$), sous les conditions rappelées au Chapitre 1.

Définition 19 (k -chaîne discrète réelle) Une k -chaîne discrète réelle α_k sur M est une somme formelle de k -simplices orientés avec des coefficients dans \mathbb{R} . L'ensemble des k -formes réelle discrètes sur M est appelé $\wedge_k(M)$.

Ainsi, les k -chaînes sont données numériquement par une simple valeur réelle $\alpha_k(s_k)$ associée à chaque k -simplex orienté. Les k -chaînes discrètes sont alors définies par dualité :

Définition 20 (k -forme discrète réelle) Une k -forme discrète réelle α^k sur M est une application linéaire de $\wedge_k(M) \rightarrow \mathbb{R}$. L'ensemble des k -formes réelle discrètes sur M est appelé $\wedge^k(M)$.

Une k -forme discrète α^k est donc aussi donnée par une valeur réelle $\alpha^k(s_k)$ associée à chaque k -simplex orienté (cf Figure 3.1). On a donc entre les chaînes et les formes la même dualité qu'entre les vecteurs covariants (colonnes) et contravariants (lignes). De plus, les espaces $\wedge_k(M)$ et $\wedge^k(M)$ des k -chaînes et k -formes sur M sont des espaces vectoriels de dimension n_k . Ainsi, on verra généralement les k -formes et les k -chaînes comme des vecteurs de taille n_k , et les opérateurs linéaires de $\wedge^k(M)$ vers $\wedge^l(M)$ comme des matrices de taille (n_k, n_l) .

Cette dualité entre chaîne et formes est évidemment la même que la dualité $\langle \cdot | \cdot \rangle$ qui correspond à l'intégration des formes sur les chaînes en continu. Ainsi l'application d'une k -forme à une k -chaîne :

$$\langle \alpha^k | \alpha_k \rangle = \alpha^k(\alpha_k) = \sum_{s_k} \alpha^k(s_k) \alpha_k(s_k)$$

correspond à l'intégrale en calcul extérieur discret.

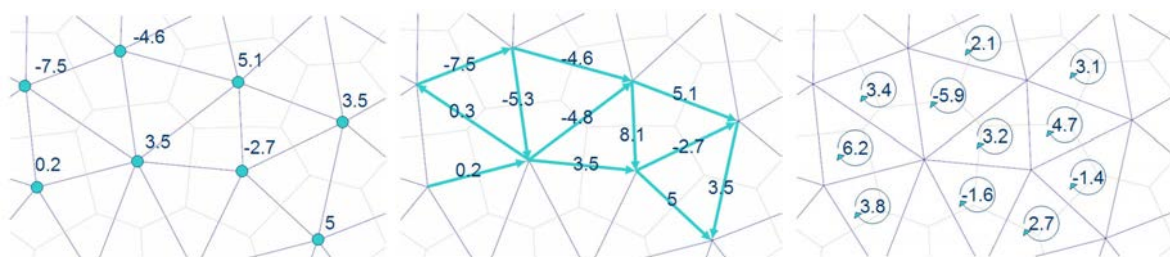


FIG. 3.1 – De gauche à droite : des 0,1 et 2-formes sur un maillage

Remarque 3.2.1 *Il est d'une certaine façon plus facile de définir un calcul extérieur dans le cadre discret qu'en continu, alors qu'en général une discrétisation d'une théorie continue nécessite plus d'arsenal mathématique (ex : éléments finis). Ce qui rend le calcul extérieur discret plus facile est que la dualité entre les k -chaines et les k -formes est plus évidente. Dans le cadre continu, on a d'ailleurs été obligé d'introduire une partition de l'espace en éléments infinitésimaux qui correspond exactement au complexe simplicial du cadre discret, qui a le mérite d'explicitement cette partition. De plus, assimiler les k -chaines à des k -variétés nécessite une gymnastique faisant intervenir les distributions, alors qu'il s'agit d'une simple somme de simplexes pondérés dans le cadre discret. Cette pondération peut d'ailleurs être vue comme une "rasterisation" d'une k -variété continue sur le simplexe. De même, une k -forme discrète doit être vue comme l'intégrale d'une k -forme continue sur un k -simplexe.*

Pour pouvoir définir les opérateurs les plus importants sur les k -chaines discrètes, et en particulier pour définir la dérivée extérieure discrète, il va nous falloir définir ce qu'est une *orientation*.

3.2.2 Orientation

Le calcul extérieur discret est fortement basé sur la notion d'orientation, et nécessite entre autre de choisir une orientation pour chaque simplexe. Étant donné un simplexe $\sigma^k = [v_1, \dots, v_k]$, son orientation est définie par l'ordre dans lequel ses sommets apparaissent. De façon plus formelle, soit σ une permutation de $\{1, \dots, k\}$, alors :

$$[v_{\sigma(1)}, \dots, v_{\sigma(k)}] = \varepsilon(\sigma)[v_1, \dots, v_k] \quad (3.7)$$

où $\varepsilon(\sigma)$ est la signature d'une permutation définie par l'équation (3.2). La signature de σ , vaut $+1$ (resp. -1) si la permutation est obtenue par un nombre pair (resp. impair) de transpositions (échange de 2 éléments). Cela permet de construire un opérateur de comparaison de l'orientation de deux simplexes basés sur les mêmes sommets :

$$\text{orient}([v_1, \dots, v_k], [v_{\sigma(1)}, \dots, v_{\sigma(k)}]) = \varepsilon(\sigma) \quad (3.8)$$

et on définit par convention $\text{orient}(s_k, s'_k) = 0$ si s_k et s'_k ne sont pas basés sur les mêmes sommets.

On peut aussi comparer les orientations d'un simplexe s_k et de l'une de ses faces maximale (face de dimension $k-1$). Si s_{k-1} est une face maximale de s_k , alors il existe un unique sommet $v_k \prec s_k$ qui n'appartient pas à s_{k-1} , et l'orientation relative est donnée par :

$$\text{orient}(s_{k-1}, s_k) = \text{orient}([s_{k-1}, v_k], s_k) \quad (3.9)$$

En pratique, l'orientation est définie de façon implicite par le choix d'un ordre sur les simplexes quand on assimile les k -formes/chaines avec des vecteurs de taille n_k . Dans ce cas, on peut choisir que tous

les k -simplices sont définis par leurs sommets (0-simplices) pris dans l'ordre croissant : $s_k = [v_1, \dots, v_k]$ avec $0 \leq i_1 < \dots < i_k \leq n_k$. Cela permet de construire les matrices d'adjacence signées A_k de dimension (n_k, n_{k+1}) , et de terme $(A_k)_i^j = \text{orient}(s_k[i], s_{k+1}[j])$. Par exemple, un maillage triangulé a deux matrices d'adjacence :

1. A_0 est la matrice d'adjacence signée arête/sommet, chaque colonne de A_0 correspond à une arête et est remplie de zéro sauf aux lignes correspondant au sommet de départ et d'arrivée de l'arête pour lesquels les coefficients sont -1 et $+1$ (respectivement).
2. A_1 est la matrice d'adjacence signée triangle/arête, chaque colonne de A_1 correspond à un triangle et est remplie de zéro sauf aux lignes correspondant aux trois arêtes pour lesquelles les coefficients sont ± 1 (le signe dépend des orientations respectives du triangle et de l'arête).

3.2.3 Dérivée extérieure et opérateur de bord

Comme dans le cadre continu, le calcul extérieur discret dispose d'une dérivée extérieure sur les k -formes, et on peut construire un opérateur de bord ∂ sur les k -chaines. Ces deux opérateurs sont très faciles et intuitifs à définir (par leurs matrices) dans le cadre discret :

- L'opérateur de bord $\partial_k : \wedge_k(M) \rightarrow \wedge_{k-1}(M)$ est défini par la matrice d'adjacence signée A_k .
- La dérivée extérieure $d_k : \wedge^{k-1}(M) \rightarrow \wedge^k(M)$ est définie par la matrice A_k^t .

ce qui rend leur dualité évidente. L'opérateur de bord correspond exactement à ce que l'on attend intuitivement : le bord d'une arête orientée est composé de ses deux sommets (avec un coefficient ± 1 selon que l'arête part de ou arrive au sommet). Le bord d'un triangle orienté est composé de ses arêtes (avec un coefficient ± 1 selon que l'orientation des arêtes et du triangle concordent). Notons que la dérivée extérieure est plus facile à définir comme la duale de l'opérateur de bord, alors que c'était le contraire dans le cadre continu où cela nécessite d'utiliser la théorie des distributions, alors que dans le cadre discret, une k -chaîne est simplement une somme pondérée de k -simplexes. La dérivée extérieure ainsi définie discrétise à elle seule tous les opérateurs différentiels usuels : la dérivée extérieure des 0-formes (scalaires) est équivalente au gradient, sur les 1-formes elle équivaut au rotationnel. Pour définir la divergence, on aura en plus besoin de la dualité de Hodge qui sera définie dans la Section 3.2.5.

Remarque 3.2.2 (Nilpotence) *On peut vérifier qu'on a toujours $A_k A_{k-1} = 0$ et $A_{k-1}^t A_k^t = 0$ quel que soit k . Cela veut dire qu'un bord n'a pas de bord, et qu'un gradient n'a pas de rotationnel. Ce résultat va nous servir à définir l'homologie et la cohomologie.*

3.2.4 (Co)homologie

Les notions d'homologie et de cohomologie nous seront très utiles dans la troisième partie de cette thèse où elles nous serviront à étudier la topologie des champs de direction. Leurs définitions se basent sur le noyau et l'image de A_k :

- $\text{Ker}(A_k)$ est l'ensemble des k -chaines sans bords : $\omega_k \in \text{Ker}(A_k) \Leftrightarrow \partial_k \omega_k = 0$. On les appelle k -chaines closes.
- $\text{Im}(A_k)$ est l'ensemble des $k-1$ -chaines qui sont le bord d'une k -chaîne : $\omega_{k-1} \in \text{Im}(A_k) \Leftrightarrow \exists \omega_k \in \wedge^k$ telle que $\omega_{k-1} = \partial_k \omega_k$. On les appelle k -chaines exactes, et elles correspondent intuitivement à des bords.
- $\text{Ker}(A_k)^t$ est l'ensemble des k -formes dont la dérivée extérieure (gradient, rotationnel pour $k = 0$ ou 1) est nulle. On les appelle k -formes closes.

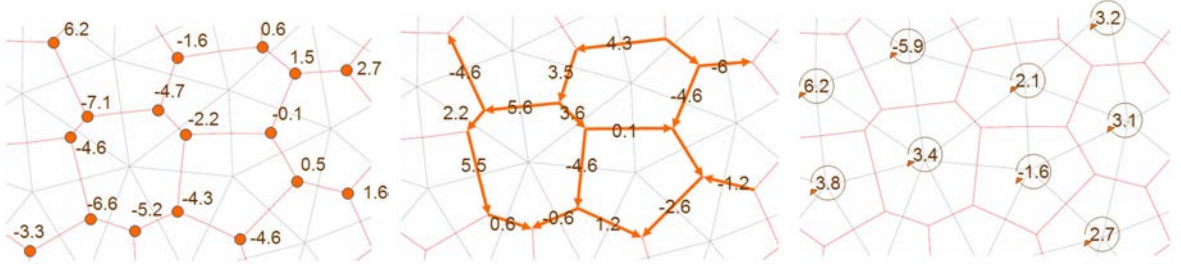


FIG. 3.2 – De gauche à droite : une 0,1 et 2-forme sur le maillage dual

– $Im(A_k)^t$ est l'ensemble des k -formes qui sont la dérivée extérieure (gradient, rotationnel) d'une $k - 1$ -forme. On les appelle k -formes *exactes*.

Comme $A_k A_{k+1} = 0$, on a $Im(A_{k+1}) \subset Ker(A_k)$. On peut donc définir le groupe quotient : $Ker(A_k)/Im(A_{k+1})$ qu'on appelle le k -ième groupe d'homologie du complexe simplicial M (A_k ne dépend que de la connexité de M). De façon duale, $Ker(A_k)^t/Im(A_{k+1})^t$ est appelé le k -ième groupe de cohomologie de M . Même si elles sont complètement duales, l'homologie est plus facile à comprendre de la cohomologie de façon intuitive. Un élément de $Ker(A_k)/Im(A_{k+1})$ est une classe de k -chaines *closes* qui ne diffèrent que par une k -chaîne *exacte*. Étant donné une k -chaîne *close* ω_k^c , alors pour toute k -chaîne *exacte* (bord) ω_k^e , $\omega_k^c + \omega_k^e$ est dans la même classe d'équivalence que ω_k^c . Dans ce cas, on dira que ω_k^c et $\omega_k^c + \omega_k^e$ sont homologues. Une autre façon de voir les classes d'équivalence de $Ker(A_k)/Im(A_{k+1})$ est de dire que deux k -chaines closes ω_{k1}^c et ω_{k2}^c sont homologues si et seulement si leur différence est exacte (est un bord) : $\exists \omega_{k+1} \in \wedge_{k+1}$ telle que $\omega_{k1}^c - \omega_{k2}^c = \partial_{k+1} \omega_{k+1}$.

Pour construire des k -chaines homologues à une k -chaîne ω_k donnée, il suffit de construire n'importe quelle $k + 1$ -chaîne ω_{k+1} dont le bord contient ω_k . Alors l'opposé du reste du bord $\omega_k - \partial \omega_{k+1}$ est homologue à ω_k . L'homologie est plus générale que l'homotopie (déformation continue) parce qu'elle permet de fusionner ou de diviser des chaînes. En particulier, on peut ajouter n'importe quelle chaîne exacte (bord) à une chaîne close sans en changer la classe d'homologie.

L'homologie est triviale dans \mathbb{R}^d où les chaînes closes sont nécessairement exactes, et il y a donc une seule classe d'homologie contenant toutes les formes closes. De plus toutes les formes closes ou exactes sont homologues à la chaîne vide \emptyset . Les classes d'homologie permettent de compter les "trous" et les "anses" du domaine sur lequel elles sont définies. En effet, une k -chaîne qui fait le tour d'un trou ou d'une anse n'est plus homologue à \emptyset puisqu'on ne peut plus définir son "intérieur".

Comme dans le cadre continu, on va maintenant définir la dualité de Hodge. Par contre, contrairement au cadre continu, cette dualité nécessite de définir au préalable un autre objet : le complexe dual.

3.2.5 Complexe dual

Pour continuer notre exposé des concepts du calcul extérieur discret, et en particulier pour définir une étoile de Hodge, nous devons maintenant définir un complexe dual. A chaque k -simplexe $s_k = [v_1 \dots v_k]$ du complexe simplicial (primal) M correspond une $n - k$ -face (et non simplexe) $\star s_k$ du complexe (non simplicial) dual dont le plongement est :

$$[\star s_k] = \{ \mathbf{x} \in [M] \mid \exists r \in \mathbb{R}^+ \text{ tel que } ||dist(\mathbf{x}, [v_1]) = \dots = dist(\mathbf{x}, [v_k]) = r \text{ et } \forall [v_i] dist(\mathbf{x}, [v_i]) \geq r \}$$

Sur un maillage, $[\star s_k]$ est une variété de dimension $2 - k$ mais pas un $2 - k$ -simplex. La face duale $\star s_k$ hérite son orientation de l'orientation du k -simplexe utilisé pour la définir.

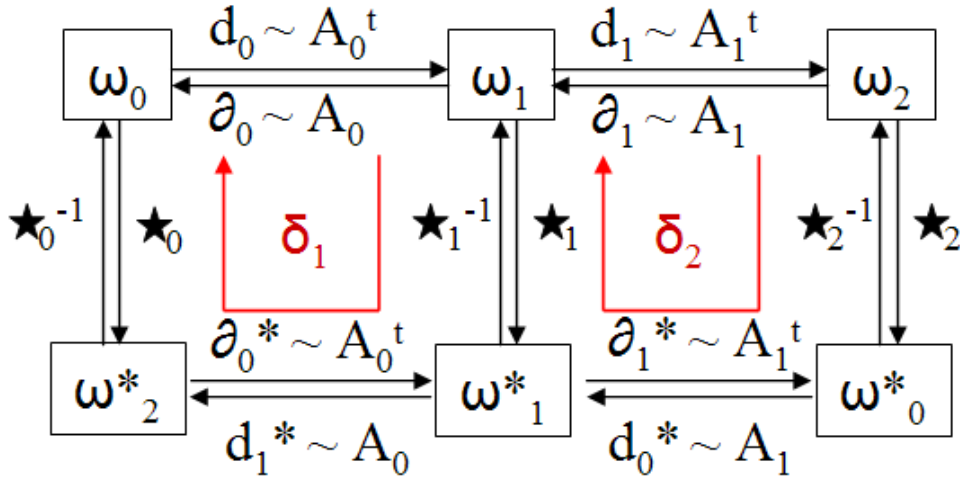


FIG. 3.3 – Le complexe de de Rham montre comment les opérateurs du calcul extérieur discret relient les k -formes et leurs duales entre elles.

On peut définir naturellement les k -chaines et k -formes duales sur le complexe dual (cf Figure 3.2) comme des vecteurs covariants et contravariants de dimension n_{n-k} correspondant à un scalaire sur chaque $n - k$ -face.

Le complexe dual a des matrices d’adjacence signées A_{n-k}^t . Son opérateur de bord ∂_k^* et sa dérivée extérieure d_k^* sont définies par les matrices A_{n-k}^t et A_{n-k} (cf Figure 3.3).

Ceci est une caractéristique très importante du calcul extérieur discret : dans le cadre continu, il n’y a pas de distinction entre les k -formes et leurs duales qui sont simplement des $n - k$ -formes (appliquer l’étoile de Hodge ne change que le degré de la forme). Au contraire, en calcul extérieur discret, les formes duales sont définies sur un complexe différent, qui n’est même pas simplicial, et ont donc une structure différente des formes primales. En particulier l’opérateur de bord et la dérivée extérieure sont définis différemment sur les chaines/formes et leurs duales. Cela rend le calcul extérieur plus compliqué en discret qu’en continu, mais a le mérite d’explicitement une subdivision sur laquelle on peut calculer l’intégrale.

Un dernier problème, et non des moindres, est que la définition que l’on a donnée ici des faces duales nécessite de calculer un diagramme de Voronoï des sommets (0-simplices) des sommets, ce qui sera en général beaucoup plus compliqué que de construire le dual circoncentrique (basé sur les centres des cercles circonscrits aux triangles) puisqu’un triangle ne contient pas son centre en général (dès qu’un de ses angles dépasse $\pi/2$). Les définitions usuelles du calcul extérieur discret passent outre ce problème, ce qui conduit à des poids qui peuvent être négatifs.

En se basant sur ce complexe dual, nous allons maintenant définir l’étoile de Hodge.

3.2.6 Étoile de Hodge

Contrairement aux concepts de calcul extérieur que l’on a défini jusqu’ici, l’étoile de Hodge est plus difficile à construire que son équivalent continu, puisqu’elle nécessite la construction explicite d’un complexe dual $\star M$. L’étoile de Hodge permet ainsi de passer des formes primales (définies sur le complexe primal M , cf Figure 3.1) aux formes duales (définies sur le complexe dual $\star M$, cf Figure 3.2). Au niveau des

simplices, l'étoile de Hodge transforme un k -simplexe s en son dual $\star s$ avec une pondération vérifiant :

$$\star_k(\|s\|s) = \|\star s\| \star s$$

où $\|\cdot\|$ est le volume du simplexe. Ceci permet naturellement de définir l'étoile de Hodge sur les k -chaines qui sont des sommes pondérées de k -simplices. En tant que matrice, l'étoile de Hodge est diagonale, d'éléments $\star_k^s = \|\star s\|/\|s\|$. Sur un maillage, nous avons donc trois étoiles de Hodge :

$$\star_0^i = \|\star i\| ; \star_1^{ij} = \frac{\|\star ij\|}{\|ij\|} ; \star_2^{ijk} = \|t\|^{-1}.$$

L'étoile de Hodge introduit ainsi une métrique dans le cadre du calcul extérieur discret qui n'était jusque ici basé que sur la connectivité de M . En effet, les termes $\|\star s_k\|/\|s_k\|$ de l'étoile de Hodge nécessitent de *mesurer* les simplexes et les faces duales, ce qui nécessite une métrique.

Comme on l'a dit précédemment, le complexe dual est le diagramme de Voronoï des sommets, qui peut devenir très difficile à calculer si les triangles ont des angles supérieurs à $\pi/2$. C'est pourquoi il est généralement approximé par le dual circoncentrique. Dans ce cas, et en utilisant la métrique induite par le plongement $[M]$ de M , on a les expressions suivantes pour les termes des matrices de Hodge :

$$\star_1^{ij} = \cot \beta_{k,ij} + \cot \beta_{l,ji} \quad \star_0^i = \sum_{v \in St_1(i)} \|iv\|^2 \star_1^{iv}.$$

Comme dans le cadre continu, l'étoile de Hodge permet de définir un produit interne (et donc une norme sur les chaines/formes).

3.2.7 Produit interne

En calcul extérieur discret, le produit interne sur $\wedge^k(M)$ se définit très simplement à partir de l'étoile de Hodge :

$$\langle \omega^k, \eta^k \rangle = (\omega^k)^T \star_k \eta^k$$

d'où l'on tire une norme :

$$\|\omega^k\|^2 = \langle \omega^k, \omega^k \rangle = (\omega^k)^T \star_k \omega^k. \quad (3.10)$$

Ce produit interne permet aussi de définir la *codifférentielle* discrète $\delta_k : \wedge^k(M) \rightarrow \wedge^{k-1}(M)$ comme l'adjoint de la dérivée extérieure d :

$$\langle d_{k-1} \omega^k, \eta^{k+1} \rangle = \langle \omega^k, \delta_k \eta^{k+1} \rangle$$

ce qui donne $\delta_k = -\star_{k-1}^{-1} d_{k-1}^T \star_k$ (cf Figure 3.3).

On va maintenant expliquer la cohérence de cette définition du produit interne discret avec le produit interne continu en se basant sur les applications de *de Rham* et *Whitney* qui permettent de passer du cadre continu au discret (et réciproquement) de façon harmonieuse.

3.2.8 Applications de De Rham et Whitney

Cette sous section présente les applications de De Rham et Whitney qui permettent d'alterner entre le cadre continu et discret. Elle est inspirée de la présentation très claire de [Zahariev 2007].

Les applications de De Rham et Whitney permettent de passer de $\wedge^k([M])$ à $\wedge^k(M)$ et inversement :

Définition 21 (Application de De Rham) *L'application de De Rham $R^M : \wedge^k([M]) \rightarrow \wedge^k(M)$ est donnée pour toute k -forme différentielle $\alpha^k \in \wedge^k([M])$ et pour tout k -simplex s^k par :*

$$R^M(\alpha^k)(s^k) = \int_{s^k} \alpha^k$$

Si $k = 0$, c'est simplement l'évaluation de α^0 au sommet σ^0 . L'application de De Rham formalise l'idée de discrétisation d'une forme continue.

Définition 22 (Application de Whitney) *L'application de Whitney W^M interpole une k -forme discrète α^k sur $[M]$. Pour un k -simplex $s^k = [v_0, \dots, v^k]$, on peut définir :*

$$W_{s^k}^M(\mathbf{x}) = k! \sum_{i=0}^k (-1)^i \lambda_i(\mathbf{x}) d\lambda_0(\mathbf{x}) \wedge \dots \wedge d\widehat{\lambda}_i(\mathbf{x}) \wedge \dots \wedge d\lambda_k(\mathbf{x})$$

où $\lambda_i(\mathbf{x})$ est la coordonnées barycentrique du point $\mathbf{x} \in [M]$ relative au point $[i]$. Ces fonctions d'interpolation ont leur support contenu dans $St_d(s^k)$. L'application de Whitney $W^M : \wedge^k(M) \rightarrow \wedge^k([M])$ est alors donnée pour n'importe quelle k -forme discrète $\omega^k \in \wedge^k(M)$ par :

$$W^M(\omega^k) = \sum_{s^k \in M} \omega^k(s^k) W_{s^k}^M$$

Les applications proposées ici entre le cadre discret et continu consistent en une simple intégration sur les simplexes, et en une interpolation basée sur les coordonnées barycentrique. Ces applications ont des propriétés très souhaitables :

- $W^M d_k = dW^M$: L'application de Whitney commute avec la dérivée extérieure.
- $R^M W^M = id$ sur $\wedge^k(K)$: discrétiser l'interpolation d'une forme discrète revient à la même forme.
- $\langle \omega_1^k, \omega_2^k \rangle = \langle W^M \omega_1^k, W^M \omega_2^k \rangle_{L^2}$: les produits intérieurs discrets et continus sont cohérents.

Si l'on appelle α_i^0 la 0-forme qui vaut 1 au sommet i et 0 aux autres, on obtient exactement $W^M(\alpha_i^0) = P_1^i$, c'est à dire que les α_i^0 sont les fonctions de la base P_1 . En faisant de même pour les formes discrètes d'ordre $k > 0$, l'application de Whitney généralise les bases de fonction P_1 sur les complexes simpliciaux aux k -formes.

3.2.9 Calcul extérieur et dualités

Nous allons conclure ici notre présentation du calcul extérieur et de son équivalent discret en discutant les deux dualités fondamentales du calcul extérieur qui nécessitent d'être bien différenciées dans le cadre continu et discret :

1. La dualité $\langle \cdot | \cdot \rangle$ entre formes et chaînes : correspond à la dualité entre ce qu'on intègre et ce sur quoi on intègre. Elle s'appréhende plus facilement dans le cadre discret où les k -formes et k -chaînes sont toutes les deux définies par un scalaire sur chaque k -simplexe, et l'intégrale somme les produits de ces scalaires. Dans le cadre continu, cette dualité est moins évidente car on a l'habitude d'intégrer une fonction sur un ensemble. Si l'ensemble est de dimension maximale, on peut l'associer à sa

fonction caractéristique pour comprendre cette dualité. Si il ne l'est pas, on est obligés de parler de foliation et de distributions ce qui rend les choses moins intuitives.

2. La dualité $\langle \cdot, \cdot \rangle$ de Hodge : cette dualité est plus facile à définir dans le cadre continu où elle correspond à un simple changement de base. Dans le cadre discret, elle nécessite de construire un complexe dual qui n'est plus simplicial, et entraîne donc une asymétrie entre formes primales et duales qui n'existe pas en continu. De plus, ce complexe dual est soit difficile à calculer (diagramme de Voronoï des sommets), soit mal défini (dual circoncentrique basé sur les cercles des centres circonscrits aux triangles, qui ne sont pas forcément dans le triangle, et peuvent générer des arêtes de longueur négative).

Ces différences viennent du fait que le calcul extérieur continu nécessite d'aller regarder la fonction à une très petite échelle où elle peut être considérée constante. Cette petite échelle va correspondre à une division de \mathbb{R}^d en cubes $\mathbf{e}^1 \wedge \dots \wedge \mathbf{e}^d$, et l'extension aux variétés ne pourra se faire que par l'utilisation d'une paramétrisation et de la métrique induite. De plus, les notions d'ensemble et de bord d'un ensemble ne seront définies que dans un cas limite, au sens des distributions, de la notion plus générale de chaîne. Cette généralisation est difficile mais nécessaire pour appréhender la dualité $\langle \cdot, \cdot \rangle$ en continu.

Dans le cadre discret, la problématique est presque inversée : la dualité $\langle \cdot, \cdot \rangle$ est évidente par construction, et facile à manipuler numériquement où elle correspond à de simples produits scalaires, ou produits vecteur ligne/vecteur colonne. En revanche, la dualité de Hodge va poser de nombreux problèmes à cause l'asymétrie entre maillage et maillage dual et entre formes et formes duales. Pour régler ce problème, il faudrait définir de nouveaux complexes plus symétriques, et en particulier dont le dual aurait la même structure. Ce besoin vient aussi du fait que la définition usuelle de l'étoile de Hodge conduit à des coefficients négatifs qui rendront les calculs instables, et auront des conséquences importantes sur les applications. De tels complexes symétriques sont faciles à définir dans \mathbb{R}^d en le coupant par tous les plans $x_i \in \mathbb{Z} \quad \forall i$. Par contre c'est impossible sur des variétés arbitraires, où une telle grille régulière présente nécessairement des singularités où la structure du primal et du dual diffèrent. Cependant, cela nous amène déjà à nous intéresser à la question du remaillage quadrangulé auquel nous nous intéresserons dans la troisième partie de cette thèse en particulier, et qui permet d'obtenir plus de symétrie entre le complexe primal et son dual.

Deuxième partie

Harmoniques variétés

Introduction

Cette partie présente une méthode efficace et justifiée pour calculer les fonctions propres de l'opérateur de Laplace-Beltrami que nous appellerons *harmoniques variétés*. Les harmoniques variétés forment une base de fonctions sur une surface triangulée qui a de nombreuses propriétés intéressantes. En particulier, elles permettent de définir une transformation de Fourier explicite sur le maillage. Il est bien connu que les vecteurs propres d'un Laplacien discret permet définir une telle base de fonctions. Toutefois, d'importants problèmes théoriques et pratiques ont empêché cette idée d'être utilisée directement. Du point de vue théorique, la symétrie et la discrétisation de l'opérateur doivent être pris en considération avec le plus grand soin. Ces problèmes seront abordés au Chapitre 4. Du point de vue pratique, le calcul de quelques vecteurs propres est actuellement hors de portée sur des surfaces triangulées de plus de quelques milliers de sommets, et le stockage de ces vecteurs propres est prohibitif pour ces surfaces. Cette question sera étudiée dans le Chapitre 5, qui présentera également certaines applications des harmoniques variétés.

La partie précédente a introduit de nombreuses notions pour comprendre et étudier les bases de fonctions sur les variétés. Dans les exemples de bases de fonctions que nous avons donnés, les fonctions de base sont localisées dans l'espace (petit support). La plupart du temps, ces fonctions de base sont interpolantes, dans le sens où les coefficients des fonctions correspondent à la valeur de la fonction sur des points de contrôle. Cela permet à la base d'être "presque" orthogonale, étant donné que seuls les produits internes avec des fonctions de base voisines ne sont pas nulles. En calcul extérieur discret, nous avons aussi vu que les fonctions constantes par morceaux sont exactement orthogonales, mais au prix de la continuité, ce qui peut être problématique lorsqu'elles sont utilisées pour approximer des fonctions lisses. Ainsi, les approches directes ne conduisent pas à une base de fonctions "parfaite". Cependant, ces bases sont intéressantes car elles génèrent et paramètrent des espaces de Hilbert de dimension finie. Dans ces espaces, nous pouvons construire des bases de fonction orthogonale à partir des fonctions interpolantes. Une approche directe serait de faire une orthonormalisation de Gram-Schmidt de la base (voir Annexe A). Cependant, les éléments de cette base seraient moins significatif, et dépendraient de l'ordre dans lequel les premiers éléments de la base sont pris.

Une approche beaucoup plus intéressante est de rechercher les fonctions propres d'un opérateur symétrique. En effet, ces fonctions propres seront orthogonales (avec des valeurs propres réelles positives). Si l'opérateur est pertinent, nous aurons une définition de notre base indépendante de l'ordre des fonctions de la base initiale et ayant une vraie signification. Cette partie de la thèse est basée sur le choix du Laplacien pour cet opérateur. Nous allons commencer par définir les Laplaciens dans le cadre continu et discret. Nous en expliquerons le sens et les raisons qui en font un opérateur pertinent à utiliser. En particulier, nous expliquerons comment ses fonctions propres permettent une généralisation de la transformée de Fourier sur les variétés. Dans notre approche, la transformée de Fourier sera considérée comme un simple changement de base.

Le travail présenté dans cette partie à été publié dans l'article "Spectral Geometry Processing with Manifold Harmonics" [Vallet and Levy 2008]. L'étude du Laplacien (Chapitre 4) a cependant été très étoffée pour étendre la construction à des bases de fonctions arbitraires, et en particulier aux bases de fonctions P_n et bilinéaire. De plus, de nombreux sujets relatifs au Laplacien ont été étudiés en détail (conditions aux bords, symétrie, localisation).

4

Le Laplacien

Sommaire

4.1 Le Laplacien continu	56
4.1.1 Laplacien dans \mathbb{R}^d	56
4.1.2 Laplacien sur une variété	57
4.1.3 Laplacien d'une k -forme	57
4.1.4 Conclusion	58
4.2 Etat de l'art	59
4.3 Laplaciens discrets et discrétisés sur un maillage	60
4.3.1 Forme générale	60
4.3.2 Formulation en éléments finis	61
4.3.3 Retrouver la symétrie	62
4.3.4 Laplacien d'interpolation	62
4.3.5 Conditions aux bords	66
4.3.6 Laplacien du calcul extérieur discret	66
4.3.7 Localisation du Laplacien P_1 par le graphe mixte	67
4.3.8 Limites d'un Laplacien discret	70
4.4 Conclusion	71

Introduction

Le Laplacien est l'un des opérateurs fonctionnels les plus importants, bien au delà de ses nombreuses applications en informatique graphique. Le Laplacien caractérise l'irrégularité d'une fonction de la façon la plus naturelle. Ainsi, la plupart des lois fondamentales de la physique impliquent le Laplacien : diffusion de la chaleur, propagation des ondes, équations des champs électromagnétiques et gravitationnels, ... qui expriment tous la distribution ou l'évolution d'une quantité scalaire (chaleur, potentiel électrique ou gravitationnel) en fonction de leur Laplacien, c'est à dire de leur irrégularité.

Pour utiliser le Laplacien sur un maillage en pratique, on va devoir en définir une version discrète ou le discrétiser. En informatique graphique, les applications d'un tel Laplacien discret sont multiples : édition, déformation, paramétrisation, segmentation et compression de maillages, mise en correspondance de formes, calcul de surfaces minimales, ... ce qui explique pourquoi il est abondamment étudié et utilisé.

Dans cette thèse, et en particulier au Chapitre 5, on s'intéressera aux fonctions propres du Laplacien qui forment une base de fonctions aux propriétés très utiles.

Plan du chapitre

Ce chapitre propose une étude détaillée du Laplacien et de ses diverses discrétisations ou équivalents discrets. On commencera par présenter dans la Section 4.1 les définitions du Laplacien dans le cadre continu. Ensuite, on exposera dans la Section 4.2 l'état de l'art dans la discrétisation de ce Laplacien, ainsi que certaines de ses applications. Enfin, on étudiera dans la Section 4.3) diverses constructions de Laplaciens discrets ainsi que leurs propriétés et limites.

Contributions

- Nous présentons l'état de l'art sur la discrétisation du Laplacien en informatique graphique, ainsi que ses applications.
- Nous proposons une étude complète du Laplacien dans le cadre discret et continu.
- Nous proposons une méthode générale pour discrétiser le Laplacien pour une base de fonctions donnée, et l'appliquons aux bases de fonctions définies dans le Chapitre 2.
- Nous discutons les définitions discrètes du Laplacien, et nous nous intéressons en particulier à la symétrie du Laplacien, aux conditions aux bords et à la localisation du Laplacien.

4.1 Le Laplacien continu

Le Laplacien joue un rôle majeur en physique et en mathématique bien au delà de ses applications courantes en informatique graphique. Il caractérise la non-linéarité et la non-régularité des fonctions. En particulier, pour lisser une fonction où générer une fonction lisse, on minimisera son Laplacien sous certaines contraintes.

Dans cette section, on va commencer par présenter le Laplacien dans un cadre simple (Laplacien sur une fonction scalaire dans \mathbb{R}^d), puis le généraliser aux variétés et aux k -formes.

4.1.1 Laplacien dans \mathbb{R}^d

Dans \mathbb{R}^d , le Laplacien est défini comme la divergence du gradient :

$$\Delta = \operatorname{div} \operatorname{grad} = \nabla^2 = \nabla \cdot \nabla = \sum_i \frac{\partial^2}{\partial x_i^2} = \sum_i \partial_i \partial^i$$

où $\partial_i = \partial / \partial x_i$ et $\partial^i = dx_i$.

Intuitivement, cela caractérise l'irrégularité d'une fonction. On note $\Delta f(\mathbf{x})$ le Laplacien d'une fonction f un point \mathbf{x} . $\Delta f(\mathbf{x})$ mesure la différence entre $f(\mathbf{x})$ et une moyenne de f dans un petit voisinage de \mathbf{x} . Ainsi, il est positif sur un minimum local et négatif sur un maximum local (on discutera plus tard le signe du Laplacien). Les fonctions pour lesquelles $\Delta f = 0$ sur un domaine $U \subset \mathbb{R}^d$ sont appelées fonctions harmoniques, et elles sont d'une grande importance puisqu'elles assurent une régularité maximale de la fonction, et en particulier que celle-ci n'aura pas de bosses ou de trous (maxima et minima).

4.1.2 Laplacien sur une variété

Sur une n -variété \mathcal{S} avec une métrique g , le Laplacien peut aussi être défini, et on l'appelle alors l'opérateur de Laplace-Beltrami :

$$\Delta = \text{div grad} = \frac{1}{\sqrt{|g|}} \partial_i \sqrt{|g|} \partial^i$$

où $|g| = \text{Det}(g)$. Le nouveau terme $\sqrt{|g|}$ peut être interprété comme une "échelle" locale puisque l'élément de volume dV_n sur \mathcal{M} est $dV_n = \sqrt{|g|} \wedge_i dx_i$. L'opérateur de Laplace-Beltrami est donc isotrope puisqu'il ne dépend que de l'échelle locale induite par la métrique.

4.1.3 Laplacien d'une k -forme

En calcul extérieur, le Laplacien peut être étendu aux k -formes par ce qu'on appelle l'opérateur de Laplace-de Rham, défini par :

$$\Delta_k = \delta_{k+1} d_k + d_{k-1} \delta_k$$

où d_k est la dérivée extérieure sur les k -formes et δ_k est la codifférentielle. Sur un maillage triangulé ($n = 2$), on a simplement $\delta_k = -\star_{3-k} d_{2-k} \star_k$ et par convention $\delta_0 = 0$ et $d_2 = 0$. d_0 est équivalent au gradient sur les scalaires et δ_1 à la divergence sur les 1-formes vues comme des champs de vecteurs. L'opérateur de Laplace-de Rham est donc équivalent (mais opposé) à celui de Laplace-Beltrami sur les scalaires.

Remarque 4.1.1 (Signe du Laplacien) *La définition de l'opérateur de Laplace-De Rham (sur les scalaires) le rend opposé à l'opérateur de Laplace-Beltrami, ce qui est une source de confusion fréquente. Cela vient de la définition de la codifférentielle comme adjoint de d : $\langle d\alpha, \beta \rangle = \langle \alpha, \delta\beta \rangle$ qui induit parfois (en fonction des dimensions) un signe -. Cette définition assure que l'opérateur de Laplace-De Rham est positif (ses valeurs propres sont positives) alors que l'opérateur de Laplace Beltrami est négatif. Ainsi, on a $\langle \Delta\alpha, \beta \rangle = \langle \nabla\alpha, \nabla\beta \rangle$ pour l'opérateur de Laplace-de Rham, alors que c'est l'opposé pour Laplace-Beltrami. Il serait donc plus naturel d'utiliser l'opérateur (positif) de Laplace-de Rham. Cependant, ceci est en général mal connu, en particulier en informatique graphique où le symbole Δ réfère toujours à l'opérateur (négatif) de Laplace-Beltrami. Même en calcul extérieur discret, Δ est souvent défini comme $\star d \star d + d \star d \star$ qui est négatif. C'est pourquoi nous conserverons cette convention dans cette thèse : le symbole Δ réfèrera **toujours** à l'opérateur (négatif) de Laplace-Beltrami. Pour assurer des valeurs propres positives, on cherchera donc les fonctions propres de l'opérateur positif $-\Delta$.*

Une propriété fondamentale du Laplacien est sa symétrie (pour le produit interne). Soient α et β des k -formes, et Ω une variété fermée de dimension n . Alors le produit interne vérifie :

$$\begin{aligned} \langle \alpha, \Delta\beta \rangle &= \int_{\Omega} \alpha \wedge \star \Delta\beta = \int_{\Omega} \alpha \wedge d_k \star_{k+1} d\beta + \star \alpha \wedge d \star d \star \beta \\ &= \int_{\Omega} d\alpha \wedge \star d\beta + d \star \alpha \wedge \star d \star \beta = \langle d\alpha, d\beta \rangle + \langle d \star \alpha, d \star \beta \rangle = \langle \Delta\alpha, \beta \rangle \end{aligned}$$

Voyons ce que cela implique pour les 0-formes (fonctions scalaires) et les 1-formes (champs de vecteurs).

Laplacien d'une 0-forme (fonction scalaire)

Soient f et g des fonctions scalaires, alors le laplacien vérifie :

$$\Delta f = \star d \star df = \nabla \cdot \nabla f = \text{div grad} (f)$$

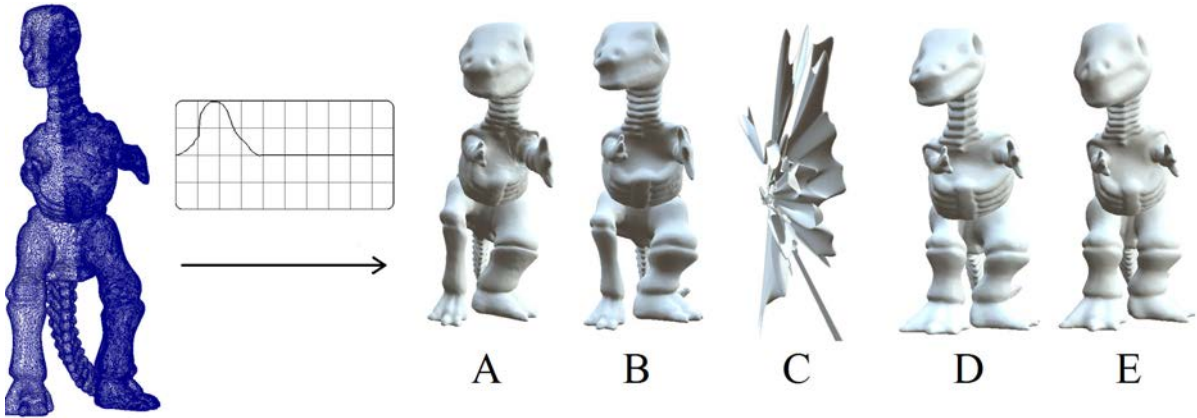


FIG. 4.1 – Filtrage d’une surface maillée de façon irrégulière (deux fois plus dense sur la moitié droite) en utilisant différents Laplaciens discrets. Le filtrage utilise la méthode proposée au Chapitre 5. Le Laplacien combinatoire (A), cotangentes non pondérées $\cot(\beta_{ij}) + \cot(\beta'_{ij})$ (B) produisent des résultats très asymétriques. Les cotangentes pondérées $(\cot(\beta_{ij}) + \cot(\beta'_{ij}))/A_i$ ne sont pas symétriques et ne permettent pas une reconstruction correcte (C). Les cotangentes symétrisées $(\cot(\beta_{ij}) + \cot(\beta'_{ij}))/(\sqrt{A_i A_j})$ (D) produisent un résultat meilleur mais toujours incorrect (la jambe droite est plus grosse). Seule une pondération symétrique $(\cot(\beta_{ij}) + \cot(\beta'_{ij}))/\sqrt{A_i A_j}$ est complètement indépendante du maillage (E).

qui correspond bien à sa définition usuelle dans \mathbb{R}^d , et on a :

$$\int_{\Omega} g \Delta f = \int_{\Omega} dg \wedge \star df = \int_{\Omega} \nabla g \cdot \nabla f = \int_{\Omega} \text{grad}(g) \cdot \text{grad}(f)$$

Laplacien d’une 1-forme (champs de vecteur)

Soient \mathbf{f} et \mathbf{g} des champs de vecteurs, alors le Laplacien vérifie :

$$\Delta \mathbf{f} = \nabla^\perp \nabla \wedge \mathbf{f} + \nabla \nabla \cdot \mathbf{f} = \text{cograd div } \mathbf{f} + \text{grad rot } \mathbf{f}$$

et on a :

$$\int_{\Omega} \mathbf{g} \Delta \mathbf{f} = \int_{\Omega} \nabla \wedge \mathbf{g} \nabla \wedge \mathbf{f} - \nabla \cdot \mathbf{g} \nabla \cdot \mathbf{f} = \int_{\Omega} \text{rot}(\mathbf{g}) \text{rot}(\mathbf{f}) - \text{div}(\mathbf{g}) \text{div}(\mathbf{f})$$

4.1.4 Conclusion

Le Laplacien continu permet de donner des solutions sous forme close à certains problèmes simples. Cependant, son utilisation pratique en informatique graphique ou en simulation numérique va nécessiter de le discrétiser sur un maillage. Il est donc important de trouver une bonne discrétisation du Laplacien qui préservera ses propriétés. Nous allons commencer par exposer l’état de l’art sur ce sujet, puis étudier en particulier les discrétisations du Laplacien basées sur les éléments finis (Chapitre 2) et le calcul extérieur discret (Chapitre 3).

4.2 Etat de l'art

La discrétisation du Laplacien joue un rôle central en informatique graphique et en particulier en traitement de la géométrie. Cet opérateur (et sa discrétisation) a été très étudié en raison de la grande variété de ses applications telles que le remaillage, la compression, la reconstruction, le calcul de surfaces minimales,... Nous classerons les études du Laplacien en deux grandes catégories : Laplaciens combinatoires et Laplaciens géométriques.

Laplaciens combinatoires : Un Laplacien *combinatoire* (ou Laplacien de *graphe*) ne dépend que de la connectivité du maillage, c'est à dire du graphe arêtes/sommets. L'analyse du spectre d'un Laplacien combinatoire a été utilisée par Taubin [1995] pour approximer des filtres passe bas. Cependant, le filtrage basé sur un Laplacien combinatoire est très dépendant de la discrétisation (cf Figure 4.1-A). Une base de vecteurs propres du Laplacien combinatoire est utilisée par Karni *et al.* [2000] pour la compression géométrique. Zhang [2004] étudie plusieurs variantes de Laplaciens combinatoires ainsi que leurs propriétés pour le filtrage spectral de la géométrie. Cependant, comme le note Meyer [2003], l'analogie entre un Laplacien de graphe et la transformation en cosinus discrets suppose un maillage uniforme de la surface. Ainsi, un Laplacien de graphe a deux propriétés indésirables sur un maillage triangulé :

- Différents plongements du même graphe auront les mêmes fonctions propres.
- Différents maillages de la même surface auront des fonctions propres différentes (cf Figure 4.1-A)

Ainsi, on cherchera à introduire plus de géométrie dans la définition du Laplacien.

Laplaciens géométriques et applications : Pour injecter plus de géométrie dans la définition d'un Laplacien discret, des poids basés sur les cotangentes des angles opposés à une même arête ont été introduits [Pinkall and Polthier 1993, Meyer *et al.* 2003]. Plusieurs raisonnements géométriques conduisent à la définition de ces poids. La méthode des éléments finis (cf Wardetzky *et al.* [2007a]) et le calcul extérieur discret conduisent aussi à des résultats similaires, et assurent la convergence vers le Laplacien continu (cf Hildebrandt *et al.* [2006] et Arnold *et al.* [2006]). Reuter *et al.* [2005] utilisent aussi les éléments finis pour calculer le spectre (i.e. les valeurs propres) d'un maillage, ce qui fournit une signature pour la classification de formes. Les cotangentes ont aussi été utilisées par Dong *et al.* [Dong *et al.* 2006] pour calculer une fonction propre utilisée pour guider un remaillage quadrangulaire. Les cotangentes seules sont encore dépendantes de la résolution du maillage (cf Figure 4.1-B), et elles doivent donc être pondérées par l'aire de l'étoile ou de la cellule duale de chaque sommet, ce qui fait perdre sa symétrie à la matrice du Laplacien, et la rend donc impropre à l'analyse spectrale (cf Figure 4.1-C). Une symétrisation empirique a été proposée dans [Levy 2006] (voir Figure 4.1-D). Dans ce chapitre, nous allons clarifier ces problèmes en nous basant sur le formalisme des éléments finis (Chapitre 2) et sur celui du calcul extérieur discret (Chapitre 3). En particulier, nous expliquerons comment retrouver la symétrie du Laplacien de façon simple et bien justifiée en exprimant le Laplacien dans une base adéquate. Cela garantira à la fois la prise en compte de la géométrie et l'orthogonalité des vecteurs propres (cf Figure 4.1-E). Notons que récemment, une preuve importante [Wardetzky *et al.* 2007b] montre qu'un Laplacien discret "parfait" vérifiant toutes les propriétés du Laplacien continu ne peut pas exister sur un maillage en général. Cela explique l'existence d'un nombre important de définitions du Laplacien discret, en fonction des propriétés désirées. Nous discuterons cela dans la Section 4.3.

4.3 Laplaciens discrets et discrétisés sur un maillage

La discrétisation du Laplacien joue un rôle central en informatique graphique puisque de nombreuses méthodes en dépendent. Il est donc très important de savoir précisément ce que nous faisons lorsque nous essayons de discrétiser le Laplacien, et en particulier de savoir quelles propriétés et garanties aura ce Laplacien discret.

4.3.1 Forme générale

Pour savoir comment discrétiser le Laplacien sur une surface triangulée M , la première question à se poser est de savoir comment les fonctions sont discrétisées sur M . L'approche la plus simple consiste à définir une valeur que prend la fonction sur chaque sommet, sans connaître la fonction ailleurs, et en particulier sans notion d'interpolation. Pour de telles fonctions purement discrètes, une forme générale a été proposée pour le Laplacien en se basant sur deux conditions de base :

1. Comme dans le cadre continu, le Laplacien doit être nul pour les fonctions constantes.
2. La valeur du Laplacien au sommet i ne doit dépendre que de la valeur de la fonction sur l'étoile de i

Cette deuxième condition vient de la localité du Laplacien continu, qui ne dépend que de la valeur de la fonction dans un voisinage arbitrairement petit du point où il est évalué.

Ces deux conditions peuvent être satisfaites en toute généralité en choisissant un poids w_{ij} sur chaque arête de M , et le Laplacien au sommet i est alors donné par :

$$\Delta f(i) = \sum_{iv \in \mathcal{S}_1(i)} w_{iv}(f(v) - f(i)). \quad (4.1)$$

Étant donné un ordonnancement des sommets, les fonctions discrètes sont vues comme des vecteurs et le Laplacien comme une matrice. Sous la forme (4.1) la matrice L du Laplacien est semi-définie et symétrique, de terme w_{ij} si ij est une arête, $-\sum_v w_{iv}$ sur la diagonale et 0 ailleurs. Si toutes les sommes $\sum_j w_{ij}$ sont positives (ou nulles), alors la matrice est (semi-)définie négative, c'est à dire que ses valeurs propres sont négatives (ou nulles). Le choix $w_{ij} = 1$ conduit à la définition du Laplacien de graphe qui ne dépend que de la connectivité, et est complètement indépendant de la géométrie. D'autres choix empiriques peuvent être faits pour ces poids, en fonctions de considérations et d'objectifs divers. Nous préférons ici utiliser les outils des éléments finis et du calcul extérieur discret qui nous permettront de définir des poids de façon justifiée. En particulier, nous aurons les preuves et les garanties de convergence présentées au Chapitre 2 et au Chapitre 3.

Cette forme générale pour les Laplaciens discrets est encore très restrictive puisque la valeur du Laplacien en un sommet i ne peut dépendre que de la valeur de la fonction dans l'étoile de i . Or cette étoile est une notion purement combinatoire et pas géométrique. En particulier, la localité géométrique conduit à une définition différente. Ainsi, en se basant sur la localité au sens de la connectivité, le Laplacien peut dépendre de la valeur de la fonction en un point plus éloigné (en distance géodésique) qu'un point dont il ne dépend pas. Une forme encore beaucoup plus générale serait alors d'autoriser $w_{ij} \neq 0$ pour toute paire de sommets (i, j) non nécessairement connectés, auquel cas la seule contrainte sur L serait $\text{span}(\mathbf{1}) \subset \ker(L)$ où $\mathbf{1}$ est le vecteur ne contenant que des 1.

Si on s'intéresse maintenant à des fonction définies sur toute la surface, la façon la plus simple et générique de faire est d'exprimer ces fonction dans une base, c'est à dire qu'on s'intéresse alors à un espace

de Hilbert. Dans ce cas, les coefficients d'une fonction dans cet espace ne correspondent pas forcément à leur valeur en un point. Dans un tel espace, le Laplacien sera représenté par une matrice carrée dont la dimension est celle de l'espace de Hilbert. Nous allons maintenant explorer cette approche, puis l'instancier sur les espaces de Hilbert présentés au Chapitre 2.

4.3.2 Formulation en éléments finis

Nous allons maintenant appliquer le mécanisme de l'approximation interne au Laplacien pour le discrétiser, c'est à dire pour trouver les coefficients de la matrice du Laplacien. Soit V_n un espace de Hilbert avec une base de fonctions $\{\Phi^i\}_{i=1..n}$. Toute fonction $f \in V_n$ s'écrit : $f = \Phi \mathbf{f} = \sum f_i \Phi^i$ où \mathbf{f} est le vecteur de terme f_i et $\Phi = (\Phi^1, \dots, \Phi^n)$ est la base de fonctions (organisée en vecteur ligne). Pour trouver le Laplacien Δf d'une fonction $f \in V_n$, on va appliquer l'approximation interne à l'équation :

$$\Delta f = g \quad (4.2)$$

où l'on utilisera l'espace $V_n = span(\Phi^i)$ à la fois comme espace de test et de solution :

$$\langle \Delta \Phi \mathbf{f}, \Phi^j \rangle = \langle \sum_i g_i \Phi^i, \Phi^j \rangle \quad (4.3)$$

qui s'écrit de façon matricielle :

$$K_{\Phi} \mathbf{f} = M_{\Phi} \mathbf{g}. \quad (4.4)$$

On appelle généralement K_{Φ} la matrice de *rigidité* et M_{Φ} la matrice de *masse*. Par définition d'une base de fonctions (et en particulier par l'indépendance linéaire), M_{Φ} est toujours inversible. Cela définit le Laplacien basé sur Φ comme :

$$\Delta_{\Phi} = M_{\Phi}^{-1} K_{\Phi}. \quad (4.5)$$

K_{Φ} et M_{Φ} ont pour termes :

$$\begin{aligned} (K_{\Phi})_i^j &= \langle \Delta \Phi^i, \Phi^j \rangle = \langle \nabla \Phi^i, \nabla \Phi^j \rangle \\ (M_{\Phi})_i^j &= \langle \Phi^i, \Phi^j \rangle \end{aligned}$$

M_{Φ} est la matrice du produit interne L_2 induit par la base Φ sur V_n , et est symétrique définie positive. K_{Φ} est la matrice d'un autre produit interne sur V_n . La norme induite par ce produit interne est généralement appelée énergie de Dirichlet :

$$\|f\|_K^2 = \mathbf{f}^t K_{\Phi} \mathbf{f} = \int \|\nabla f\|^2.$$

La matrice K_{Φ} est symétrique et semi-définie positive (puisque cette norme est nulle sur les fonctions constantes), exactement comme le Laplacien en tant qu'opérateur. D'ailleurs, si $\{\Phi^i\}_{i=1..n}$ est orthonormale, alors M_{Φ} est l'identité et la matrice du Laplacien est exactement K_{Φ} . Dans ce cas, Δ_{Φ} a donc les mêmes propriétés (symétrique semi-définie positive) que le Laplacien continu. Si $\{\Phi^i\}_{i=1..n}$ n'est pas orthogonale, alors le Laplacien $\Delta_{\Phi} = M_{\Phi}^{-1} K_{\Phi}$ perd sa symétrie (en tant que matrice), puisque le produit matriciel n'est pas commutatif (et donc le produit de deux matrices symétriques n'est pas symétrique). Par contre, Δ_{Φ} reste symétrique en tant qu'opérateur, c'est à dire vis à vis du produit interne :

$$\langle \Delta_{\Phi} f, g \rangle = \mathbf{f}^t K_{\Phi} M_{\Phi}^{-1} M_{\Phi} \mathbf{g} = \mathbf{f}^t K_{\Phi} \mathbf{g} = \langle f, \Delta_{\Phi} g \rangle.$$

En général, ce n'est pas un problème puisque la convergence est toujours garantie par le Théorème 2.1.2. Si on a besoin de la symétrie de la matrice (par exemple pour s'assurer que les vecteurs propres sont orthogonaux et que les valeurs propres sont réelles positives), on a besoin de transformer M_{Φ} pour la rendre orthonormale. Nous allons maintenant expliquer comment faire cela.

4.3.3 Retrouver la symétrie

La matrice du Laplacien n'est pas symétrique (alors que l'opérateur l'est) parce que la base sur laquelle elle est exprimée n'est pas orthonormale. Pour retrouver la symétrie, on a besoin de rendre Φ orthonormale, pour que la matrice de masse soit l'identité, ou en d'autres termes que le produit interne coïncide avec le produit scalaire. Dans la méthode des éléments finis, la solution consiste en deux étapes :

- Condensation de masse : on approxime M_Φ par \tilde{M}_Φ , qui est la matrice diagonale de terme $(\tilde{M}_\Phi)_i^i = \sum_j (M_\Phi)_{ij}$. On fait donc l'approximation (au niveau du produit interne) que la base de fonctions est orthogonale alors qu'elle ne l'est pas. Comme les éléments P_n sont très localisés et donc M_Φ a très peu de termes non diagonaux, et ceux-ci sont beaucoup plus faibles que les termes diagonaux. De plus, il est justifié dans [Prathap 1999] que cette approximation apporte plus de stabilité numérique.
- Normalisation : \tilde{M}_Φ est une bonne approximation du produit interne sur V_n . En d'autres termes, pour deux fonctions $f = \mathbf{f} \cdot \Phi$ et $g = \mathbf{g} \cdot \Phi$, leur produit interne est approximé par $\langle f, g \rangle \approx \mathbf{f}^T \tilde{M}_\Phi \mathbf{g}$. On peut donc normaliser Φ pour ce produit interne : $\bar{\Phi} = \Phi \tilde{M}_\Phi^{-1/2}$ (Φ est un vecteur ligne), de sorte que :

$$f = \Phi \mathbf{f} = \Phi \tilde{M}_\Phi^{-1/2} \tilde{M}_\Phi^{1/2} \mathbf{f} = \bar{\Phi} \bar{\mathbf{f}} \quad \bar{\Phi} = \Phi \tilde{M}_\Phi^{-1/2} \quad \bar{\mathbf{f}} = \tilde{M}_\Phi^{1/2} \mathbf{f} \quad (4.6)$$

Dans cette base (et seulement cette base), le produit interne est le produit scalaire habituel sur les vecteurs :

$$\langle f, g \rangle \approx \mathbf{f}^T \tilde{M}_\Phi \mathbf{g} = \bar{\mathbf{f}}^T \bar{\mathbf{g}}$$

et le Laplacien est :

$$\Delta_{\bar{\Phi}} = \tilde{M}_\Phi^{-1/2} K_\Phi \tilde{M}_\Phi^{-1/2}$$

qui est maintenant symétrique.

Nous allons maintenant appliquer cette méthode avec les bases d'éléments P_n et bilinéaires. Nous allons ensuite voir comment transposer ces concepts au calcul extérieur discret.

4.3.4 Laplacien d'interpolation

Nous appelons Laplacien d'interpolation un Laplacien Δ_Φ où Φ est une base de fonctions interpolantes. Nous allons maintenant présenter le cas des bases de fonctions que nous avons présentées dans le Chapitre 2 : les bases de fonctions interpolantes P_n ($n \leq 3$) et bilinéaires.

Laplacien P_1

Les fonctions d'interpolation P_1 sont les coordonnées barycentriques $P_1^i(\mathbf{x}) = \lambda_{i,jk}(\mathbf{x})$ d'un point $\mathbf{x} \in [ijk]$ relatives au sommet i , de sorte que :

$$\sum_{v=i,j,k} \Phi^v(\mathbf{x})[v] = \mathbf{x} \quad \text{et} \quad \sum_{v=i,j,k} \Phi^v(\mathbf{x}) = 1$$

Les intégrales impliquant des coordonnées barycentriques en dimension N peuvent être calculées grâce à une formule très générale :

$$\int_{\mathbf{x} \in s_n} \prod_{i=1}^{n+1} \Phi^i(\mathbf{x})^{\alpha_i} dp = |s_n| \frac{n! \prod_{i=1}^{n+1} \alpha_i!}{(n + \sum_{i=1}^{n+1} \alpha_i)!} \quad (4.7)$$

où s_n est un n -simplexe et $\Phi^i(\mathbf{x})$ sont les fonctions P_1 (coordonnées barycentriques). Cela permet de calculer les coefficients de M (avec $n = 2$) :

$$M_i^i = \langle \Phi^i, \Phi^i \rangle = \int_{\mathbf{x} \in [St_2(i)]} \Phi^i(\mathbf{x}) \Phi^i(\mathbf{x}) d\mathbf{x} = \frac{\|St_2(i)\|}{6} = \frac{1}{6} \sum_{t \in St_2(i)} \|t\| \quad (4.8)$$

$$M_i^j = \langle \Phi^i, \Phi^j \rangle = \int_{\mathbf{x} \in [St_2(i)]} \Phi^i \Phi^j d\mathbf{x} = \frac{\|St_2(ij)\|}{12} = \frac{\|ijk\| + \|jil\|}{12} \quad (4.9)$$

où ijk et jil sont les deux triangles partageant l'arête ij . M_i^j est nulle si i et j ne sont pas connectés ($ij \notin \mathcal{E}$).

Pour calculer les termes de K , on utilise l'expression (2.7) des gradients des coordonnées barycentriques :

$$K_i^i = \langle \nabla \Phi^i, \nabla \Phi^i \rangle = \int_{\mathbf{x} \in [St_2(i)]} \nabla \Phi^i(\mathbf{x}) \cdot \nabla \Phi^i(\mathbf{x}) d\mathbf{x} = \sum_{iv_1v_2 \in St_2(i)} \frac{\|v_1v_2\|^2}{2\|iv_1v_2\|} = \sum_{iv_1v_2 \in St_2(i)} s_{i,v_1v_2} \quad (4.10)$$

où l'on a défini $s_{i,jk} = \|jk\|^2/2\|ijk\|$.

$$K_i^j = \langle \nabla \Phi^i, \nabla \Phi^j \rangle = \int_{\mathbf{x} \in [St_2(ij)]} \nabla \Phi^i(\mathbf{x}) \cdot \nabla \Phi^j(\mathbf{x}) d\mathbf{x} = \frac{|jk| \cdot |ki|}{4\|ijk\|} + \frac{|il| \cdot |lj|}{4\|jil\|} = -\frac{1}{2}(c_{k,ij} + c_{l,ji}) \quad (4.11)$$

où ijk et jil sont les deux triangles partageant l'arête ij . Les coefficients

$$c_{i,jk}^t = \frac{|ij| \cdot |ik|}{2\|ijk\|} = \frac{\|ij\| \cdot \|ik\| \cos(\angle(|ij|, |ik|))}{\|ij\| \cdot \|ik\| \sin(\angle(|ij|, |ik|))} = \cot(\angle(|ij|, |ik|))$$

sont appelés les poids *cotangente*. Nous supposons que notre maillage n'a pas de triangles plats, de sorte que ces poids ne peuvent être infinis. Toutefois, ils peuvent être négatifs, ce qui pose des problèmes dans certaines applications (nous détaillerons ce problème dans la Section 4.3.8) Notons également que K_i^i n'a pas besoin d'être calculés en utilisant (4.10) car nous avons simplement $K_i^i = -\sum_{j \neq i} K_i^j$.

Laplacien P_2

Les éléments P_2 sont définis en (2.8) et leurs gradients sont donnés dans (2.9). Avec l'aide d'un outil de calcul formel, on peut calculer les coefficients de M et K . Nous pouvons maintenant calculer les coefficients de la matrice de masse M en P_2 :

M	P_2^i	P_2^{ij}
P_2^i	$\ St_2(i)\ /30$	
P_2^j	$-\ St_2(ij)\ /180$	
P_2^{ij}	0	$8\ St_2(ij)\ /45$
P_2^{jk}	$-\ ijk\ /45$	$4\ ijk\ /45$

et les autres coefficients sont nuls. Pour la matrice de rigidité K , les coefficients sont donnés par :

K	P_2^i	P_2^{ij}
P_2^i	$\sum_{t \in St_2(i)} s_{i,jk}/2$	
P_2^j	$-\sum_{t \in St_2(ij)} c_{k,ij}/6$	
P_2^{ij}	$2\sum_{t \in St_2(ij)} c_{k,ij}/3$	$4\sum_{t \in St_2(ij)} (s_{i,jk}^t + c_{k,ij}^t + s_{j,ki}^t)/3$
P_2^{jk}	0	$2(c_{k,ij} + c_{i,jk} + s_{j,ki} + 2c_{j,ik})/3$

et de nouveau les termes diagonaux de K peuvent être calculés en utilisant $K_i^i = -\sum_{j \neq i} K_i^j$.

Laplacien P_3

Les éléments P_3 sont définis en (2.10) et leurs gradients sont donnés dans (2.11). Avec l'aide d'un outil de calcul formel, on peut calculer les coefficients de M et K :

M	P_3^i	P_3^{ij}	P_3^{ijk}
P_3^i	$19 St_2(i) /1680$		
P_3^j	$11 St_2(ij) /6720$		
P_3^{ij}	$3 St_2(ij) /1120$	$9 St_2(ij) /112$	
P_3^{ji}	0	$-9 St_2(ij) /320$	
P_3^{jk}	$9 ijk /2240$	$-9 ijk /448$	
P_3^{kj}	$9 ijk /2240$	$-9 ijk /1120$	
P_3^{ijk}	$3 ijk /560$	$27 ijk /1120$	$81 ijk /280$

K	P_3^i	P_3^{ij}
P_3^i	$17 \sum_{iv_1v_2 \in St_2(i)} s_{i,v_1v_2}/40$	
P_3^j	$7 \sum_{ijv \in St_2(ij)} c_{v,ij}/80$	
P_3^{ij}	$\sum_{ijv \in St_2(ij)} (3s_{i,jv} + 57c_{v,ij}^t)/80$	$27 \sum_{t \in St_2(ij)} (c_{v,ij} + s_{i,jv} + s_{j,vi})/80$
P_3^{ji}	$\sum_{ijv \in St_2(ij)} (3s_{i,jv} - 24c_{v,ij})/80$	$27 \sum_{ijv \in St_2(ij)} (c_{v,ij} - s_{i,jv} - s_{j,vi}^t)/80$
P_3^{jk}	$(3c_{k,ij} + 3c_{j,ik})/80$	$-27(2c_{j,ik} + 2c_{i,jk} + c_{k,ij} + s_{j,ki})/160$
P_3^{kj}	$(3c_{k,ij} + 3c_{j,ik})/80$	$-27(2c_{j,ik} + 2c_{i,jk} + c_{k,ij} + s_{j,ki})/160$
P_3^{ijk}	$9(s_{i,jk} + c_{k,ij} + c_{j,ik})/80$	$81(s_{j,ki} + c_{k,ij} + c_{i,jk} + c_{j,ik})/80$
	P_3^{ijk}	
P_3^{ijk}	$81(s_{i,jk} + c_{k,ij} + c_{j,ik} + s_{j,ki} + c_{i,jk} + s_{k,ij}^t)/20$	

et à nouveau on peut calculer les termes diagonaux de K en utilisant $K_i^i = -\sum_{j \neq i} K_i^j$.

Comme les solutions des EDPs impliquant un Laplacien sont habituellement lisses, il est intéressant de construire des espaces de Hilbert basés sur des polynômes d'ordre supérieur (en coordonnées barycentriques). Étant donné que ces espaces sont plus grands, leur utilisation permettra d'approcher plus finement la solution. En outre, la solution que nous cherchons à approcher est en général lisse, et donc son approximation le sera aussi, même si nous ne pouvons garantir une régularité meilleure que C_0 .

Dans le cas où notre maillage n'est pas considéré comme une approximation mais comme la surface exacte, ou que nous voulons en préserver certaines caractéristiques, il ne doit pas être modifié, et l'utilisation d'éléments d'ordre supérieur garantit plus de précision que la subdivision ordinaire pour un nombre donné de degrés de liberté, au prix d'une matrice un peu moins creuse. L'implémentation est également un peu plus délicate, mais assez directe en utilisant les coefficients donnés ci-dessus (calculés avec l'aide de Maple[®]). Si nous sommes autorisés à remailler, on peut utiliser le remaillage pour améliorer la qualité (forme) des triangles, mais les éléments d'ordre supérieur sont toujours souhaitables si nous voulons améliorer la précision et la régularité.

Comme la complexité augmente rapidement, nous n'examinerons pas les Laplaciens d'ordre supérieur à 3, pour lesquels nous conseillons d'utiliser une bibliothèque de calcul formel.

Laplacien bilinéaire

Pour la base de fonctions de l'interpolation bilinéaire, les coefficients de M sont donnés par :

$$M_{ij}^{kl} = \langle B_{ij}, B_{kl} \rangle_{L_2} = \int_S B_{ij} B_{kl} dS = \int_{[0,1]^2} B_{ij} B_{kl} \text{Det}(J) du dv$$

Définissons $J_{ij} = \text{Det}(J(i, j)) = \|\mathbf{v}_{uj} \times \mathbf{v}_{iv}\|$. Comme l'expression exacte de $\text{Det}(J)$ est difficile à intégrer analytiquement, on approxime $\text{Det}(J)$ par son interpolation bilinéaire : $\text{Det}(J) \approx \sum_{i,j} J_{ij} B_{ij}(u, v)$ auquel cas :

$$\begin{aligned} M_{ij}^{ij} &\approx \frac{1}{144} (9J_{ij} + 3(J_{(1-i)j} + J_{i(1-j)}) + J_{(1-i)(1-j)}) \\ M_{ij}^{i(1-j)} &\approx \frac{1}{144} (3(J_{i0} + J_{i1}) + J_{(1-i)0} + J_{(1-i)1}) \\ M_{ij}^{(1-i)(1-j)} &\approx \frac{1}{144} (J_{00} + J_{01} + J_{10} + J_{11}) \end{aligned}$$

La matrice de rigidité K est donnée par :

$$K_{ij}^{kl} \approx \langle \nabla B_{ij}, \nabla B_{kl} \rangle_{L_2} = \int_{[0,1]^2} J^\perp B_{ij} \cdot J^\perp B_{kl} \text{Det}(J)^{-1} du dv$$

Cette fois, on simplifie l'intégrale en approximant $\text{Det}(J)^{-1}$ par son interpolation bilinéaire : $\text{Det}(J)^{-1} \approx \sum_{i,j} J^{-1}(i, j) B_{ij}(u, v)$ ce qui donne (avec l'aide Maple[©]) :

$$\begin{aligned} K_{00}^{00} &\approx (3J_{01}^{-1} + J_{11}^{-1} + 9J_{00}^{-1} + 3J_{10}^{-1}) \mathbf{v}_{u0}^2 + (3J_{00}^{-1} + 3J_{11}^{-1} + J_{10}^{-1} + 9J_{01}^{-1}) \mathbf{v}_{u1}^2 \\ &\quad + (3J_{01}^{-1} + J_{11}^{-1} + 9J_{00}^{-1} + 3J_{10}^{-1}) \mathbf{v}_{0v}^2 + (9J_{10}^{-1} + 3J_{00}^{-1} + 3J_{11}^{-1} + J_{01}^{-1}) \mathbf{v}_{1v}^2 \\ &\quad + ((2J_{11}^{-1} + 2J_{10}^{-1} + 6J_{01}^{-1} + 6J_{00}^{-1}) \mathbf{v}_{u1} + (-18J_{00}^{-1} - 6J_{01}^{-1} - 6J_{10}^{-1} - 2J_{11}^{-1}) \mathbf{v}_{0v} + (-6J_{00}^{-1} - 6J_{10}^{-1} - 2J_{01}^{-1} - 2J_{11}^{-1}) \mathbf{v}_{1v}) \mathbf{v}_{u0} \\ &\quad + ((-2J_{11}^{-1} - 2J_{10}^{-1} - 6J_{01}^{-1} - 6J_{00}^{-1}) \mathbf{v}_{0v} + (-2J_{10}^{-1} - 2J_{00}^{-1} - 2J_{11}^{-1} - 2J_{01}^{-1}) \mathbf{v}_{1v}) \mathbf{v}_{u1} + (2J_{01}^{-1} + 2J_{11}^{-1} + 6J_{10}^{-1} + 6J_{00}^{-1}) \mathbf{v}_{1v} \mathbf{v}_{0v} \end{aligned}$$

$$\begin{aligned} K_{00}^{01} &\approx (J_{01}^{-1} + J_{11}^{-1} + 3J_{00}^{-1} + 3J_{10}^{-1}) \mathbf{v}_{u0}^2 + ((2J_{11}^{-1} + 2J_{10}^{-1} + 2J_{00}^{-1} + 2J_{01}^{-1}) \mathbf{v}_{u1} \\ &\quad + (2J_{01}^{-1} + 6J_{00}^{-1}) \mathbf{v}_{0v} + (-2J_{11}^{-1} - 6J_{10}^{-1}) \mathbf{v}_{1v}) \mathbf{v}_{u0} + (J_{00}^{-1} + J_{10}^{-1} + 3J_{01}^{-1} + 3J_{11}^{-1}) \mathbf{v}_{u1}^2 \\ &\quad + ((2J_{01}^{-1} + 2J_{00}^{-1}) \mathbf{v}_{0v} + (-2J_{11}^{-1} - 2J_{10}^{-1}) \mathbf{v}_{1v}) \mathbf{v}_{u1} + (-3J_{01}^{-1} - J_{11}^{-1} - 9J_{00}^{-1} - 3J_{10}^{-1}) \mathbf{v}_{0v}^2 \\ &\quad + (-6J_{00}^{-1} - 6J_{10}^{-1} - 2J_{01}^{-1} - 2J_{11}^{-1}) \mathbf{v}_{1v} \mathbf{v}_{0v} + (-3J_{11}^{-1} - J_{01}^{-1} - 9J_{10}^{-1} - 3J_{00}^{-1}) \mathbf{v}_{1v}^2 \end{aligned}$$

$$\begin{aligned} K_{00}^{11} &\approx (-J_{01}^{-1} - J_{11}^{-1} - 3J_{00}^{-1} - 3J_{10}^{-1}) \mathbf{v}_{u0}^2 + ((-2J_{10}^{-1} - 2J_{00}^{-1} - 2J_{11}^{-1} - 2J_{01}^{-1}) \mathbf{v}_{u1} \\ &\quad + (6J_{00}^{-1} + 4J_{01}^{-1} + 4J_{10}^{-1} + 2J_{11}^{-1}) \mathbf{v}_{0v} + (4J_{00}^{-1} + 10J_{10}^{-1} + 2J_{01}^{-1} + 4J_{11}^{-1}) \mathbf{v}_{1v}) \mathbf{v}_{u0} \\ &\quad + (-J_{00}^{-1} - 3J_{11}^{-1} - J_{10}^{-1} - 3J_{01}^{-1}) \mathbf{v}_{u1}^2 + ((4J_{11}^{-1} + 2J_{10}^{-1} + 10J_{01}^{-1} + 4J_{00}^{-1}) \mathbf{v}_{0v} \\ &\quad + (4J_{10}^{-1} + 2J_{00}^{-1} + 6J_{11}^{-1} + 4J_{01}^{-1}) \mathbf{v}_{1v}) \mathbf{v}_{u1} + (-J_{10}^{-1} - 3J_{01}^{-1} - J_{11}^{-1} - 3J_{00}^{-1}) \mathbf{v}_{0v}^2 \\ &\quad + (-2J_{10}^{-1} - 2J_{00}^{-1} - 2J_{11}^{-1} - 2J_{01}^{-1}) \mathbf{v}_{1v} \mathbf{v}_{0v} + (-3J_{10}^{-1} - J_{00}^{-1} - 3J_{11}^{-1} - J_{01}^{-1}) \mathbf{v}_{1v}^2 \end{aligned}$$

Ces poids peuvent être utilisés pour construire le Laplacien sur un maillage quadrangulé sans subdivision. Il est conseillé d'utiliser un outil de calcul formel, ou d'approximer $\text{Det}(J)$ (moins finement) par une constante car ces coefficients peuvent se révéler pénibles à saisir manuellement.

4.3.5 Conditions aux bords

Dans le cas où le maillage sur lequel on travaille a des bords, ils doivent être traités avec précaution lorsqu'on résout $\Delta f = g$. Dans la méthode des éléments finis, il y a essentiellement deux façons de traiter les bords :

- Les conditions au bord de Dirichlet : on impose à la fonction f d'être nulle sur la frontière.
- Les conditions au bord de Neumann : on impose à la fonction f d'avoir un gradient nul à travers le bord.

Conditions de Dirichlet

Pour distinguer l'intérieur du bord, appelons \mathcal{V}^{int} l'ensemble des sommets *intérieurs* du maillage, c'est à dire ceux qui ne sont pas sur son bord. Pour résoudre (4.2) avec des conditions de Dirichlet, on assure les conditions au bord par construction en écrivant f dans la base $\{\Phi^i | i \in \mathcal{V}^{int}\}$, qui est la base des Φ dont on a retiré les Φ^i pour lesquels i est sur le bord. On peut projeter l'équation (4.2) sur cette base :

$$\forall j \in \mathcal{V}^{int} \sum_{i \in \mathcal{V}^{int}} f_i \langle \Delta \Phi_i, \Phi_j \rangle = \sum_{i \in \mathcal{V}^{int}} g_i \langle \Phi_i, \Phi_j \rangle$$

$$K(\mathcal{V}^{int}, \mathcal{V}^{int}) \mathbf{f}_{int} = M(\mathcal{V}^{int}, \mathcal{V}) \mathbf{g}$$

où $K(\mathcal{V}_1, \mathcal{V}_2)$ désigne la sous-matrice de M obtenue en ne gardant que les lignes d'indice dans \mathcal{V}_1 et le colonnes d'indice dans \mathcal{V}_2 , et \mathbf{f}_{int} est le vecteur des valeurs de f sur les sommets intérieurs.

Conditions de Neumann

Dans l'équation (4.2), les conditions de Neumann sont implicitement vérifiées lorsqu'on écrit $\langle \Delta \Phi_i, \Phi_j \rangle = \langle \nabla \Phi_i, \nabla \Phi_j \rangle$, puisque la différence de ces termes est le flux du gradient à travers le bord. Ainsi, on a :

$$\forall j \in \mathcal{V} \sum_{i \in \mathcal{V}} f_i \langle \nabla \Phi_i, \nabla \Phi_j \rangle = \sum_{i \in \mathcal{V}} g_i \langle \Phi_i, \Phi_j \rangle$$

En d'autres termes, on a exactement la même équation qu'en absence de bord :

$$K \mathbf{f} = M \mathbf{g}$$

Après cette présentation du Laplacien obtenu par la méthode des éléments finis, nous allons nous intéresser à la façon de définir le Laplacien en calcul extérieur discret.

4.3.6 Laplacien du calcul extérieur discret

Sur la base des opérateurs vus au Chapitre 3, on peut définir l'équivalent discret de l'opérateur de *Laplace-de Rham* agissant sur les k -formes. Il est donné par : $\Delta_k = d_{k-1} \delta_k + \delta_{k+1} d_k$. Dans cette thèse, on ne s'intéresse qu'au laplacien sur les scalaires : $\Delta = \Delta_0$. Comme δ_0 et d_2 ne sont pas définis (et donc choisis nuls par convention), $\Delta_0 = \delta_1 d_0 = \star_0^{-1} d_1^T \star_1 d_0$ et ses coefficients sont donnés par :

$$(\Delta_0)_i^j = \frac{\cot(\beta_{k,ij}) + \cot(\beta_{l,ji})}{\|\star i\|} \quad ; \quad (\Delta_0)_i^i = - \sum_j (\Delta_0)_i^j$$

où $\beta_{k,ij}$ est l'angle au sommet k (opposé à l'arête ij) dans le triangle ijk . Pour les surfaces avec bords, si l'arête ij est sur le bord, le coefficient $(\Delta_0)_i^j$ correspondant n'aura qu'un terme cot correspondant au seul triangle basé sur ij , et le calcul de $\|\star i\|$ devra prendre en compte le fait que cette cellule est coupée par le bord. Cela correspond à la formulation P_1 en éléments finis avec des conditions aux limites de Neumann, à l'exception d'une différence mineure.

Remarque 4.3.1 (Coefficients de masse) *Dans le calcul extérieur discret, l'étoile de Hodge \star_0 correspond à la matrice de masse M des éléments finis P_1 . Cependant, les coefficients de \star_0 (aires des cellules duales) sont légèrement différents des poids calculés par les éléments finis avec condensation de masse (aires des étoiles $St_2(i)$ des sommets i). Cette différence vient du fait que les fonctions Φ^i de la base P_1 ont pour support $St_2(i)$. En revanche, le calcul extérieur discret est basé sur un schéma mixte éléments finis/volumes finis et ces volumes finis sont ceux des cellules duales.*

Le Laplacien Δ ainsi défini (on ne met pas l'indice 0 puisqu'on ne s'intéressera qu'au Laplacien scalaire dans cette thèse) est connu pour avoir toutes les propriétés désirées sur un maillage qui admet un dual circoncentrique [Wardetzky et al. 2007a], à part qu'il perd (en tant que matrice) la symétrie de son équivalent discret : $\Delta_i^j \neq \Delta_j^i$ (il ne perd pas la symétrie en tant qu'opérateur). On retrouve cette structure de symétrie matricielle de la même façon (mais encore plus simplement) qu'en éléments finis. Comme on l'a dit, l'étoile de Hodge \star_0 joue le rôle de matrice de masse, c'est à dire qu'elle est la matrice du produit interne (cf Chapitre 3), puisqu'on a $\langle \Phi_i, \Phi_j \rangle = (\Phi_i)^T \star_0 \Phi_j$. Contrairement à ce qu'on avait vu dans la méthode des éléments finis, cette matrice est déjà diagonale, donc on n'a même pas besoin de faire une condensation de masse, mais une simple normalisation :

$$\bar{\Phi}_i = \star_0^{-1/2} \Phi_i$$

où Φ est maintenant la base canonique des 0-formes : $\Phi_i = 1$ au sommet i et 0 aux autres, et $(\bar{\Phi}_i)$ est la base canonique normalisée. Dans la base $(\bar{\Phi}_i)$, le Laplacien $\bar{\Delta}$ est maintenant symétrique, et il est donné par :

$$\bar{\Delta} = \star_0^{1/2} \Delta \star_0^{-1/2} = -\star_0^{-1/2} d_1^T \star_1 d_0 \star_0^{-1/2} \quad ; \quad \bar{\Delta}_i^j = -\frac{\cot \beta_{ij} + \cot \beta'_{ij}}{\sqrt{\|\star i\| \|\star j\|}} \quad (4.12)$$

Cette symétrisation est faite pour préparer les matrices à un calcul de leurs valeurs propres, puisqu'un tel calcul est plus efficace et offre des garanties (valeurs propres réelles, vecteurs propres orthogonaux). Avant de conclure ce chapitre, on va discuter deux aspects des Laplaciens discrets :

- La localisation du Laplacien P_1 (le plus utilisé)
- Les conditions auxquelles un Laplacien discret a toutes les propriétés du Laplacien continu.

4.3.7 Localisation du Laplacien P_1 par le graphe mixte

Dès qu'on dispose d'une interpolation pour une fonction définie sur une surface triangulée, on peut se demander s'il est possible de calculer explicitement son Laplacien (dans le cadre continu). Malheureusement, les interpolations P_n ne sont en général que C_0 de sorte qu'elles manquent de régularité pour faire ce calcul explicite. Toutefois, nous avons facilement accès à l'intégrale du Laplacien sur un sous domaine de la surface par :

$$\int_{\Omega} \Delta f = \int_{\partial\Omega} \nabla f \cdot \mathbf{n}$$

où \mathbf{n} est le vecteur normal à $\partial\Omega$ (plongé dans la surface). Cela va nous permettre de localiser les Laplacien P_1 des fonctions définies sur des surfaces triangulées. Pour ce faire, nous allons définir le graphe mixte

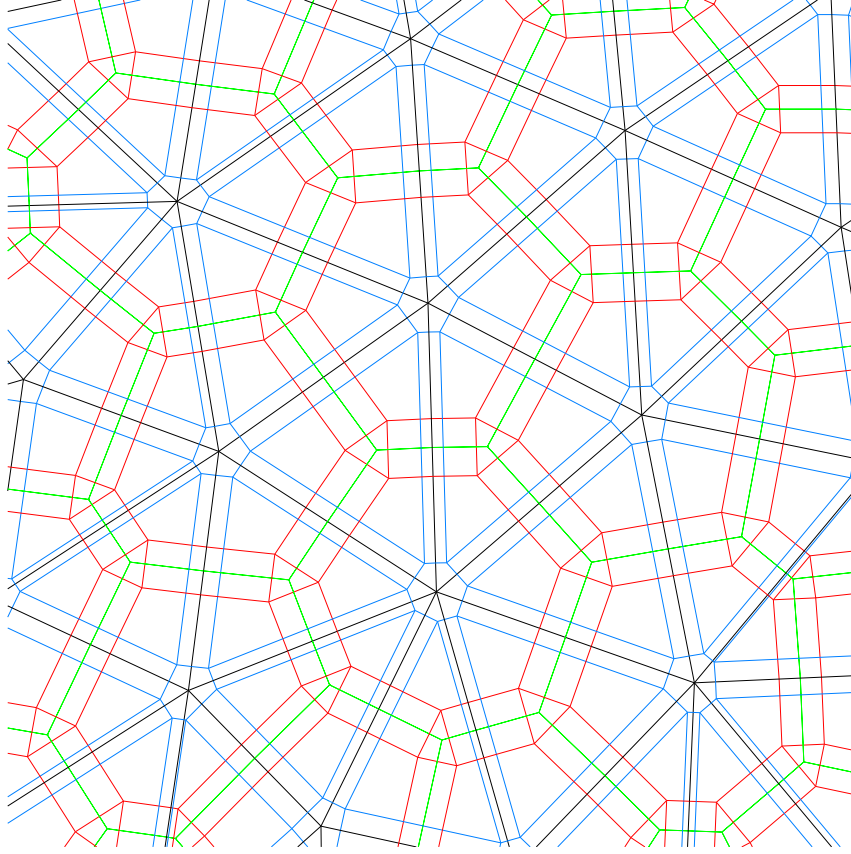


FIG. 4.2 – Une surface triangulée (noir) et son graphe dual (vert), les graphes mixtes pour $s = 1/5$ (bleu) et $s = 4/5$ (rouge).

comme la somme de Minkovsky des graphes primaux et duaux (voir Figure 4.2) :

$$M^+(s) = (1-s)M + s \star M$$

où s est un paramètre d'échelle qui varie de 0 ($M^+(0) = M$) à 1 ($M^+(1) = \star M$). Dans le graphe mixte, nous avons trois types de cellules :

1. $[i^+] = (1-s)[i] + s[\star i]$ sont les cellules duales des sommets, mis à l'échelle par s .
2. $[ij^+] = (1-s)[ij] + s[\star ij]$ est un rectangle aligné avec les arêtes ij et leurs duales $\star ij$, de longueur $(1-s)||ij||$ et de largeur $s||\star ij|| = s(\cot(\beta_{k,ij}) + \cot(\beta_{l,ji}))||ij||$.
3. $[ijk^+] = (1-s)[ijk] + s[\star ijk]$ est un triangle mis à l'échelle par $(1-s)$.

Toutes ces cellules sont de dimension 2. Ce qui est intéressant est d'étudier le comportement de l'intégrale du Laplacien sur ces cellules quand s tend vers 0 et s tend vers 1, puisque la dimension des cellules change aux limites.

Dans chaque triangle ijk , le gradient est constant :

$$\nabla f_{ijk} = \frac{|ij|^\perp f_k + |jk|^\perp f_i + |ki|^\perp f_j}{2||ijk||}$$

Cellule	i^+	ij^+	ijk^+
Laplacien intégré	s	1-s	0
Laplacien moyen	1/s	1/s	0

TAB. 4.1 – Ordre de grandeur du Laplacien intégré et moyen sur les cellules mixtes.

Dans le calcul des intégrales de gradients, on retrouve plusieurs termes de la forme :

$$\frac{|ij|^\perp \cdot |jk|^\perp}{2||ijk||} = \frac{|ij| \cdot |jk|}{|ij| \wedge |ik|} = -\frac{||ij|| \cdot ||jk|| \cos(\beta_{j,ki})}{||ij|| \cdot ||jk|| \sin(\beta_{j,ki})} = \cot(\beta_{j,ki}) =_{def} c_{j,ki}$$

où $\beta_{j,kl}$ est l'angle au sommet j dans le triangle ijk . Pour intégrer le gradient le long d'une arête duale $\star ij$, on doit la séparer entre les deux triangles :

$$\begin{aligned} \int_{[\star ij]} \Delta f &= \int_{\partial[\star ij]} \nabla f \cdot \mathbf{n} \\ &= (||[\star ij] \cap [ijk]|| \nabla f_{ijk} + ||[\star ij] \cap [jil]|| \nabla f_{jil}) \cdot |ji| / ||ji|| \\ &= (c_{k,ij} \nabla f_{ijk} + c_{l,ji} \nabla f_{jil}) \cdot |ji| \\ &= c_{k,ij} \frac{|ij|^\perp \cdot |ji| f_k + |jk|^\perp \cdot |ji| f_i + |ki|^\perp \cdot |ji| f_j}{||ijk||} + c_{l,ji} \frac{|ji|^\perp \cdot |ji| f_i + |il|^\perp \cdot |ji| f_j + |lj|^\perp \cdot |ji| f_i}{||jil||} \\ &= 2c_{k,ij}(-f_i + f_j) + 2s \cdot c_{l,ji}(f_j - f_i) = 2(c_{k,ij} + c_{l,ji})(f_j - f_i) \end{aligned}$$

qui sont les poids cotangente. En intégrant sur toute la cellule mixte $[i^+]$, on obtient :

$$\int_{[i^+]} \Delta f = \int_{\partial[i^+]} \nabla f \cdot \mathbf{n} = s \sum_{ij \in St_2(i)} L_{ij}^*$$

Sur $[ij^+]$ l'intégrale de $\nabla f \cdot \mathbf{n}$ le long des arêtes du rectangle parallèles à $[\star ij]$ se simplifient, donc il reste :

$$\begin{aligned} \int_{[ij^+]} \Delta f &= \int_{\partial[ij^+]} \nabla f \cdot \mathbf{n} = (\nabla f_{ijk} - \nabla f_{jil}) \cdot (1-s)|ij|^\perp = \\ &= \left(\frac{|ij|^\perp f_k + |jk|^\perp f_i + |ki|^\perp f_j}{2||ijk||} - \frac{|ji|^\perp f_i + |il|^\perp f_j + |lj|^\perp f_i}{2||jil||} \right) \cdot (1-s)|ij|^\perp = \\ &= (1-s) \left((c_{j,il} - c_{j,ki}) f_i + (c_{i,lj} - c_{i,jk}) f_j + |ij|^2 \left(\frac{f_k}{2||ijk||} - \frac{f_l}{2||jil||} \right) \right) \end{aligned}$$

Et enfin comme ∇f_{ijk} est constant sur ijk :

$$\int_{[ijk^+]} \Delta f = \int_{\partial[ijk^+]} \nabla f \cdot \mathbf{n} = 0$$

Les ordres de grandeur du Laplacien intégré et de sa moyenne sur une cellule mixte sont données dans la Table 4.1. On voit que le Laplacien moyen tend vers ∞ quand s tend vers 0 sur i^+ et ij^+ , et qu'il est nul à l'intérieur des triangles. Cela signifie que le Laplacien est localisé sur les arêtes primales du maillage M . Ainsi, l'intégrale du Laplacien sur n'importe quel sous ensemble $\Omega \subset [M]$ ne dépend que des arêtes

de M qui sont dans Ω . En d'autres termes, le Laplacien est une distribution sur les arêtes, donnée par une constante sur chaque arête :

$$L(ij) = (c_{j,il} - c_{j,ki})f_i + (c_{i,lj} - c_{i,jk})f_j + |ij|^2 \left(\frac{f_k}{2||ijk||} - \frac{f_l}{2||jil||} \right)$$

et l'intégrale sur Ω est alors :

$$\int_{\Omega} \Delta f = \int_{\Omega \cap [\mathcal{E}]} L(e) = \sum_{e \in \mathcal{E}} \frac{||e \cap \Omega||}{||e||} L(e)$$

En calcul extérieur discret et dans la méthode des éléments finis (P_1), on cherche le Laplacien d'une fonction dans le même espace que cette fonction, alors qu'on voit bien ici que le Laplacien a une structure très différente. Cette distribution de Laplacien est donc reprojctée sur l'espace dans lequel on cherche le résultat. Dans le cas des éléments finis P_1 , il est simplement reprojcté sur les fonctions P_1 :

$$\int \Delta f \Phi^i = 1/2 \sum_{v \in S_1(i)} L(iv)$$

Ceci correspond à la condensation de masse, puisque la projection suppose que la base est orthogonale alors qu'elle ne l'est pas. Dans le cas du calcul extérieur discret, il est projeté sur les constantes par morceaux sur les cellules duales, et cette valeur intégrale (2-forme) est transformée en valeur moyenne (0-forme) en divisant par l'aire de ces cellules duales, ce qui est un autre moyen d'interpréter la différence entre les termes de masse en éléments finis P_1 et en calcul extérieur discret.

4.3.8 Limites d'un Laplacien discret

Nous finissons ce chapitre avec la présentation de certaines limites connues à la discrétisation du Laplacien, exhibées dans [Wardetzky et al. 2007b]. Ces limites concernent quatre caractéristiques du Laplacien continu :

1. Précision linéaire : le Laplacien est nul pour les fonctions linéaire
2. Symétrie : le Laplacien est un opérateur symétrique : il vérifie $\langle \Delta f, g \rangle = \langle f, \Delta g \rangle$
3. Extrema : une fonction harmonique (de Laplacien nul) ne peut pas avoir d'extrema ailleurs que sur la frontière de son domaine de définition.
4. Localité : La valeur du Laplacien d'une fonction f en un point \mathbf{x} ne dépend que des valeurs de f dans un voisinage arbitrairement petit de \mathbf{x} .

Pour la matrice du Laplacien discret, cela se traduit par :

1. Précision linéaire : $\sum_i \Delta_i^j = 0$ ce qui est toujours le cas dans nos définitions même les plus génériques du Laplacien discret.
2. Symétrie : $\Delta_i^j = \Delta_j^i$
3. Extrema : $\Delta_i^j \geq 0$.
4. Localité : $\Delta_i^j = 0$ si $ij \notin \mathcal{E}$ (ij n'est pas une arête du maillage).

En pratique, il n'est possible de construire un Laplacien avec ces propriétés que si le maillage admet un dual orthogonal. S'il n'en admet pas, l'une de ces conditions ne peut pas être remplie. C'est un obstacle majeur à la construction de Laplaciens. En particulier, la méthodes des éléments finis et le calcul extérieur discret ne peuvent garantir la propriété d'extrema (à cause de leurs poids éventuellement négatifs).

Toutefois, contrairement à la plupart des autres approches, ils assurent la convergence du Laplacien, dans le sens où, si f_n tend vers f alors Δf_n tend vers Δf . Nous avons également un autre moyen de contourner l'obstruction : la condition de localité. En fait, dans la définition de la localité dans le cadre continu doit s'entendre au sens géodésique (la mesure de la localité se fait par rapport à la métrique de la surface), alors que l'obstruction donne une définition de la localité fondée sur la connectivité. En fait, la condition de localité devrait être considérée au sens de Voronoï : deux points ne devraient être considérés comme voisins que si leurs cellules de Voronoï (plongées dans la surface) sont adjacentes. Cela conduit à la définition d'une connectivité utilisée pour calculer le Laplacien différente de celle de la surface. Pour aller encore plus loin, nous pourrions imaginer discrétiser une fonction sur une surface d'une manière complètement indépendante du maillage de cette surface. Par exemple, on pourrait introduire des points de contrôles répartis de façon homogène sur la surface, et dont on construirait la triangulation de Delaunay. Cela permettrait d'installer une base de fonctions plus indépendante de la qualité de la discrétisation de la surface. Nous n'avons pas exploré cette idée plus profondément, mais dès que la triangulation est Delaunay, cela assure que Laplacien n'a pas de termes négatifs, et ce sans modifier le maillage de la surface.

4.4 Conclusion

Discrétiser le Laplacien sur un maillage est une question qui doit être traitée avec le plus grand soin, en fonction des propriétés que nous souhaitons pour le Laplacien discrétisé. La méthode des éléments finis et le calcul extérieur discret sont les approches les mieux justifiées le sens où elles garantissent la convergence sous certaines conditions. Même si nous travaillons sur un maillage fixé (que nous ne voulons ou pouvons pas modifier), la convergence est importante, car elle garantit que le Laplacien ainsi défini est indépendant du maillage. Nous entendons par là que si la résolution du maillage est variable, seule la résolution du résultat va changer, mais pas sa forme générale. C'est la raison pour laquelle nous utiliserons seulement des discrétisations convergentes au lieu de Laplaciens purement discrets. Les inconvénients de ces discrétisations (poids négatifs) ont été mentionnés dans la sous-section précédente, mais on a proposé des moyens pour les contourner.

L'existence d'un "bon" Laplacien est étroitement liée à la qualité du maillage le quel il est défini. Cela conduit à envisager que la définition d'un "bon" Laplacien (et d'une "bonne" base de fonctions pour le calculer) doit plus s'abstraire de la structure du maillage. C'est pourquoi la dernière partie de cette thèse porte sur les méthodes de calcul des paramétrisations quad qui peuvent servir d'espace de base pour définir de meilleures bases de fonctions, permettant de calculer un Laplacien avec de bonnes propriétés, quelle que soit la qualité du maillage.

Pour l'instant, nous allons laisser de côté ces considérations et nous concentrer sur les Laplaciens explicites dans cette section pour construire de nouvelles bases de fonctions qui ont de nombreuses propriétés intéressantes : les fonctions propres du Laplacien, que nous appellerons *harmoniques variétés*.

5

Harmoniques variétés

Sommaire

5.1 Previous Work	75
5.2 Spectral Analysis on Manifolds	76
5.2.1 Fourier Analysis	76
5.2.2 Intern approximation for Laplacian eigenfunctions	77
5.2.3 Manifold Harmonics Basis (MHB)	77
5.2.4 Interpretation of the MHB	78
5.3 The Manifold Harmonic Transform (MHT)	79
5.3.1 MHT and inverse MHT	79
5.3.2 Spectral filtering	80
5.4 Numerical Solution Mechanism	81
5.4.1 Band-by-band computation of the MHB	82
5.4.2 Limited-memory spectral filtering	83
5.5 Results	83
5.6 Conclusion	85
5.6.1 Discussion	85
5.6.2 Toward more localized basis functions	87

Introduction

Avec l'avènement des technologies d'acquisition d'objets en 3D, on dispose maintenant facilement de représentations informatiques d'objets réels. Cependant, l'acquisition peut présenter un certain bruit qui doit être filtré. De façon plus générale, on peut chercher à atténuer certains détails (c'est à dire certains niveaux de résolution) tout en exagérant d'autres. Dans son article fondateur [Taubin 1995], Taubin a montré que le formalisme du traitement du signal peut être appliquée avec succès à la géométrie numérique. Son approche est basée sur les similitudes entre les vecteurs propres du Laplacien de graphe et la base fonctions utilisée dans la transformée de Fourier discrète. Cette base de fonctions de Fourier permet à un signal donné d'être décomposé en une somme d'ondes sinusoïdales de fréquences de plus en plus élevées. Taubin utilise cette analogie comme un outil théorique pour concevoir et analyser des approximations de filtres passe-bas. Plusieurs variantes de cette approche ont été ensuite suggérées, comme on le verra ci-dessous.

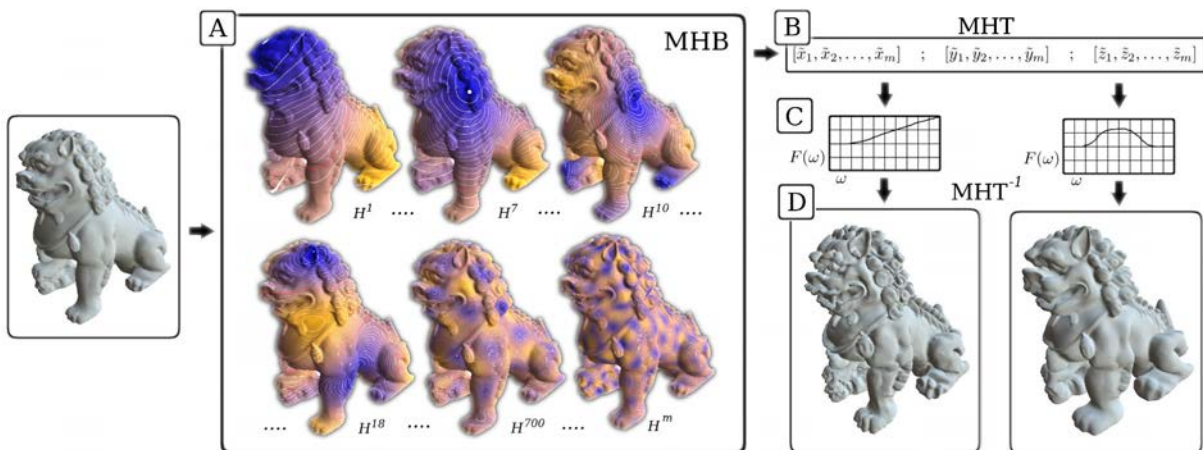


FIG. 5.1 – Etapes principales de notre méthode : A : Calculer la base d’harmoniques variétés (MHB pour *Manifold Harmonics Basis*) de la surface triangulée. B : Transformer la géométrie dans l’espace fréquentiel en calculant sa transformée spectrale (MHT pour *Manifold Harmonics Transform*). C : Appliquer un filtre sur la géométrie dans l’espace fréquentiel. D : Revenir dans l’espace géométrique en appliquant la MHT inverse.

Dans ce chapitre, nous présenterons les harmoniques variétés (*Manifold Harmonics*), qui sont définies comme étant les fonctions propres du Laplacien sur un maillage. Les concepts vus dans la première partie de cette thèse, ainsi que le chapitre précédent sur les Laplaciens serviront de fondements théoriques permettant de définir les multiples harmoniques. Nous présenterons ensuite le schéma numérique que nous avons utilisé pour calculer explicitement un grand nombre de fonctions de base. Ce schéma permet de repousser les limitations antérieures de plusieurs ordres de grandeur. La principale application des harmoniques variétés sur laquelle nous allons nous concentrer est la généralisation de la transformée de Fourier et de l’analyse spectrale sur une surface triangulée. Nous allons appliquer cette analyse spectrale généralisée au filtrage de la géométrie, de l’éclairage et des couleurs. Enfin, nous examinerons d’autres applications pour cette base.

Au lieu de n’utiliser l’analyse de Fourier que comme un outil théorique pour analyser des approximations de filtres, notre idée est de la généraliser pleinement sur les surfaces triangulées, et de l’utiliser pour construire un outil de filtrage interactif. Notre chaîne de traitement est similaire à celle décrite dans [Pauly and Gross 2001] pour les nuages de points. La principale différence est que nous ne nécessitons ni segmentation, ni ré-échantillonnage : la transformée de Fourier est calculée *directement* sur le maillage.

Plan du chapitre

Nous commencerons par un bref état de l’art des travaux antérieurs dans le domaine du filtrage des surfaces dans la Section 5.1. L’analyse de Fourier sera ensuite rappelée et étendue aux surfaces triangulées (Section 5.2). Cette extension nécessite de définir une base d’harmoniques variétés (MHB pour *Manifold Harmonics Basis*) dont nous donnerons une interprétation. Équipés de ce nouvel outil, il est alors facile de généraliser le filtrage spectral aux surfaces triangulées par l’intermédiaire d’une transformée spectrale (MHT pour *Manifold Harmonics Transform*) d’une MHT inverse. Ces transformées permettent de passer de l’espace géométrique à l’espace fréquentiel (Section 5.3). Nous allons finalement expliquer comment calculer efficacement la MHB en la pratique, et comment mettre en oeuvre un traitement spectral expli-

cite de la géométrie (Section 5.4). Enfin, nous montrerons des résultats de la méthode de filtrage dans la Section 5.5 avant de conclure dans la Section 5.6.

Les étapes principales du filtrage spectral sont présentées dans la Figure 5.1 :

- A : Étant donnée une surface triangulée avec n sommets, calculer une base de fonctions $H^k, k = 1 \dots m$ que l'on appelle *Manifold Harmonics Basis* (MHB). Le k -ième élément de la MHB est une fonction définie par ses valeurs h_i^k au sommet i ;
- B : une fois la MHB calculée, transformer la géométrie dans l'espace fréquentiel en calculant la *Manifold Harmonic Transform* (MHT) de la géométrie, c'est à dire les trois vecteurs de coefficients $[\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_m]$, $[\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_m]$, et $[\tilde{z}_1, \tilde{z}_2, \dots, \tilde{z}_m]$.
- C : appliquer un filtre $F(\omega)$ dans l'espace des fréquences en multipliant chaque $(\tilde{x}_k, \tilde{y}_k, \tilde{z}_k)$ par $F(\omega_k)$, où ω_k est la fréquence associée à H^k ;
- D : enfin, retransformer l'objet dans l'espace géométrique en appliquant la MHT inverse.

Contributions

- Nous formalisons le problème de trouver les fonctions propres du Laplacien et fournissons une interprétation de ces fonctions propres.
- Nous étendons explicitement les notions de transformée de Fourier et de filtrage spectral aux surfaces triangulées.
- Nous proposons une méthode numérique efficace pour calculer les vecteurs propres de la matrice du Laplacien symétrisée, qui repousse les limites actuelles (quelques milliers de sommets) de plusieurs ordres de grandeur (jusqu'à un million de sommets). La symétrie assure que ces fonctions propres forment une base orthonormale appelée (MHB) dont les valeurs propres sont réelles et positives.
- Une fois la MHB calculée sur une surface triangulée, le filtrage spectral sur celle-ci devient interactif. Nous le démontrons appliqué à la géométrie, à l'éclairage et aux couleurs.

Cette approche présente des points communs avec le traitement du signal avec des harmoniques sphériques et la transformée en cosinus discrets. La différence principale avec notre approche est que nous n'avons pas besoin de rééchantillonner notre surface initiale. Cependant, la MHB dépend de cette surface, et nécessite donc d'être précalculée (étape A). Une fois que la MHB est connue, les étapes suivantes peuvent être effectuées de façon très rapide. Cela permet à la solution d'être mise à jour de façon interactive lorsque l'utilisateur modifie le filtre $F(\omega)$.

Ce travail a été publié en anglais dans le journal "Computer graphics forum" sous le nom "Spectral geometry processing with Manifold Harmonics" [Vallet and Levy 2008], et a été présenté par l'auteur à la conférence Eurographics 2008. Nous en reproduisons ici une version allégée de sa partie théorique (qui a été reprise et étoffée dans les chapitres précédents).

5.1 Previous Work

The eigenfunctions of the Laplacian are known to define a Fourier-like function basis, that can be used to define spectral analysis on manifolds. We will present an overview of previous works related with spectral analysis and filtering.

Laplacian based Geometric " : A Laplacian-based geometric filtering method was recently proposed [Kim and Rossignac 2005] and was applied to a wide class of filters (e.g. band-exaggeration). This me-

thod combines explicit and implicit schemes to reach the different frequency bands involved in the filter. In comparison, our method can use arbitrary user-defined filters, and offers in addition the possibility of changing the filter interactively.

Geometric filtering through energy minimization : Other approaches to geometric filtering are based on energy minimization (e.g. [Mallet 1992]). These methods are called *discrete fairing* in [Kobbelt 1997, Kobbelt et al. 1998], in reference to their continuous-setting counterparts [Bloor and Wilson 1990]. Recently, a method was proposed [Nealen et al. 2006] to optimize both inner fairness (triangle shapes) and outer fairness (surface smoothness), by using a combination of the combinatorial Laplacian and the discrete Laplace-Beltrami operator.

Spectral transform : To directly implement the spectral transform on manifolds, several methods consist in putting the input surface in one-to-one correspondence with a simpler domain [Zhou et al. 2004], or to partition it into a set of simpler domains [Lee et al. 1998, Pauly and Gross 2001] on which it is easier to define a frequency space. Note that these methods generally need to resample the geometry, with the exception of Mousa *et al.* [Mousa et al. 2006] who directly compute the Spherical Harmonic Transform of a star-shaped mesh. It is also possible to extract the frequencies from a progressive mesh [Lee et al. 1998] and avoid resampling the geometry by using irregular subdivision [Guskov et al. 1999]. Finally, an original approach is presented in [Sorkine et al. 2005] that computes "geometry aware" basis functions, defined as solutions of some least-squares problems. Our method computes the frequency-space basis functions as eigenvectors of a symmetric matrix, for a surface of arbitrary topology without needing any resampling nor segmentation.

5.2 Spectral Analysis on Manifolds

Manifold harmonics (also called *shape harmonics*) are defined as the eigenfunctions of the Laplace operator. This section starts by recalling the familiar Fourier analysis to justify the choice of the eigenfunctions of the Laplace operator to generalize this setting to arbitrary manifolds. We will then explain the min-max principle which provides a nice interpretation of the MHB.

5.2.1 Fourier Analysis

As in Taubin's article [1995], we start by studying the case of a closed curve, but staying in the continuous setting. Given a square-integrable periodic function $f : x \in [0, 1] \mapsto f(x)$, or a function f defined on a closed curve parameterized by normalized arclength, it is well known that f can be expanded into an infinite series of sines and cosines of increasing frequencies :

$$f(x) = \sum_{k=1}^n \tilde{f}_k H^k(x) \quad ; \quad \begin{cases} H^{2k} & = \sin(kx) \\ H^{2k+1} & = \cos(kx) \end{cases} \quad (5.1)$$

where the coefficients \tilde{f}_k of the decomposition are given by :

$$\tilde{f}_k = \langle f, H^k \rangle = \int_0^1 f(x) H^k(x) dx \quad (5.2)$$

and where $\langle \cdot, \cdot \rangle$ denotes the inner product (i.e. the “dot product” for functions defined on $[0, 1]$ that enables projecting onto a function basis) The "Circle harmonics" basis H^k is orthonormal with respect to $\langle \cdot, \cdot \rangle$: $\langle H^k, H^k \rangle = 1$, $\langle H^k, H^l \rangle = 0$ if $k \neq l$.

The set of coefficients \tilde{f}_k (Equation (5.2)) is called the Fourier Transform (FT) of the function f . Given the coefficients \tilde{f}_k , the function f can be reconstructed by applying the inverse Fourier Transform FT^{-1} (Equation (5.1)). Our goal is now to generalize these notions to arbitrary manifolds. To do so, we can consider the functions H^k of the Fourier basis as the eigenfunctions of $-\partial^2/\partial x^2$: the eigenfunctions H^{2k} (resp. H^{2k+1}) are associated with the eigenvalues k^2 :

$$-\frac{\partial^2 H^{2k}(x)}{\partial x^2} = k^2 \sin(kx) = k^2 H^{2k}(x)$$

To extend this construction to arbitrary manifolds, we need a generalization for both inner product $\langle \cdot, \cdot \rangle$ and second-order derivative $\partial^2/\partial x^2$. The issue of generalizing the inner product on a mesh in both FEM and DEC settings was tackled in the first part of this thesis. The generalization of the second order derivative is the Laplacian, and its generalization to meshes was studied in the previous chapter.

Based on these solid theoretical foundations, we can now generalize the Fourier basis on a manifold by finding the eigenfunctions of discrete Laplacians.

5.2.2 Intern approximation for Laplacian eigenfunctions

Let $V(\mathcal{S})$ be a Hilbert space on a surface \mathcal{S} with dot product $\langle \cdot, \cdot \rangle$. The eigenfunctions and eigenvectors of the Laplacian Δ on \mathcal{S} are the non-trivial solutions of the problem :

$$\text{Find } f \in V(\mathcal{S}), \lambda \in \mathbb{R}^* \text{ such that } -\Delta f = \lambda f \quad (5.3)$$

where λ is real, and $f \in V(\mathcal{S})$. Problem (5.3) has the variational formulation :

$$\text{Find } f \in V(\mathcal{S}), \lambda \text{ s.t. } \langle -\Delta f, g \rangle = \langle \nabla f, \nabla g \rangle = \lambda \langle f, g \rangle \forall g \in V(\mathcal{S}) \quad (5.4)$$

Let V_n be a finite dimensional subset of V of dimension n . The *intern approximation* of problem (5.4) relative to V_n is the problem :

$$\text{Find } f \in V_n(\mathcal{S}), \lambda \text{ s.t. } \langle \nabla f, \nabla g \rangle = \lambda \langle f, g \rangle \quad \forall g \in V_n(\mathcal{S}) \quad (5.5)$$

Once again, this is a double approximation :

1. We restrict the solution we look for (f) to be in V_n
2. We restrict the test functions (g) to be in V_n

and the Lax Milgram theory exposed in Chapitre 2 ensures the convergence of this approximation under the appropriate conditions. We will now apply the intern approximation to a mesh to build the Manifold Harmonics Basis (MHB).

5.2.3 Manifold Harmonics Basis (MHB)

Let $M = \langle \mathcal{V}, \mathcal{E}, \mathcal{F} \rangle$ be a connected orientable mesh, and $\{\Phi^i\}_{i=1..n}$ a function basis on M such as the interpolating function basis that we defined in Part I. If we call $\Phi = (\Phi^1, \dots, \Phi^n)$ Any function $f \in V_n(M) = \text{span}(\Phi^i)$ writes $f = \Phi \mathbf{f} = \sum_i \Phi^i f_i$, and (5.5) rewrites :

$$\forall j, \quad \langle \nabla \Phi \mathbf{f}, \nabla \Phi^j \rangle = \lambda \langle \Phi \mathbf{f}, \Phi^j \rangle \quad (5.6)$$

or in matrix notation :

$$\forall j, \quad K_{\Phi} \mathbf{f} = \lambda M_{\Phi} \mathbf{f} \quad (5.7)$$

where K_{Φ} and M_{Φ} are the stiffness and mass matrix defined in the previous chapter. In the FEM setting, we do a lumped mass approximation on M_{Φ} by replacing it with a diagonal matrix \tilde{M}_{Φ} as explained in the previous chapter. In the DEC setting, we get the same matrices as in P_1 FEM, except that the lumped mass matrix \tilde{M}_{Φ} is replaced with \star_0 . By normalizing the basis (see Section 4.3.3), we find that the eigenfunctions H^j of the Laplacian relative to Φ are defined by the eigenvectors of the symmetric normalized Laplacian $\Delta_{\Phi} = \tilde{M}_{\Phi}^{-1/2} K_{\Phi} \tilde{M}_{\Phi}^{-1/2}$:

$$\Delta_{\Phi} \bar{\mathbf{h}}^k = \lambda_k \bar{\mathbf{h}}^k$$

Given that Δ_{Φ} is symmetric, we can compute its normalized eigenvectors $\{\bar{\mathbf{h}}^k\}$ which form an orthonormal basis of \mathbb{R}^n . We call Manifold Harmonics basis (MHB) the corresponding basis of functions $\{H^k = \Phi \bar{\mathbf{h}}^k = \Phi \tilde{M}_{\Phi}^{-1/2} \bar{\mathbf{h}}^k = \Phi \mathbf{h}^k\}$. An example is shown in Figure 5.2. The MHB is orthonormal with respect to the inner product by construction. This property is important to define the associated transforms, as explained below.

To summarize, the computation of the MHB can be done in the following way :

1. Assemble the symmetrized discrete Laplacian $\Delta_{\Phi} = \tilde{M}_{\Phi}^{-1/2} K_{\Phi} \tilde{M}_{\Phi}^{-1/2}$;
2. Compute its eigenvectors ($\bar{\mathbf{h}}^k$) (see Section 5.4) ;
3. Map them into the canonical basis : (\mathbf{h}^k) = ($\tilde{M}_{\Phi}^{-1/2} \bar{\mathbf{h}}^k$), such that the Manifold Harmonics Basis (MHB) is the basis of function :

$$H^k = \Phi \mathbf{h}^k = \Phi \bar{\mathbf{h}}^k$$

Finally, to gain insight on what these basis functions are, we are going to present the min-max principle, allowing for an intuitive interpretation of these function basis.

5.2.4 Interpretation of the MHB

Consider the problem of finding the smoothest unit function on \mathcal{S} , i.e. the function minimizing the Dirichlet energy $E(f) = \int_{\mathcal{S}} \|\nabla f\|^2 = \langle \nabla f, \nabla f \rangle$ under a normalization constraint $\|f\|_{L_2}^2 = \int_{\mathcal{S}} \|f\|^2 = \langle f, f \rangle = 1$. In both FEM and DEC formalism, this problem has Lagrangian :

$$\mathcal{L}(\mathbf{f}, \lambda) = \mathbf{f}^t K \mathbf{f} - \lambda (\mathbf{f}^t M \mathbf{f} - 1) \quad (5.8)$$

which fixed points are given by :

$$\frac{\partial \mathcal{L}}{\partial \mathbf{f}} = K \mathbf{f} - \lambda M \mathbf{f} = 0 \quad \frac{\partial \mathcal{L}}{\partial \lambda} = \mathbf{f}^t M \mathbf{f} - 1 = 0$$

The first equation is exactly (5.7), and the second is simply a normalization condition, which means that the fixed points of the Lagrangian are precisely our manifold harmonics (normalized eigenfunctions of the Laplacian) and that the eigenvalues are the corresponding Lagrange multipliers at the fixed points. As these solutions satisfy $Kf = \lambda Mf$, we have $f^t K f = \lambda f^t M f = \lambda$, $E(f)$ is minimum for the smallest eigenvalues. This means that if we sort the basis functions by increasing eigenvalue, each basis function minimizes $E(f)$ (achieves maximum smoothness) in the subspace orthogonal to the span of previous ones. For instance, the first basis function is the smoothest unit function, which is the obvious constant solution $f_1^0 = \dots = f_n^0 = 1/\sqrt{\sum_{i,j} M_i^j}$, $\lambda^0 = 0$. Then the second basis function is the smoothest unit



FIG. 5.2 – Some functions of the Manifold Harmonic Basis (MHB). Note the similarity with the sine products used by the DCT.

function in the subspace orthogonal to constants (subset of functions of null mean), which is usually called the Fiddler function, and so on.

If these functions are used to approximate a given function, this means that the smoothest components of the function will be recovered first, and as we get to higher eigenvalues more details will be captured. This justifies to see each eigenfunction as a certain resolution level of the function.

We will now then explain how the MHB allows to define a Manifold Harmonic Transform (MHT) and inverse MHT.

5.3 The Manifold Harmonic Transform (MHT)

Transforming 0-forms between the canonical basis (Φ^i) *i.e.*, *geometric space* and the MHB (H^k) *i.e.*, *frequency space* will be called the Manifold Harmonic Transform (MHT). It is also easy to define the inverse transform (MHT⁻¹). They generalize the notion of Fourier transform to simplicial complexes. We now derive the expressions of MHT and MHT⁻¹, and apply them to spectral geometric filtering.

5.3.1 MHT and inverse MHT

We consider the geometry x (resp. y, z) of the simplicial complex S , that can be seen as a piecewise linear function. In other words, x is a linear combination of the P_1 basis functions Φ^i :

$$x = \sum_{i=1}^n x_i \Phi^i \quad (5.9)$$

where x_i denotes the x coordinate at vertex i . Computing the MHT means projecting the function x into the MHB, *i.e.* finding the coefficients \tilde{x}_k such that :

$$x = \sum_{k=1}^m \tilde{x}_k H^k \quad (5.10)$$

Since the MHB is orthonormal, we can easily project (5.9) and (5.10) onto each H^k with the inner product :

$$\begin{aligned} \langle x, H^k \rangle &= \sum_{i=1}^n x_i \langle \Phi^i, H^k \rangle = \mathbf{x}^T \tilde{\mathbf{M}} H^k \\ \langle x, H^k \rangle &= \sum_{k=1}^m \tilde{x}_k \langle H^k, H^k \rangle = \tilde{x}_k \end{aligned}$$

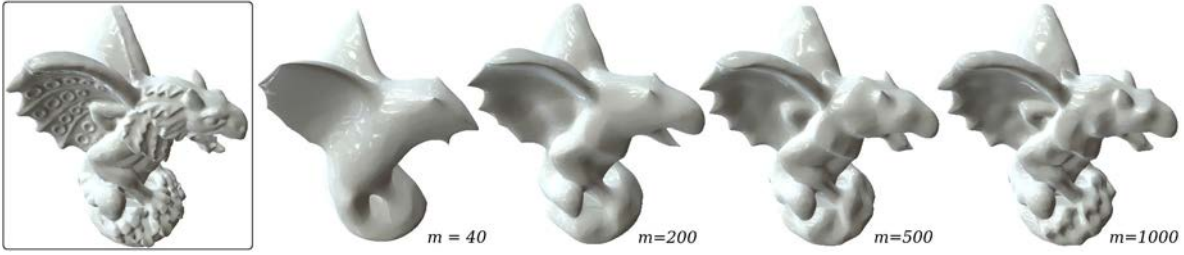


FIG. 5.3 – Reconstructions obtained with an increasing number of MH functions.

By comparing both expressions, we finally obtain the expression of \tilde{x}_k :

$$\tilde{x}_k = x^T \tilde{M} H^k = \sum_{i=1}^n x_i \tilde{M}_i^k H_i^k \quad (5.11)$$

Using these equations, we know how to transform 0-forms between geometric space (x) and frequency space (\tilde{x}). This defines the MHT and inverse MHT :

- MHT : $(x_i) \rightarrow (\tilde{x}_k)$ (Equation 5.11)
- MHT⁻¹ : $(\tilde{x}_k) \rightarrow (x_i)$ (Equation 5.10)

Figure 5.3 shows the geometry reconstructed from the MHT of a surface using a different number m of MHT coefficients. As can be seen, the first H^k functions capture the general shape of the functions and the next ones correspond to the details.

5.3.2 Spectral filtering

Once the geometry is converted into the MHB, each component $(\tilde{x}_k, \tilde{y}_k, \tilde{z}_k)$ of the MHT corresponds to an individual spatial frequency ω_k . In the case of a closed curve (Section 5.2.1), we have $-\partial^2 \sin(\omega x) / \partial x^2 = \omega^2 \sin(\omega x)$. Therefore, the spatial frequency ω_k corresponds to the square root of the eigenvalue λ_k . In the case of a manifold, similar results can be obtained, that connect $\omega_k = \sqrt{\lambda_k}$ with the length of the nodal sets [Donnelly and Fefferman 1988].

A frequency-space filter is a function $F(\omega)$ that gives the amplification to apply to each spatial frequency ω . Since all frequencies are separated by the MHT, applying a filter $F(\omega)$ to the geometry becomes a simple product in frequency space. The filtered coordinate x_i^F (resp y_i^F, z_i^F) at vertex i is then given by :

$$x_i^F = \sum_{k=1}^m F(\omega_k) \tilde{x}_k H_i^k = \sum_{k=1}^m F(\sqrt{\lambda_k}) \tilde{x}_k H_i^k$$

In practice, we stop computing the MHB at a given ‘‘cutoff’’ frequency $\omega_m = \sqrt{\lambda_m}$ (inverse of 10 times the average edge length in our experiments). Therefore, smaller geometric details are not represented in the MHT. However, it is possible to keep track of them, by storing in each vertex the difference x_i^{hf} (resp y_i^{hf}, z_i^{hf}) between the original geometry and its projection onto the MHB :

$$x_i^{hf} = \sum_{k=m+1}^{\infty} \tilde{x}_k H_i^k = x_i - \sum_{k=1}^m \tilde{x}_k H_i^k$$

The frequency-space filter can be applied to the high-frequency components of the signal, by re-injecting them into the inverse MHT, as follows :

$$x_i^F = \sum_{k=1}^m F(\omega_k) \tilde{x}_k H_i^k + f^{hf} x_i^{hf} \quad (5.12)$$

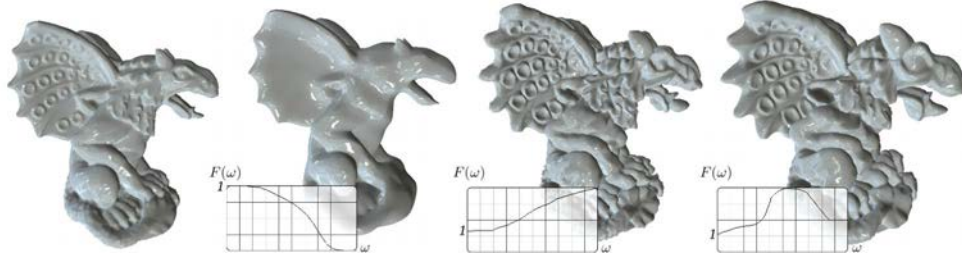


FIG. 5.4 – Low-pass, enhancement and band-exaggeration filters. The filter can be changed by the user, the surface is updated interactively.

In this equation, the term f^{hf} denotes the average value of the filter F on $[\omega_m, \omega_M]$, where ω_M denotes the maximum (Nyquist) frequency of the mesh (half the inverted edge length). The high-frequency component behaves like a wave packet that can be filtered as a whole, but that cannot be considered as independent frequencies.

Figure 5.4 demonstrates low-pass, enhancement and band-exaggeration filters. Note that unlike existing methods, ours does not need to re-scale the model. Moreover, only the filtered inverse MHT (Equation 5.12) depends on the filter F . As a consequence, after storing the MHB and the MHT coefficients, the solution can be updated interactively when the user changes the filter F .

We now proceed to explain how to compute the coefficients H_i^k of the MHB and the associated eigenvalues λ_k .

5.4 Numerical Solution Mechanism

Computing the MHB means solving for the eigenvalues λ_k and eigenvectors \bar{H}^k for the symmetric positive semi-definite matrix $\Delta_{\bar{\Phi}}$:

$$\Delta_{\bar{\Phi}} \bar{\mathbf{h}}^k = \lambda_k \bar{\mathbf{h}}^k \quad (5.13)$$

However, eigenvalues and eigenvectors computations are known to be extremely computationally intensive. To reduce the computation time, Karni *et al.* [2000] partition the mesh into smaller charts, and [Dong *et al.* 2006] use multiresolution techniques. In our case, we need to compute multiple eigenvectors (typically a few thousands). This is known to be currently impossible for meshes with more than a few thousand vertices [Wu and Kobbelt 2005]. In this section, we show how this limit can be overcome by several orders of magnitude.

To compute the solutions of large sparse eigenproblems, several iterative algorithms exist. The publicly available library ARPACK (used in [Dong *et al.* 2006]) provides an efficient implementation of the Arnoldi method. Yet, two characteristics of eigenproblem solvers hinder us from using them directly to compute the MHB for surfaces with more than a few thousand vertices :

- first of all, we are interested in the lower frequencies, i.e. eigenvectors with associated eigenvalues lower than ω_m^2 . Unfortunately, iterative solvers perform much better for the other end of the spectrum. This contradicts intuition as in mechanical simulations for instance, it is difficult to ensure the stability of numerical schemes in high frequencies. However, it can be explained in terms of filtering as lower

frequencies correspond to higher powers of the smoothing kernel, which may have a poor condition number ;

- second, we need to compute a large number of eigenvectors (typically a thousand), and it is well known that computation time is superlinear in the number of requested eigenvectors. In addition, if the surface is large (a million vertices), the MHB does not fit in system RAM.

5.4.1 Band-by-band computation of the MHB

We address both issues by applying spectral transforms to the eigenproblem. To get the eigenvectors of a spectral band centered around a value λ_S , we start by shifting the spectrum by λ_S , by replacing $\Delta_{\bar{\phi}}$ with $\Delta_S = \Delta_{\bar{\phi}} - \lambda_S Id$. Then, we can swap the spectrum by inverting Δ_S . This is called the *Shift-Invert* spectral transform, and the new eigenproblem to solve is given by :

$$\Delta_S^{-1} \bar{\mathbf{h}}^k = \mu_k \bar{\mathbf{h}}^k \quad (5.14)$$

The fact that Δ_S may be singular is in practice not a problem as the iterative solver we use to solve (5.14) does not ask for Δ_S^{-1} explicitly. Instead, it outputs some vectors \mathbf{v} and asks in return for the result of the multiplication $\mathbf{x} = \Delta_S^{-1} \mathbf{v}$. Hence we only need to compute a sparse *indefinite* Cholesky factorization of Δ_S using sparse direct solvers (TAUCS, SUPERLU). Thanks to this factorization, it is very fast to solve $\Delta_S \mathbf{x} = \mathbf{v}$ by simple back-substitution each time the iterative solver asks for a multiplication with Δ_S^{-1} . For large models (a million vertices), we used the sparse OOC (out-of-core) symmetric indefinite factorization [Meshar et al. 2006] implemented in the future release of TAUCS, kindly provided by S. Toledo.

It is easy to check that (5.13) and (5.14) yield the same eigenvectors, and that the eigenvalues are related by $\lambda_k = \lambda_S + 1/\mu_k$. Our iterative solver returns the high end of the spectrum (largest μ 's), corresponding to a band of eigenvalues of $\Delta_{\bar{\phi}}$ centered around λ_S . It is then possible to split the MHB computation into multiple bands, and obtain a computation time that is linear in the number of computed eigenpairs. Based on the considerations above, we derive the following "band-by-band" algorithm :

- (1) $\lambda_S \leftarrow 0$; $\lambda_{last} \leftarrow 0$
- (2) **while**($\lambda_{last} < \omega_m^2$)
- (3) compute the Cholesky factorization of $\Delta_S = \Delta_{\bar{\phi}} - \lambda_S Id$
- (4) find the 50 first eigenpairs ($\bar{\mathbf{h}}^k, \mu_k$) of Δ_S^{-1}
- (5) **for** $k = 1$ **to** 50
- (6) $\lambda_k \leftarrow \lambda_S + 1/\mu_k$
- (7) **if** ($\lambda_k > \lambda_{last}$) **write**($\bar{\mathbf{h}}^k, \lambda_k$)
- (8) **end** //for
- (9) $\lambda_S \leftarrow \max(\lambda_k) + 0.4(\max(\lambda_k) - \min(\lambda_k))$
- (10) $\lambda_{last} \leftarrow \max(\lambda_k)$
- (11) **end** //while
- (12) sort the ($\bar{\mathbf{h}}^k, \lambda_k$) by increasing λ_k

If the MHB does not fit in RAM, the new eigenpairs of each band of can be stream-written into a file (Line 7). Since the eigenvalues are centered around the shift λ_S , the shift for the next band is given by the last computed eigenvalue plus slightly less than half the bandwidth to ensure that the bands overlap and that we are not missing any eigenvalue (Line 9). If the bands do not overlap, we recompute a larger band until they do. Note that this is different from the shift-invert spectral transform implemented by

ARPACK, dedicated to iterative solvers. Ours makes use of the factorization of the matrix, resulting in much better performances.

To speed-up computation, since the non-zero pattern of the matrix does not change, we reuse the same permutation for all the bands. In addition, the Arnoldi iteration is initialized with the vector obtained from the previous frequency band.

5.4.2 Limited-memory spectral filtering

After computing the MHB (Section 5.4.1) the filtering algorithm (Section 5.3.2) allows for interactive spectral geometry processing. However, for large meshes, storing the MHB may be prohibitive. Fortunately, it is possible to compute the MHB and the filter *all together* (at the cost of losing interactivity). In addition, computations can be reordered in a way that requires storing only one eigenvector at a time :

- (1) $x^F \leftarrow x$
- (2) **for each** eigenpair(H^k, ω_k)
- (3) $x^F \leftarrow x^F + (F(\omega_k) - 1) \langle x, H^k \rangle H^k$
(where $\langle x, H^k \rangle = \tilde{x}_k = x^T \star_0 H^k$)
- (4) **end for**

In a certain sense, this algorithm *replaces* a given frequency in the mesh by its filtered version. Once again, this is made possible by the *orthonormality* of the MHB. In practice, the combined MHB+filtering algorithm is obtained by replacing line (7) of the MHB computation algorithm with line (3) of the algorithm above (instead of streaming the eigenpair to a file, we replace the corresponding frequency in the mesh). Interestingly, since our limited-memory algorithm changes a specific frequency *in-place* (i.e. without storing the MHB), it is possible to limit the computation of the MHB to the frequencies changed by the filter, by using the shift-invert spectral transform.

For smaller meshes (hundreds thousands vertices), a faster in-core sparse factorization can be used. Note that before the next release of TAUCS is available, the reader who wants to reproduce our results may use SuperLU instead (at the expense of losing scalability).

5.5 Results

We have experimented the OOC factorization combined with the streamed band-by-band eigenvectors algorithm for computing up to a thousand eigenvectors on a mesh with one million vertices. We have also implemented an OOC version of the MHT, filtering and inverse MHT, that reads one frequency band at a time and accumulates its contribution.

We have experimented our spectral filtering method with objects of different sizes. The timings are reported in table 5.1. The last row (limited-memory version) uses a band-exaggeration filter that spans 1/4th of the object's spectrum. As can be seen, since we only compute the concerned eigenpairs, this results in better performances (as compared to summing $MHB + MHT + MHT^{-1}$ rows). For filters that involve low frequencies, performances are better than Geofilter [Kim and Rossignac 2005], since this latter method needs to compute the inverse of a power of the Laplacian matrix, that is not sparse anymore for lower ends of the spectrum. For large models, e.g. the 1M vertices dragon (Figure 5.5), performance is still slow (1h). However, note that no-other existing method would allow to reach and act on the lower-end of the spectrum (that corresponds to large geometric details). Our MH-based filtering can be applied to objects of

	dino Fig. 2	drago Fig. 1	drago1*	drago2**	drago3** Fig. 6
n	56K	150K	244K	500K	1M
m	447	315	667	800	1331
MHB	77 s	160 s	9 m	2 h 21 m	6 h
MHT	0.34 s	0.65 s	18 s	32 s	76 s
MHT ⁻¹	0.53 s	1.02 s	4 s	48 s	85 s
LM-filt	18 s	41 s	135 s	28 m	1h

TAB. 5.1 – Timings for the different phases of the algorithm. For each data set, we give the number of vertices n , the number of computed eigenfunctions m , and the timings for the MHB, MHT and inverse MHT with filtering (Intel T7600 2.33 GHz). (*) using OOC MHT (**) using both OOC factorization and OOC MHT. The limited-memory (LM-filt) example uses a band-exaggeration filter that spans 1/4th of the object’s spectrum.

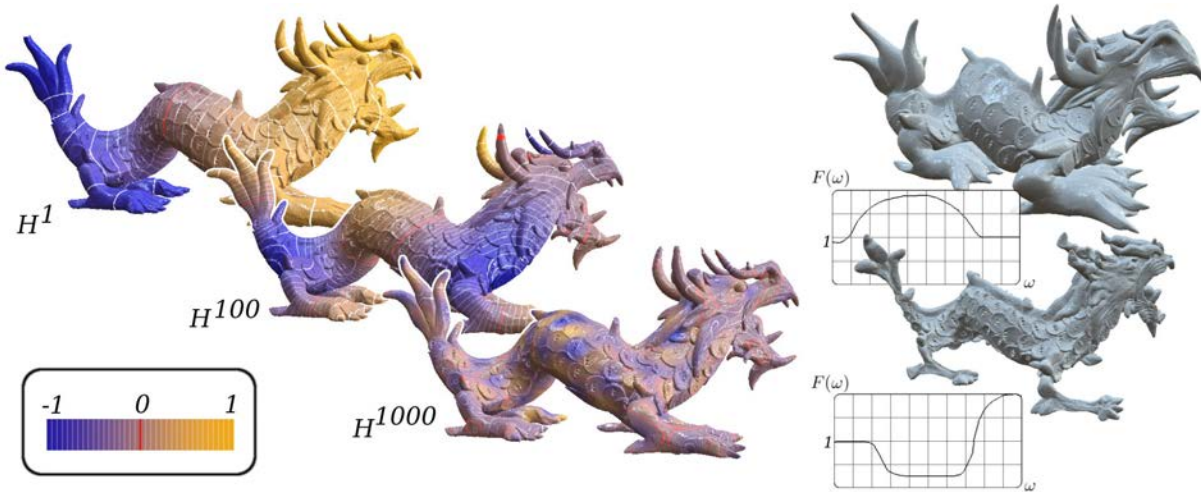


FIG. 5.5 – The MHB computed on 1M vertices (XYZ dragon) and limited-memory spectral filtering (that does not store the MHB).

arbitrary topology. Figure 5.6 shows a low-pass filter used to remove high-frequency noise from a sphere and from a genus 4 object. The low-pass filter nearly preserves the symmetry of the sphere. Figure 5.7 shows how our method implements an interactive version of *geofilter* [Kim and Rossignac 2005]. In addition, since our filter explicitly implements the spectral transform, it does not introduce the shrinking effect usually encountered with existing ones.

We demonstrate the versatility of our method, by applying it to various attributes attached to the vertices of surfaces. Figure 5.8 demonstrates our method applied to colors attached to the vertices of the mesh (enhancement and low-pass filters). Figure 5.9 shows how our framework applied to the normal vector can simulate various lighting effects. Applying a low-pass filter to the normal vector is approximately equivalent to filtering light intensity. This yields a very simple approximation of subsurface scattering, that the user can easily tune by adjusting the filter. Once the user is pleased with the result, only an additional normal vector is required to display the effect. The effect is simply obtained by replacing the normal vector with the filtered vector in the shader. Conversely, applying an enhancement filter to the normal vector yields a result very similar to the *exaggerated shading* method [Rusinkiewicz et al. 2006].

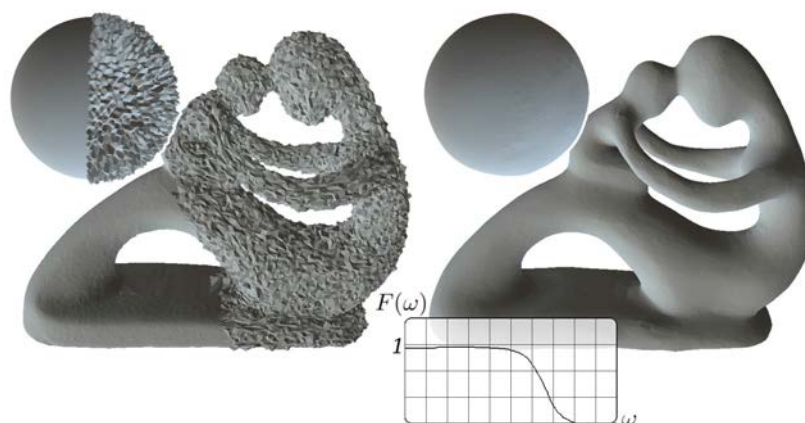


FIG. 5.6 – **Left** : a sphere and a genus-4 model with random noise added. **Right** : the low-pass filtered result.

Note that the user can interactively generate any intermediate shading style between these two extremes.

5.6 Conclusion

5.6.1 Discussion

In this chapter, we have presented a method for filtering functions defined on manifolds. We have explained how to recover the symmetry of discrete Laplace operators and the orthogonality of its eigenvectors. We used this theoretical framework to define the Manifold harmonic transform and inverse. The resulting filter is interactive, and does not encounter the shrinking effect obtained with classical schemes. On the practical side, we have overcome the current size limits of spectral geometry processing by several order of magnitudes, by making it usable for meshes with up to $10^5 - 10^6$ vertices. With our limited-memory spectral filtering algorithm, storage space is no-longer a limit. However, processing time for the MHB start to be expensive (hours) beyond 10^6 vertices. This will be optimized in future works, by introducing multiresolution in our solution mechanism.

Our implementation of MH-based geometry filtering computes the MHT of the x , y and z coordinates,

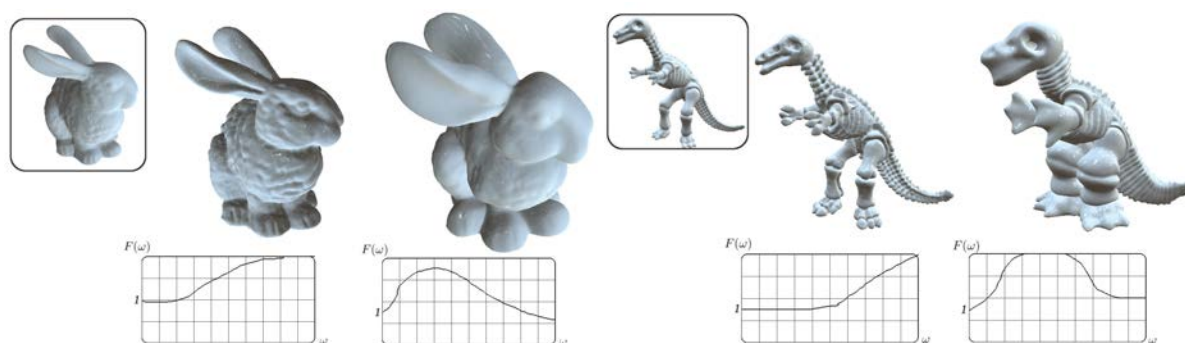


FIG. 5.7 – Filtering Stanford's bunny and Cyberware's dinosaur. Results similar to geofilter are obtained, with the addition of interactivity, and without any shrinking effect.



FIG. 5.8 – Filtering the colors attached to the vertices of an object of arbitrary topology.

that are dependent on the global orientation of the object. Therefore, we think that better results may be obtained by computing the MHT of some invariant local differential coordinates instead of using the absolute (x, y, z) coordinates, or by introducing the metric in the loop.

Another limitation of our method concerns objects with creases. It is well known that low-pass filters based on Fourier-like methods cannot preserve the creases (see Figure 5.10). Using the eigenfunctions of an anisotropic version of the Laplace operator may improve the frequency localization of the creases and therefore better preserve them when filtering. Our method and associated numerical solution mechanism may find applications in various contexts, e.g. segmentation, mesh watermarking or reconstruction.

Also note that filtering the geometry defined as a function on the surface itself is somehow misleading, as it is a function from the original surface to another embedding of the surface. Thus, the Laplacian is computed on the original surface independently of the deformation of the mesh through filtering. This can lead to counterintuitive results (see Figure 5.11), where an intensive smoothing leads to very elongated spikes.

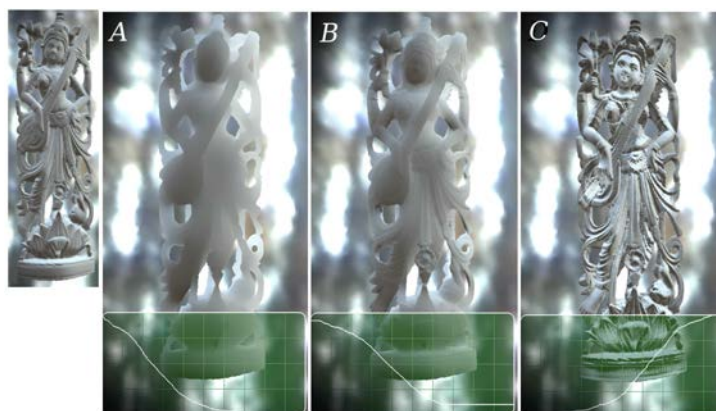


FIG. 5.9 – Signal processing approach to shading design. A : high scattering ; B : moderate scattering ; C : exaggerated shading

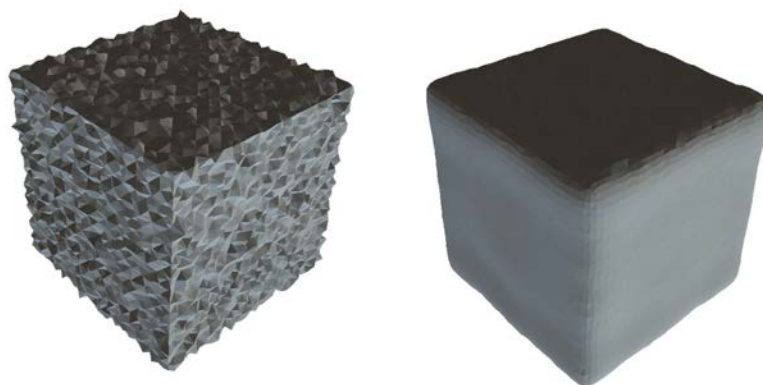


FIG. 5.10 – Sharp creases yield harmonics of many different frequencies, and are therefore difficult to preserve when filtering.



FIG. 5.11 – Very intense smoothing (right) of a dinosaur model (left) generates counterintuitive spikes.

Even though our applications use P_1 elements, we have implemented the MHB computation with P_2 and P_3 elements based on the coefficients of Section 4.3.4. As expected, the solution is much more regular in P_2 than in P_1 , and in particular more independent from the mesh. The difference between P_2 and P_3 is less visible as it regularizes second order derivative. This means that our applications can be enhanced with these higher order elements in a rather straightforward manner. In particular, the geometric filtering can be performed in P_n elements. However, the result of the filtering will no longer be a triangulated mesh as the resulting triangles can be curved. Also notice that as predicted in Section 2.2.1, P_2 elements gives a visually better and smoother solution than P_1 on a subdivided mesh, for the same number of degrees of freedom.

5.6.2 Toward more localized basis functions

Our method cannot be directly applied to mesh compression as we took particular care in making our Laplacian geometry dependent. In this case, our solver applied to the uniform Laplacian can be also used to implement an efficient version of Karni *et. al.*'s Spectral Mesh Compression [2000]. Moreover, it

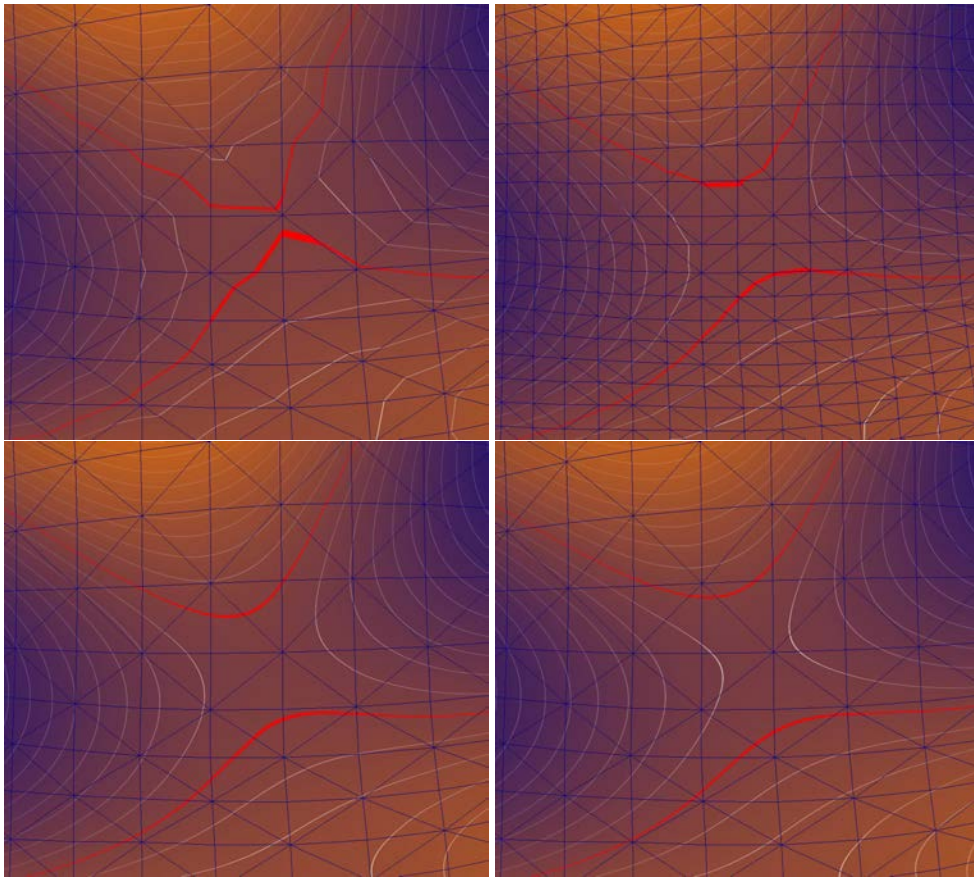


FIG. 5.12 – Comparison of a Manifold Harmonic computed based on P_1 elements (top left), P_1 on a twice denser mesh (top right), P_2 (bottom left) and P_3 (bottom right).

turns out that because the MHB is not spatially localized, many MHT coefficients (several thousands) are required to accurately reconstruct the geometry, even with a geometry aware Laplacian (see Figure 5.3). Besides Karni's initial concern of reducing computation time, we think that partitioning also partially fixes the problem of spatial localization (this is also why JPEG uses small blocks instead of applying the DCT to the whole image) at the expense of losing continuity. This leads to forecast that defining *Manifold Wavelets* localized in both frequency and spatial domains [Grinspun et al. 2002] will be an exciting research avenue in the future. Contrary to rectangular images that can be trivially cut into squares, the problem of finding a base domain for such functions on surfaces is complex and subtle. This is why the next part of this thesis tackles the problem of quad parameterization in order to provide such base domains, and focuses in particular on direction fields which are often necessary to define such quad parameterizations.

Troisième partie

Champs de directions

Introduction

De nombreux algorithmes en informatique graphique et en traitement de la géométrie utilisent des champs de directions (vecteurs tangents unitaires) lisses définis sur des surfaces triangulées. Par exemple, ces champs de directions peuvent être utilisés :

- En synthèse de texture (Figure 5.13 gauche)³ : on a besoin de définir une direction commune entre la texture (2D) et la surface à texturer, ou en d'autres termes on a besoin d'orienter la texture sur la surface.
- En rendu non-photoréaliste (Figure 5.13 droite) : si on veut représenter un objet 3D comme si il avait été dessiné par un artiste par exemple, il faut définir une direction pour les traits de pinceau ou d'encre.
- En remaillage (Figure 5.14) : pour remailler une surface, une des approches consiste à définir une direction pour les arêtes du remaillage, puis de générer le maillage lui-même.

Dans ces applications, seule la direction est importante. Dans le cas du remaillage, la norme pourrait être utilisée pour définir la taille des cellules du remaillage, mais il est en général plus judicieux de séparer le choix de la direction du choix de la taille. De plus, dans certain cas, on peut vouloir que la direction soit définie avec une symétrie de rotation. En particulier, le rendu non photoréaliste ne s'intéressera pas au sens des traits, et donc on a une symétrie par rotation d'angle π , et le remaillage quad s'intéressera à des symétries par rotation d'angle $\pi/2$. De tels objets sont différents des champs de vecteurs en ce qu'ils ont une topologie plus liée à celle de la surface sur laquelle ils sont définis. En particulier, leurs singularités ne peuvent plus être définies comme des zéros (puisque leur norme est unitaire), mais comme des "trous" dans leur domaine de définition.

Malgré les nombreuses avancées récentes dans le domaine de la géométrie numérique, un formalisme mathématique unifié fait toujours défaut pour étudier ces champs de directions généralisés. En conséquence, les algorithmes existants de manipulation de champs de directions sont en général limités à des procédures non optimales de relaxation locale. Dans cette partie de la thèse, nous allons commencer par présenter au Chapitre 6 les enjeux liés à la manipulation de champs de vecteurs sur les surface. Nous nous intéresserons aux applications de ces champs, aux caractéristiques qu'un outil de manipulation de champs doit avoir pour ces applications, et les travaux déjà réalisés dans ce domaine. On introduira ensuite au Chapitre 7 une théorie (continue) complète des champs de directions à N symétries proposant un formalisme unifié pour les champs de directions (\rightarrow , $N = 1$), de lignes ($-$, $N = 2$), et de croix ($+$, $N = 4$). Le Chapitre 7 proposera aussi une discrétisation permettant d'avoir accès aux concepts de la théorie continue.

Les résultats théoriques du Chapitre 7 et la structure proposée pour représenter les champs de directions à symétries nous permettrons par la suite de proposer deux algorithmes efficaces de manipulation ces champs de directions :

- Manipulation avec contrôle topologique *fort* : le système génère un champ de directions lisse sous un ensemble de contraintes sur les singularités (nombre, type et placement), ainsi que des contraintes sur les directions. Cet algorithme sera décrit au Chapitre 9.
- Manipulation avec contrôle topologique *faible* : le système génère un champ de directions lisse qui interpole un ensemble de directions données par l'utilisateur et empêche l'émergence de singularités dues aux détails géométriques à une certaine résolution (ainsi seules les singularités correspondant à la forme globale de l'objet apparaissent). Cette méthode sera l'objet de la Chapitre 10.

³cet exemple est tiré de l'article [Ray et al. 2008c] (en cours de soumission) auquel l'auteur a contribué mais qui n'est pas présenté dans cette thèse car il en dépasse le cadre. C'est un exemple de méthode de synthèse de texture basée sur un champ de directions créé en utilisant des méthodes présentés dans cette partie. L'originalité de l'article repose sur la possibilité de contrôler un paramètre de texture qui peut varier sur la surface, comme l'illustre la figure

Pour visualiser nos résultats, nous avons aussi du développer un outil de visualisation efficace et spécialisé à notre discrétisation des champs de directions à symétries proposée à la Section 7.4. Cette visualisation est décrite au Chapitre 8.

Les chapitres 7 à 10 sont issus des trois articles :

- [Li et al. 2006b] pour la discrétisation basée sur les sauts de périodes (Section 7.4) et la visualisation (Chapitre 8) ;
- [Ray et al. 2008b] pour la théorie sur la topologie des champs de directions (de la Section 7.1 à la Section 7.3), et pour l’algorithme de manipulation de champs de directions avec contrôle topologique *fort* décrit au Chapitre 9 ;
- [Ray et al. 2008a] pour la formalisation de tous ces concepts en calcul extérieur discret, et pour l’algorithme de manipulation de champs de directions avec contrôle topologique *faible* décrit au Chapitre 10.

Ces articles ayant été écrits en anglais, nous avons choisi de préserver le contenu correspondant dans cette thèse afin de ne pas fausser la terminologie utilisée par la communauté internationale. Par contre, nous avons harmonisé les notations et réorganisé le contenu pour qu’il s’intègre dans la structure globale de la thèse (en particulier les parties théoriques sont factorisées dans ce chapitre). De plus, nous présenterons systématiquement une introduction, un plan/résumé et une liste de contributions en français pour chacun de ces chapitres.

Les contributions principales de cette partie sont :

- une présentation des enjeux liés à la manipulation des champs de directions et l’état de l’art dans ce domaine (Chapitre 6) ;
- une formalisation pour les champs de directions à symétries et une étude approfondie de leur topologie (Chapitre 7) ;
- une discrétisation de ces champs permettant d’avoir accès à leur topologie.
- l’introduction d’outils pour interpoler et visualiser ces champs discrétisés (Chapitre 8) ;
- un algorithme de manipulation de champs de directions à symétries avec une topologie prescrite. Cet algorithme est le premier à assurer un contrôle exact de la topologie de façon prouvable (Chapitre 9) ;
- un algorithme de manipulation de champs de directions à symétries qui permet d’extrapoler des contraintes utilisateur avec un contrôle plus faible sur la topologie. Au lieu de spécifier toutes les singularités du champs, l’utilisateur définit un niveau de détail auquel le lissage doit être sensible, de sorte que les singularités n’apparaissent que pour capturer la forme de l’objet au dessus de ce niveau de détail (Chapitre 10).

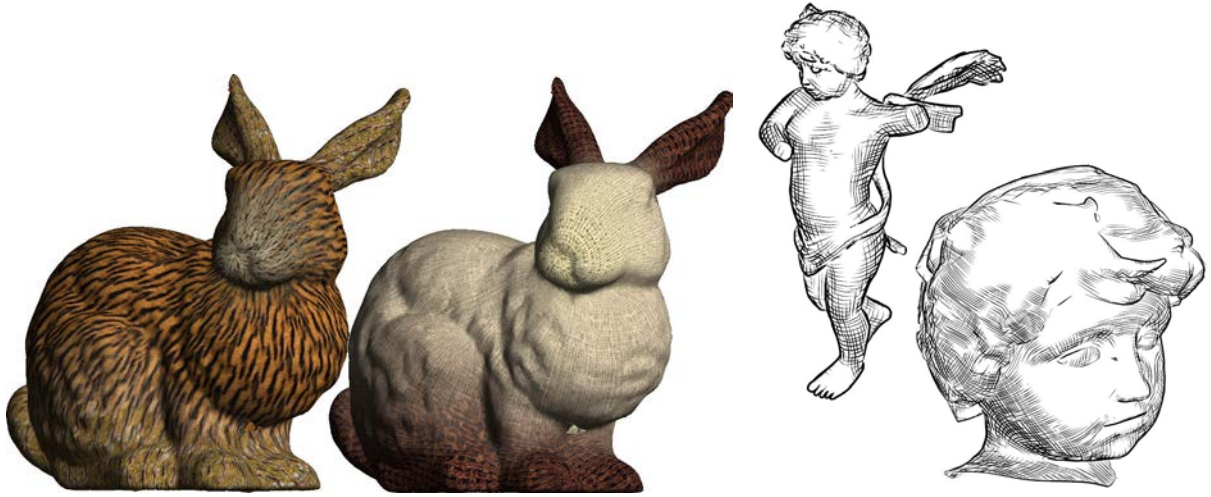


FIG. 5.13 – Deux applications qui nécessitent des champs de directions : la synthèse de texture (gauche) et le rendu non photoréaliste (droite)

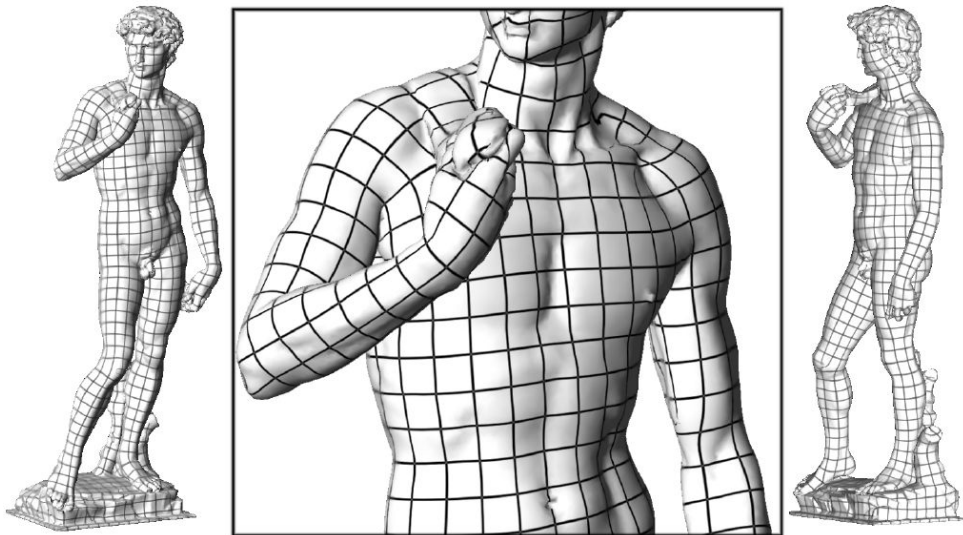


FIG. 5.14 – Cette paramétrisation quad d'une statue de David se base sur un champs directions à 4 symétries.

6

Les champs de directions

Sommaire

6.1 Applications	98
6.1.1 Rendu non photoréaliste	98
6.1.2 Synthèse de texture	98
6.1.3 Paramétrisation et remaillage quad	99
6.1.4 Conclusion	100
6.2 Caractéristiques des outils de manipulation de champs de directions	100
6.3 Etat de l'art	102
6.3.1 Lissage géométrique des champs de directions (GDS)	104
6.3.2 Création de champs de vecteurs tangents (TVFD)	105
6.3.3 N-rosy	106
6.4 Conclusion	108

Introduction

Le principal objectif de ce chapitre est d'introduire la notion de champ de directions définis sur des surfaces triangulées, ainsi que les problèmes liés à leur manipulation (création interactive ou traitement automatique) :

- Nous désignons par champs de directions tout objet qui munit une surface d'une orientation locale de son plan tangent en chaque point (vecteur tangents, vecteurs tangents unitaires), avec d'éventuelles symétries par rotation (tenseurs, ensembles de vecteurs invariants par rotation)
- Ce que nous appelons *création interactive* est la génération de champ de directions lisses extrapolés à partir des contraintes fournies par l'utilisateur. Ces contraintes peuvent être de nature géométriques (imposer une direction) ou topologiques (positionner des singularités).
- Ce que nous appelons *traitement automatique* fait référence aux opérations de lissage et de simplification topologique de champs existants. Ce qui différencie ce problème de la création interactive est que l'on suppose l'existence d'un champ de direction initial, mais le traitement de ce champ fera ensuite appel aux mêmes outils que la création interactive (opérations géométriques et topologiques, lissage).

Organisation du chapitre

Le problème de la manipulation des champs de directions fut initialement abordée par des algorithmes simples utilisés comme une première étape de divers algorithmes. Ainsi, nous commencerons par présenter les applications des champs de directions dans Section 6.1. Nous en déduirons les propriétés importantes qu'un système de manipulation de champs de directions peut avoir (Section 6.2). Nous présenterons ensuite dans Section 6.3 les travaux précédents dans ce domaine, et discuterons leurs capacité à fournir les propriétés souhaitées. Nous détaillerons plus particulièrement les trois méthodes les plus en relations avec les objectifs poursuivis dans cette thèse.

Contributions

- Nous présentons une synthèse des diverses applications des champs de directions afin d'en déduire les besoins réels auxquels devrait répondre un système de manipulation de champs de directions.
- Nous fournissons aussi une vue globale de l'état de l'art de ce domaine, et détaillons trois travaux dont les objectifs sont particulièrement proches de ceux que nous nous sommes fixés.

6.1 Applications

Les principales applications des champs de directions sont : le rendu non réaliste, la synthèse de texture, la paramétrisation/remailage quad. Nous présentons rapidement les deux premières applications, puis donnerons d'avantage de détails pour la paramétrisation quad qui fut notre motivation initiale.

6.1.1 Rendu non photoréaliste

Le rendu non photoréaliste (NPR pour *Non-Photorealistic Rendering*) est une expression qui désigne l'ensemble des algorithmes de rendus dont l'objectif n'est pas de rendre compte au mieux d'une réalité physique, mais plutôt d'enrichir les images générées par ordinateur avec des styles inspirés du monde artistique. En effet, depuis la découverte de la photographie, le réalisme n'est plus l'objectif principal des artistes qui cherchent plutôt à véhiculer des émotions, des impressions, des concepts, des styles, des revendications... Le rendu non photoréaliste est un domaine dans lequel l'objectif est de mettre au point des algorithmes capables de simuler ces différents styles. Ce type de rendu prend en compte les contraintes physiques auxquelles sont soumis les artistes : crayon, encre, peinture et texture du support (toile, papier). L'action de base est généralement le tracé de courbes grâce à l'outil (pinceau, crayon). Dans le cas du rendu d'une scène 3D, la direction de ces courbes peut être définie directement sur les surfaces des objets à afficher. Par exemple, Hertzmann et al. [Hertzmann and Zorin 2000] ont réalisé un rendu de surface avec un style proche du "dessin à l'encre" (cf Figure 5.13 droite) dans lequel les coups de crayon sont orientés selon le champ des courbures principales de la surface (un champ de directions à 4 symétries de rotation).

6.1.2 Synthèse de texture

Les textures sont des images qui stockent l'information de couleur des surfaces. Utiliser une seule texture pour représenter les couleurs d'un objet n'est pas toujours une solution optimale car si les couleurs de

l'objet varient beaucoup, une importante quantité de mémoire sera nécessaire pour la stocker. De plus, il est nécessaire de calculer un atlas de texture pour déterminer comment la texture va s'appliquer à l'objet.

Une solution alternative est de synthétiser une texture directement sur la surface en se basant sur un échantillon de texture (voir Figure 5.13 dans l'introduction). Cette méthode permet de représenter efficacement des matériaux dont les couleurs varient rapidement tels que les textiles, les revêtements de murs, etc. Pour tenir compte d'éventuelles anisotropies dans les textures et pour simplifier les calculs, ces algorithmes de synthèse utilisent souvent un champ de directions pour définir a priori l'orientation de la texture sur les objets.

Par exemple, dans les *lapped textures* [Praun et al. 2000] et dans les *texture particles* [Dichler et al. 2002], un champ de directions permet de contrôler l'orientation et la taille des morceaux de texture qui seront distribués sur l'objet. Les méthodes qui réalisent une synthèse à la résolution du texel [Turk 2001] utilisent aussi des champs de directions pour orienter leur texture. Dans [Wei and Levoy 2001], les textures utilisées sont invariantes par rotations de π ou $\pi/2$. Leur synthèse nécessite donc d'être guidée par des champs de directions présentant les symétries correspondantes.

6.1.3 Paramétrisation et remaillage quad

Le problème de la paramétrisation quad est relativement récent dans le domaine de la géométrie numérique. Il fut tout d'abord étudié pour résoudre les problèmes de discontinuités entre morceaux dans les atlas de textures. En effet, un objet doit nécessairement être découpé pour pouvoir être aplati (mis en bijection avec un sous ensemble de \mathbb{R}^2), ce qui est requis pour pouvoir lui appliquer directement une texture. Le principe de la paramétrisation quad est faire disparaître ces discontinuités de textures en imposant des contraintes à travers les coupures qui préservent la grille de pixels dans l'espace des textures (*texels*) afin de permettre une certaine continuité visuelle. Nous allons maintenant détailler cette méthode.

Une paramétrisation d'une surface triangulée $M = \{\mathcal{V}, \mathcal{E}, \mathcal{F}\}$ est un ensemble de fonctions bijectives $\phi_t : t_2 \subset \mathbb{R}^2 \rightarrow t \in \mathcal{F}$ où t_2 est un disque topologique de \mathbb{R}^2 et t un triangle de la surface triangulée. Tous les travaux existants traitent des paramétrisations linéaires pour lesquelles t_2 est un triangle et ϕ_t est définie de façon unique par les coordonnées (u, v) des sommets de t_2 .

On s'intéresse particulièrement aux fonctions de transition $\Phi_{ij} = \phi_{jil} \circ \phi_{ijk}^{-1}$ entre deux triangles ijk et jil qui partagent l'arête ij . Dans les méthodes de paramétrisation usuelles, la surface est coupée en un ou plusieurs disques topologiques par un graphe de coupure $G \subset \mathcal{E}$. Cela correspond à imposer $\Phi_{ij} = Id$ le long de toutes les arêtes ij , sauf pour celles de G . Une telle paramétrisation est entièrement définie par un point et les gradients ∇u et ∇v , c'est à dire par deux champs de vecteurs définis sur le disque topologique $[M]/[G]$. Le but de la paramétrisation globale est d'éviter de telles coupures. Comme il est impossible d'imposer $\Phi_{ij} = Id$ sur toutes les arêtes, une paramétrisation globale impose seulement que les transitions à travers G soient des multiples entiers le long des axes u et v :

$$\Phi_{ij}(\mathbf{u}) = \mathbf{u} + (n_u, n_v), \quad n_u, n_v \in \mathbb{Z}.$$

Enfin la paramétrisation quad impose une condition plus faible puisque les transitions à travers G sont de la forme :

$$\Phi_{ij}(\mathbf{u}) = \mathbf{u} + (n_u, n_v)^t \text{ with } n_u, n_v \in \mathbb{Z}$$

où J est la matrice de rotation d'angle $+\pi/2$ dans \mathbb{R}^2 . Cette condition plus faible rend très difficile de définir la paramétrisation par l'intermédiaire de deux champs de vecteurs, car les conditions sur les transitions sont difficiles à assurer directement sur les champs de vecteurs. Au mieux, on peut imposer

la transition, ou trouver des méthodes heuristiques pour déterminer n_u , n_v et r sur chaque arête, ce qui conduit à la méthode proposée dans [Tong et al. 2006]. Pour simplifier le problème, on peut alors le séparer en deux parties séparées :

- Trouver les directions de ∇u et ∇v de sorte qu’elles vérifient la condition de rotation sur les transitions. A cause de ces rotations multiples de $\pi/2$, les directions de ∇u et ∇v peuvent être interverties ou changées en leurs opposées. Ainsi, on les représentera par un champ de directions avec une invariance par rotation de $\Pi/2$. La manipulation de tels champs est le problème qui est abordé dans cette partie.
- Trouver la norme de ∇u et ∇v de sorte que les conditions sur les translations soient vérifiées. Pour assurer que la paramétrisation/remailage ne comporte que des quadrilatères, on doit en plus imposer que les singularités du champ de directions formé par ∇u et ∇v aient des coordonnées entières dans l’espace des textures. Une solution (délicate) à ce problème a été récemment proposée dans [Kalberer et al. 2007].

La paramétrisation quad n’assure pas la continuité dans \mathbb{R}^2 (c’est d’ailleurs impossible sans retournement pour les surfaces fermées), mais assure que la grille $\mathbb{Z} \times \mathbb{R} \cup \mathbb{R} \times \mathbb{Z}$ est préservée par les transitions. L’image de cette grille sur la surface forme des quadrilatères (cf Figure 5.14), de sorte qu’une paramétrisation quad implique trivialement un remailage par des quadrilatères. Sur ces quadrilatères, on peut installer de nouvelles bases de fonctions comme la base bilinéaire définie au Chapitre 2, mais aussi la base des cosinus discrets (utilisée par la norme JPEG) ou même des ondelettes, ce qu’on ne pouvait pas faire directement sur les triangles.

Cela justifie notre intérêt pour la paramétrisation quad dans le cadre de cette thèse sur les bases de fonctions. Ces quads seront orientés par un champ de directions à symétrie de rotation, et leur taille peut être réglée par l’algorithme décrit dans [Kalberer et al. 2007]. Cet algorithme fonctionne bien, mais il est fortement dépendant de la qualité et des caractéristiques du champ de directions, ce qui explique que nous nous soyons intéressés à ce sujet.

6.1.4 Conclusion

De toute évidence, ces applications ont besoin que les champ de directions qu’elles utilisent présentent certaines caractéristiques et qualités particulières. Nous allons maintenant faire la liste de ces caractéristiques afin de disposer de critères pour évaluer les travaux antérieurs sur les champs de directions.

6.2 Caractéristiques des outils de manipulation de champs de directions

Différentes caractéristiques sont nécessaires pour un outil de manipulation de champs de directions en fonction des applications mentionnées dans la section précédente. Nous allons commencer par exposer quatre de ces caractéristiques, puis nous nous concentrerons sur l’importante distinction entre les vecteurs et les directions.

Pour les applications que nous avons présentées, quatre caractéristiques nous paraissent significatives (cf Figure 6.1) :

1. *Régularité* : le champ de directions devrait être aussi lisse que possible. Cela assure une bonne qualité visuelle, mais fait aussi émerger les singularités naturellement.
2. *Invariance par rotation* : Ces applications utilisent des champs de directions qui présentent des invariances par rotation de π ou $\pi/2$, et qu’on généralisera par le concept de directions à N symétries de rotation (ensemble de N directions invariants par rotation multiples de $2\pi/N$). En particulier,

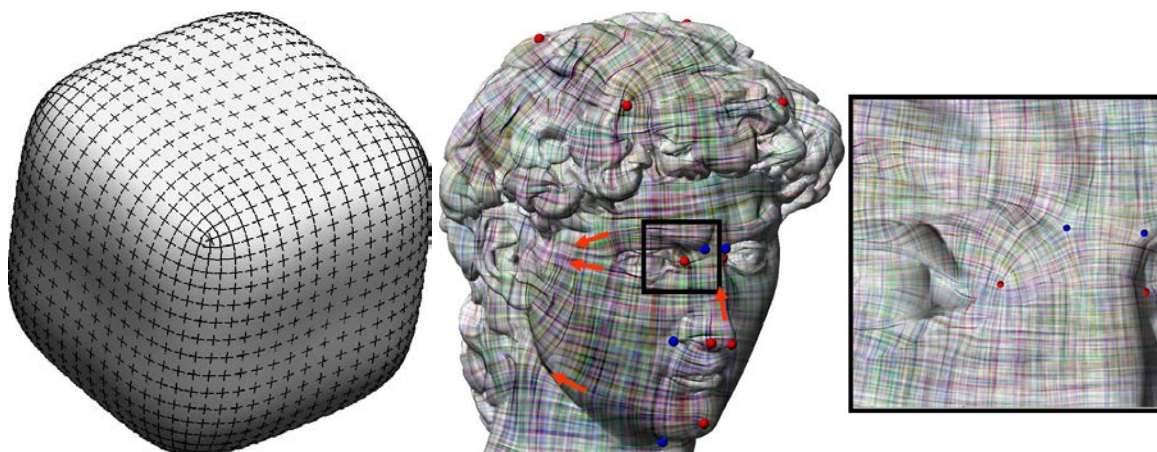


FIG. 6.1 – Direction field features : **left** : tolerating symmetries allows for the definition of "cross fields" on which directions might be switched, **right** : we call *geometric control* the ability to control directions at given points (orange arrows) and *topologic control* the ability to control the type and placement of singularities (red and blue dots).

les champs de directions à 1,2 et 4 symétries correspondent aux champs de directions (\rightarrow), lignes ($-$) et croix ($+$). Les champs de lignes sont utilisés pour le rendu non photo réaliste (hachures, traits de pinceau). Les champs de croix sont utilisés en paramétrisation/remailage quad.

3. *Contrôle géométrique* : on veut contrôler la géométrie du champ, c'est à dire soit spécifier certaines contraintes sur la direction, soit lisser un champ de directions existant (tel que les directions principales de courbure d'une surface). Dans les deux cas, cela impliquera des contraintes (dures ou faibles) sur la géométrie du champ.
4. *Contrôle topologique* : dans la plupart des applications, les singularités du champ jouent un rôle important. En remailage quad par exemple, une singularité dans le champ de directions se traduira par une cellule non-quadrilatérale ou par point de valence différente de 4 dans le remailage, ce qu'on cherche à éviter autant que possible. Dans une application de rendu non photoréaliste, une singularité correspondra à un point de convergence des traits, ce qui aura un fort impact visuel. Il est donc très important pour la plupart des applications de pouvoir contrôler les singularités. Ainsi, on voudra soit contrôler la position, le type et le nombre de singularités (Chapitre 9) ou au moins limiter l'apparition des singularités de façon à ce qu'elles ne dépendent que de la forme générale de l'objet, et non des détails secondaires (Chapitre 10).

Une autre question importante est de décider si nous voulons travailler avec des vecteurs ou vecteurs unitaires. Pour les applications géométriques, la norme du vecteur n'est pas toujours significative (par exemple en rendu non photoréaliste où seules les direction des traits nous intéresse). De plus, lorsque nous parlons de la géométrie et de la topologie d'un champ de vecteur, nous faisons référence à ceux de ses lignes de champs, qui dépendent uniquement de la direction du champ de vecteur, et non de sa norme. Même pour les applications où la norme a un sens, il peut être souhaitable de traiter la direction et la norme séparément. Par exemple, en remailage quad, nous voulons aligner les quads avec une certaine direction (comme une estimation des directions de courbure principale), mais les tailles des quad (qui correspondent aux normes des vecteurs) doivent être déterminées séparément pour assurer une bonne topologie. En d'autres termes, le problème est rendu plus facile en séparant la détermination de l'orientation et de la taille.

Cela aura un impact sur le formalisme et les structures de données que nous allons utiliser. En parti-

culier, un champ de vecteurs tangents est généralement décrit par une paire de coordonnées dans une base locale de chaque triangle (structure standard), ou par son intégrale le long des arêtes (structure du calcul extérieur discret). Pour passer aux directions, il faut rajouter la contrainte de normalisation à ces structures. Dans la structure standard, ce sera plutôt simple. Dans la structure du calcul extérieur discret, la contrainte de normalisation est difficile à appliquer aux 1-formes, mais on utilisera cependant le formalisme du calcul extérieur discret dans lequel la courbure et les indexes des singularités des champs de directions seront caractérisés de façon simple. Dans cette thèse, nous présenterons un point de vue purement basé sur les directions, définies par des angles, et non par des vecteurs donnés par deux coordonnées dans une base locale avec une contrainte de normalisation. La principale différence est que nous introduisons le nombre exact de degrés de liberté : 1 angle par direction, au lieu de 2 coordonnées + 1 contrainte.

Enfin, la régularité du champ de directions telle que nous allons la définir au Chapitre 7 correspond à la rectitude (faible courbure) des lignes de champ, alors que le critère habituel de régularité (intégrale du carré des dérivées) est moins géométrique. Par exemple, imaginons un champ de vecteurs aléatoires avec de très faibles normes. Il aura une très bonne régularité (pour le critère habituel) alors que ses lignes de champs peuvent être très fortement courbées. Si l'on ajoute un vecteur constant de norme importante (ce qui ne change pas le critère de régularité), les lignes seront beaucoup plus droites, et le champ sera visuellement beaucoup plus régulier (lisse). Dans la pratique, cela signifie que si nous avons un champ de vecteurs lisse (pour le critère usuel) sous certaines contraintes, la norme du va décroître rapidement pour devenir très faible loin des contraintes où la géométrie des lignes de champs y sera mal maîtrisée. En revanche, si nous nous basons sur les directions, la norme n'a pas d'influence, de sorte que le critère de régularité sera géométriquement équivalent partout, et correspondra à la rectitude des lignes de champs.

Nous pouvons maintenant présenter l'état de l'art en utilisant ces caractéristiques pour comparer les différentes approches.

6.3 Etat de l'art

Nous allons maintenant présenter l'état de l'art en commençant par un aperçu général des méthodes existantes. Ensuite, nous nous concentrerons sur trois méthodes particulièrement intéressantes dans le cadre de cette thèse.

Présentation générale

Les méthodes que nous allons présenter ont été en général conçues pour répondre à une application spécifique telles que celles que nous avons mentionnées dans la Section 6.1. A part quand nous le mentionnerons explicitement, ces méthodes s'intéressent aux champs de vecteurs, c'est à dire sans symétries de rotation et non normés. Ainsi, on s'intéressera surtout au contrôle géométrique et topologique qu'elles permettent d'obtenir.

Dans le cadre de la synthèse de texture, [Praun et al. 2000] propose de générer un champ de vecteurs lisse en utilisant des bases de fonctions radiales gaussiennes nécessitant de calculer des distances géodésiques sur la surface. Cette méthode permet d'interpoler des vecteurs définis par l'utilisateur, mais la manipulation est très dépendante de la largeur de la Gaussienne et on n'a pas de contrôle sur la topologie. La méthode de lissage de champs de vecteurs proposée dans [Turk 2001] s'inspire de [Gortler et al. 1996] qui est basé sur un lissage multi-résolution par le Laplacien. Cela rend le lissage efficace et les contraintes

géométriques faciles à introduire, mais donne peu de contrôle sur la régularité du champ loin des contraintes. Une procédure similaire est décrite dans [Ohtake et al. 2001], avec en addition des poids non linéaires qui assurent une meilleure continuité du champ.

Toujours dans le cadre de la synthèse de texture, [Zelinka and Garland 2004] orientent une méthode de génération de texture en utilisant un champ de vecteurs défini comme le gradient d'une fonction de Morse. Les singularités du champ correspondent donc exactement aux extréma et points selle de la fonction. En se basant sur l'étude du complexe de Morse de fonctions harmoniques [Ni et al. 2004], cela permet de contrôler le nombre et la configuration des singularités. Cependant, cette approche offre des possibilités limitées car le champ de vecteurs résultant est nécessairement un gradient, qui est donc irrotationnel. Le contrôle topologique fort que nous présenterons au Chapitre 9 a certain points communs avec [Zelinka and Garland 2004] et [Gu and Yau 2003] dans le sens où il permet de contrôler explicitement les singularités. La différence majeure est que dans ces deux méthodes, le champ de vecteur est défini comme le gradient d'un champ scalaire, ce qui restreint fortement les possibilités. De plus ces approches ne sont applicables que pour des champs de vecteurs sans symétries, alors qu'en les permettant, nous pouvons représenter une classe plus large de singularités avec des indices *arbitraires*.

Des applications différentes ont parfois conduit à des méthodes de manipulation de champs de vecteurs similaires. Par exemple, l'orientation des traits dans le hachurage en temps réel de [Praun et al. 2001] et des quadrilatères dans le remaillage anisotrope de [Alliez et al. 2003] sont tous les deux définis par les axes principaux du tenseur de courbure de la surface. Ce tenseur définit une direction avec une symétrie de rotation de $\pi/2$, qui doit être lissées pour éliminer une partie de ses singularités et pour éviter les effets de tremblement dans les applications. Cependant, même si le lissage fait décroître le nombre de singularités, il n'assure pas la génération d'un nombre minimum de singularités.

La plupart des travaux ayant traité des champs de vecteurs à symétries de rotation se sont portés sur le cas des "champs de croix" : $\mathbf{v}_1, \mathbf{v}_2 = R(\mathbf{v}_1, \pi/2)$, où les vecteurs \mathbf{v}_1 et \mathbf{v}_2 peuvent être échangés (cf l'objet cubique de la Figure 6.1). La plupart des algorithmes de lissage de champs de vecteurs mentionnés précédemment utilisent des procédures de relaxation *locale*, en mettant à jour les valeurs $\mathbf{v}_1(i)$ et $\mathbf{v}_2(i)$ *sommet par sommet*. Lors de cette relaxation, $\mathbf{v}_1(i)$ sera influencé soit par $\mathbf{v}_1(j)$ soit par $\mathbf{v}_2(j)$ pour les sommets j voisins de i . L'algorithme choisit la direction la plus proche de $\mathbf{v}_1(i)$. En conséquence, des singularités peuvent apparaître sans contrôle et la convergence est lente. Au contraire, en se basant sur une analyse topologique des champs de directions définis sur des surfaces (Chapitre 7), on dérive dans le Chapitre 9 une formulation *globale* du problème par la minimisation d'une forme quadratique, conduisant à une optimisation plus efficace (gradient conjugué), qui permet de calculer directement la solution optimale.

Certaines publications récentes abordent des problèmes spécifiques liés à la manipulation de champs de vecteurs. Par exemple, des procédures permettant de calculer la décomposition de Hodge (cf Section 6.3.2) d'un champ de vecteur sont proposées et utilisées dans [Polthier and Preuss 2002] et [Tong et al. 2003]. Une telle décomposition isole certaines caractéristiques du champ et permet de les contrôler. Dans [Tricoche et al. 2003], une méthode est présentée pour simplifier la topologie d'un champ de tenseurs du second ordre en 2D. De tels tenseurs ont deux vecteurs propres orthogonaux et peuvent donc être assimilés aux "croix" de nos champs de croix (de plus la norme d'un vecteur propre \mathbf{v} n'a pas de sens puisque $\lambda \mathbf{v}$ est encore vecteur propre). Enfin, [Wang et al. 2006] propose un schéma de subdivision pour les 0- et 2-formes différentielles, et s'en sert pour définir un champ de vecteurs tangents (1-forme) sur une surface de subdivision.

Nous allons maintenant nous intéresser plus en détail à trois méthodes qui sont particulièrement pertinentes dans le cadre de cette thèse.

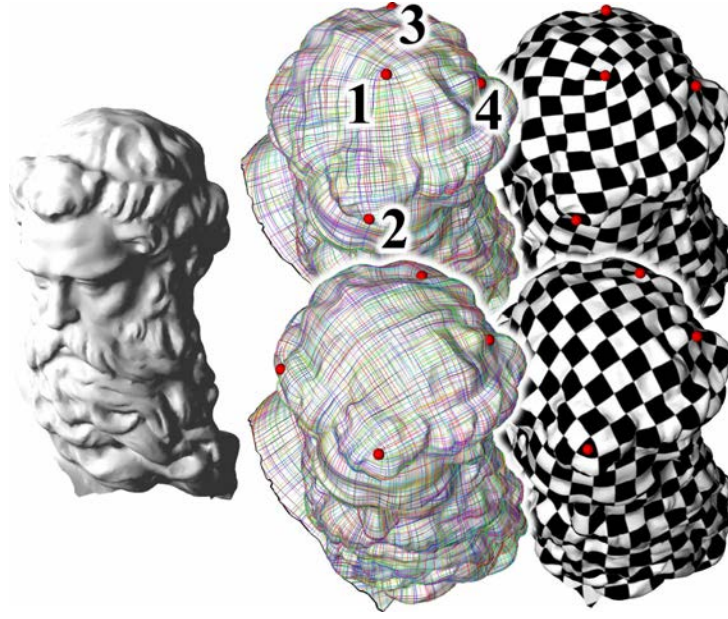


FIG. 6.2 – Pour calculer une paramétrisation quad de cette statue de Neptune, des singularités peuvent être introduites itérativement (en haut), ou calculées simultanément par lissage d’un champ de directions (en bas). Cette seconde approche permet une meilleure distribution des singularités.

6.3.1 Lissage géométrique des champs de directions (GDS)

Dans [Hertzmann and Zorin 2000] (pour le rendu non-photoréaliste) et [Li et al. 2006a, Ray et al. 2006] (pour le remaillage quad), ses champs de directions sont définis modulo $\pi/2$ pour prendre en compte l’invariance de l’orientation des hachures ou des quadrilatères par rotation de $\pi/2$. Comme les méthodes proposées dans ces trois travaux sont assez similaires, on utilisera pour les désigner le même terme GDS (pour *Geometric Direction Smoothing*). Cette méthode se propose de lisser un champ de directions à N -symétries, tels que des champs de directions ($N = 1$), de lignes ($N = 2$) ou de croix ($N = 4$).

La méthode GDS génère un champ de directions lisse sous un ensemble de contraintes géométriques faibles ou fortes. Une direction à N symétries est définie sur chaque triangle t_i par un *vecteur de représentation* V_i défini dans une base locale (choisie de façon arbitraire) de t_i . On obtient les N directions en divisant l’angle entre V_i et le premier vecteur de la base par N , et en ajoutant tous les multiples de $2\pi/N$. L’objectif de régularité de la fonction prend alors la forme :

$$V_i - R(N, \kappa_1^r(ij))V_j = 0 \quad (6.1)$$

où $\kappa_1^r(ij)$ est l’angle entre les bases locales (arbitraires) de t_i et t_j , et $R(\alpha)$ est la matrice de rotation d’angle α :

$$R(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}$$

(6.1) est un objectif linéaire avec seulement des variables continues, ce qui permet de construire une énergie quadratique :

$$E_{smooth} = \sum_{ij \in \mathcal{E}^*} (V_i - R(N, \kappa_1^r(ij))V_j)^2$$

On peut ajouter une contrainte faible d’attache aux données $E_{fit} = \sum_i ||t_i|| (V_i - V_i^{init})^2$, où $||t_i||$ est l’aire du triangle t_i . Cela conduit à minimiser $E_{smooth} + \lambda E_{fit}$ où λ permet de choisir le compromis entre régularité

du champ et attache aux données. Des contraintes fortes peuvent être introduites dans le système en bloquant certains vecteurs V_i comme expliqué dans [Levy 2005] ou dans l'Annexe A. La norme des solutions obtenues décroît en général rapidement en s'éloignant des contraintes (puisque le champ le plus lisse possible est nul). On a mentionné ce problème important dans la Section 6.2, et la solution proposée dans [Li et al. 2006a] est de normaliser tous les $\|V_i\|$ après chaque itération.

Pour lisser un champ de directions donné, on choisit simplement ces directions comme V^{init} , et on itère plusieurs fois en utilisant le résultat de chaque étape comme V^{init} pour l'étape suivante. Pour extrapoler des directions définies par l'utilisateur, elles sont imposées par des contraintes fortes dans une première étape sans attache aux données, puis le résultat est lissé itérativement comme précédemment en préservant les contraintes fortes.

6.3.2 Création de champs de vecteurs tangents (TVFD)

Récemment, Fischer *et al.* [Fisher et al. 2007] ont proposé une méthode qu'on nommera TVFD (pour *Tangent Vector Field Design*) basée sur le calcul extérieur discret. Cette méthode se base sur la décomposition de Hodge des champs de vecteurs pour générer des 1-formes (équivalentes aux champs de vecteurs). Les auteurs appliquent cette méthode à la synthèse de texture. Nous allons présenter la méthode TVFD en commençant par expliquer la décomposition de Hodge, puis en décrivant la méthode aux moindres carrés pondérés utilisée dans [Fisher et al. 2007] pour l'implémenter.

Décomposition de Hodge

La décomposition de Hodge consiste à écrire une 1-forme discrète c^1 comme somme de trois termes :

$$c^1 = dc^0 + \delta c^2 + h$$

où c^0 et c^2 sont des 0- et 2-formes discrètes (scalaires sur les sommets et sur les triangles), et h est une 1-forme harmonique (i.e. $dh = 0$ et $\delta h = 0$). En termes de champs de vecteurs, cela signifie que l'on peut séparer le champ c^1 en un champ sans rotationnel (dc^0), un champ incompressible (δc^2) et un champ harmonique (h). Le rotationnel et la divergence de c^1 sont $dc^1 = d\delta c^2 = r^2$ et $\delta c^1 = \delta dc^0 = s^0$ respectivement.

Moindres carrés pondérés

En se basant sur cette décomposition de Hodge, on peut contrôler le rotationnel et la divergence de c^1 , ainsi que rajouter des contraintes linéaires de la forme $Zc^1 = z$ sur c^1 par un simple système linéaire :

$$\begin{pmatrix} d \\ \delta \\ Z \end{pmatrix} c^1 = \begin{pmatrix} r^2 \\ s^0 \\ z \end{pmatrix}$$

que l'on peut résoudre au sens des moindres carrés pondérés :

$$(\star\delta, \star d, Z^T W) \begin{pmatrix} d \\ \delta \\ Z \end{pmatrix} c^1 = (\star\delta, \star d, Z^T W) \begin{pmatrix} r^2 \\ s^0 \\ z \end{pmatrix}$$

$$(M + Z^T W Z)c_e = \star \delta r^2 + \star ds^0 + Z^T W z$$

Cette approche permet de manipuler des champs de vecteurs tangents sous un ensemble de contraintes linéaires et en contrôlant la divergence et le gradient du champ. En absence de contraintes linéaires (pas de Z), le contrôle sur la divergence permet de placer des sources (coefficients positifs dans s^0), des puits (coefficients négatifs dans s^0) et des tourbillons (coefficient non nul dans r^2 , le signe définissant l'orientation). Si la surface n'a pas de bords, s^0 et r^2 doivent être de moyenne nulle puisqu'elles sont définies comme le Laplacien de 0 et 2 formes. Si il y a un bord, la matrice du Laplacien doit être modifiée en conséquence pour incorporer une condition d'angle fixe à la frontière où une frontière libre.

Les contraintes linéaires les plus naturelles que l'on peut imposer sur un champ de vecteurs est d'imposer sa direction. En calcul extérieur discret, les champs de vecteurs (1-formes) sont définis par un scalaire sur chaque arête orientée et contraindre ce scalaire ne contraint pas la direction du champ. Ainsi, des contraintes plus globales peuvent être appliquées en laissant l'utilisateur dessiner une courbe (que le champ devrait suivre) sur la surface, et en imposant que les scalaires sur les arêtes intersectées par la courbe valent $\mathbf{t} \cdot |ij|$ où \mathbf{t} est la tangente à la courbe au point où elle intersecte ij . Les contraintes de ce type ont bien la forme linéaire requise : $Zc^1 = z$.

Discussion

Cette méthode permet d'obtenir un très bon contrôle des singularités en l'absence de contraintes géométriques (s^0 et r^2 non nuls, Z nul), et un bon contrôle géométrique en l'absence de contraintes topologiques (s^0 et r^2 nuls, Z non nul). Cependant, ces contrôles sont exclusifs : si l'utilisateur veut à la fois contrôler les directions et les singularités, les contraintes correspondantes seront en conflit. En particulier, un rotationnel non nul n'assure la présence d'un tourbillon qu'en l'absence d'une contrainte directionnelle à proximité. Réciproquement, une contrainte sur une direction peut générer des singularités non désirées. De plus, la norme du champ de vecteurs décroît rapidement loin des contraintes, et donc le critère de régularité y aura moins d'effet. Les auteurs de [Fisher et al. 2007] proposent de régler ce problème en ajoutant un petit multiple du bilaplacien au Laplacien utilisé, ce qui ne fait que modérer la vitesse de décroissance, au prix d'un système linéaire moins creux. Enfin, si la surface n'est pas une sphère topologique, l'existence de 1-formes harmoniques non triviales nécessite une attention particulière.

Cette méthode est simple et pratique à utiliser dans des cas simples, et peut sûrement être utilisée de façon efficace par un artiste expérimenté. Elle est basée sur un formalisme mathématique solide et clair, et utilise la décomposition de Hodge de façon simple. En revanche, elle ne permet pas de bien contrôler l'apparition des singularités, en particulier sur les objets compliqués. Enfin, cette méthode n'est pas généralisable aux champs de directions à symétries.

Une telle généralisation est en revanche possible sur des champs de vecteurs décrits de façon classique. Nous allons maintenant présenter les travaux de Zhang et Palacios qui expliquent comment faire cette généralisation.

6.3.3 N-rosy

Zhang *et al.* fournissent dans [Zhang 2004] un ensemble complet d'outils pour l'édition interactive de champs de vecteurs, basé sur le déplacement et la simplification de paires de singularité. Cette approche est ensuite généralisée aux tenseurs dans [Zhang et al. 2006] et [Zhang et al. 2007] et finalement aux champs de vecteurs à N symétries de rotations (N -RoSy) par Palacios dans [Palacios and Zhang 2007].

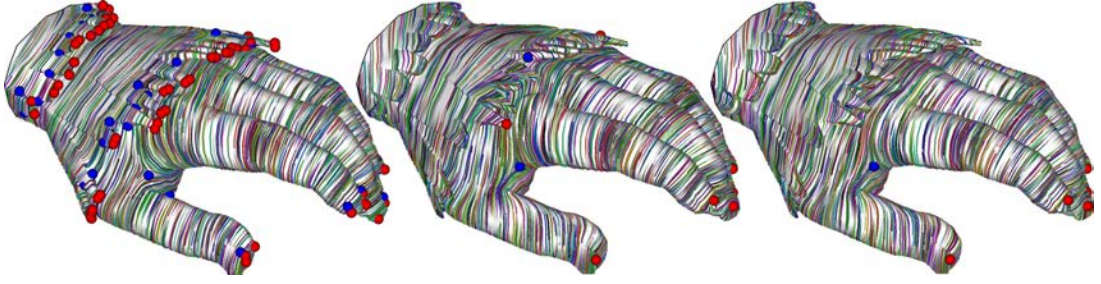


FIG. 6.3 – Lissage d'un champ de directions à 2 symétries sur une surface avec des détails géométriques fins. **Gauche** : Un simple lissage local (comme GDS) génère de nombreuses singularités. **Milieu** : La stratégie de simplification de paires de singularités proposée dans N-RoSy réduit le nombre de singularités mais n'assure pas qu'un minimum global sera trouvé. **Droite** : La méthode que nous exposerons au Chapitre 10 permet d'ignorer les détails tout en restant sensible à la forme globale de la surface (doigts).

Cette section se concentre sur cette dernière approche [Palacios and Zhang 2007] qui décrit un outil permettant de manipuler des champs de vecteurs à N symétries à partir de contraintes utilisateur.

La généralisation des champs de vecteurs proposée par N -rosy est basée sur l'idée de *vecteurs de représentation*. Dans cette approche, un vecteur à N symétries s_N est un ensemble de vecteurs de la forme :

$$\begin{pmatrix} R \cos(\theta + 2k\pi/N) \\ R \sin(\theta + 2k\pi/N) \end{pmatrix} \quad (6.2)$$

dans une base locale. Le *vecteur de représentation* correspondant est :

$$\begin{pmatrix} R \cos(N\theta) \\ R \sin(N\theta) \end{pmatrix}.$$

Cette idée est déjà présente (sans ce nom ni ce formalisme) dans l'outil de manipulation de champs de vecteurs de [Ray et al. 2006], où des variables de la forme $x = \cos(N\theta)$ and $y = \sin(N\theta)$ sont introduites pour chaque direction avec la contrainte $x^2 + y^2 = 1$. Cette contrainte de normalisation constitue la différence majeure entre les deux approches.

La difficulté principale lorsque l'on travaille avec des vecteurs de représentation est qu'ils ne se comportent pas comme des vecteurs normaux par changement de base. En fait, Palacios *et al.* montrent qu'un N -Rosy s_N sous la forme (6.2) se comporte plutôt comme un tenseur covariant d'ordre N et de terme :

$$T_{i_1, \dots, i_n} = R \cos(N\theta - (\sum_j i_j) \pi/2)$$

qui est symétrique puisque la somme ne dépend pas du choix d'un ordre pour les i_j . L'avantage majeur de cette construction est de pouvoir travailler avec des N -RoSys comme avec des vecteurs usuels en travaillant directement sur le vecteur de représentation. De cette façon, on peut définir des opérations telles que l'addition ou le mélange de différents champs, et donc de construire de nouveaux champs en combinant des champs de base. De plus, cette approche permet de relier la topologie des N -RoSys à celle (bien étudiée) des champs de vecteurs, ce qui fournit une caractérisation pour les singularités et les séparatrices.

Cependant, Palacios *et al.* commencent par générer un champ à partir de contraintes géométriques introduites par l'utilisateur sans contrôler la topologie du champ. Comme le théorème de Poincaré Hopf

Méthode	GDS	DTVf	N-RoSy	NSDF	GADF
Directions	oui ⁴	non	non	oui	oui ⁴
Symétries	oui	non	oui	oui	oui
Contrôle géométrique	oui	oui ¹	oui	oui ³	oui
Contrôle topologique	non	non ¹	oui ²	oui ²	oui ²

TAB. 6.1 – Comparaison des caractéristiques de quelques méthodes de manipulation de champs de directions.

montre qu'il est impossible d'introduire une singularité sans en introduire une autre d'indice opposé, leur interface est basée sur le déplacement et la simplification de paires de singularités générées par leur méthode. Comme illustré sur la Figure 6.3, ces opérations ne conduisent pas nécessairement à une simplification optimale de la topologie.

6.4 Conclusion

Une comparaison synthétique de ces méthodes, ainsi que de celles que l'on va décrire au Chapitre 9 et au Chapitre 10, est présentée dans la Table 6.1. Les remarques suivantes s'imposent sur cette comparaison :

- ¹ L'approche de "Design of Tangent Vector Field" (DTVf) assure le contrôle topologique seulement en absence de contraintes géométrique. En d'autres termes les contrôles géométriques et topologiques sont exclusifs.
- ² Dans "N-Rotational symmetry vector fields" (N-RoSy), la topologie est contrôlée *a posteriori* (après le lissage) contrairement à "N-symmetry direction fields" (NSDF) que nous présentons au Chapitre 9 et qui permet un contrôle topologique *a priori*. Dans le Chapitre 10 (GADF), la topologie est contrôlée plus faiblement en limitant l'apparition de singularités en fonction du niveau de détail.
- ³ Les contraintes géométriques ne peuvent être imposées qu'*a posteriori* en imposant une certaine rotation à une direction existante, donc cette approche ne peut pas être utilisée pour lisser un champ existant.
- ⁴ GDS (cf Section 6.3.1) et GADF (cf Chapitre 10) utilisent un lissage similaire où le champ est normalisé à chaque itération. Ils permettent donc aussi de lisser des champs de vecteurs en relaxant cette contrainte, bien que ce ne soit pas conseillé car la régularité des lignes de champs est alors moins bonne.

Toutes les méthodes présentées, et en particulier les trois que nous avons détaillées, donnent un certain contrôle géométrique sur les champs produits d'une façon plus ou moins directe. Cependant, elles ne prévoient pas de contrôle satisfaisant sur la topologie. Dans l'approche TVFD décrite dans Section 6.3.2, on contrôle la divergence et le rotationnel, ce qui ne garantit pas le placement des singularités en présence de contraintes géométriques. Dans l'approche N-RoSy décrite dans Section 6.3.3, la topologie est simplifiée *a posteriori* par le déplacement ou la simplification de paires de singularités. Ces opérations sont soit appliquées par l'utilisateur, soit automatisées afin réduire le nombre de singularités. En présence de hautes fréquences géométriques, un grand nombre de singularités sont produites si bien que l'approche manuelle n'est pas envisageable. L'approche automatique quand à elle ne garantit pas un nombre minimal de singularités.

Dans le chapitre suivant, nous allons proposer une étude en profondeur des champs de directions qui va faire apparaître certains concepts importants pour la manipulation des champs de directions, et en particulier concernant leur topologie. Sur la base de cette étude, nous allons proposer une nouvelle discrétisation pour les champs de directions qui va nous permettre de contrôler complètement leur topologie.

Les champs de directions à symétries et leur discrétisation

Sommaire

7.1	<i>N</i>-symmetry direction fields on surfaces with boundary	111
7.1.1	Surfaces with boundary	112
7.1.2	Curves and Cycles on \mathcal{S}	112
7.1.3	Direction field	114
7.1.4	Curvature	115
7.2	Turning numbers and their fundamental properties	115
7.2.1	Definition	115
7.2.2	Boundary property	116
7.2.3	Topological equivalence	119
7.3	Singularities	119
7.3.1	2D vector field Singularity Classification	120
7.3.2	2D <i>N</i> -symmetry direction field Singularities	121
7.3.3	<i>N</i> -symmetry direction field Singularities on surfaces	122
7.3.4	Poincaré Hopf theorem	122
7.4	Discretization of <i>N</i>-symmetry direction fields	123
7.4.1	<i>N</i> -symmetry directions on facets	124
7.4.2	Curvature and period jumps	125
7.4.3	Turning numbers	126
7.4.4	Singularity index	127
7.4.5	Converting vector fields into our representation	128
7.4.6	Conclusion	129

Introduction

Lorsque l'on s'intéresse aux applications des champs de directions, on s'aperçoit que ces applications utilisent des objets qui ont plus de symétries que de simples champs de directions, par exemple des objets invariants par rotation de π ou $\pi/2$ autour de la normale de la surface. Par exemple, si on conçoit une paire de champs de directions orthogonaux sur le cube arrondi de la figure 7.1(gauche) on aimerait connecter

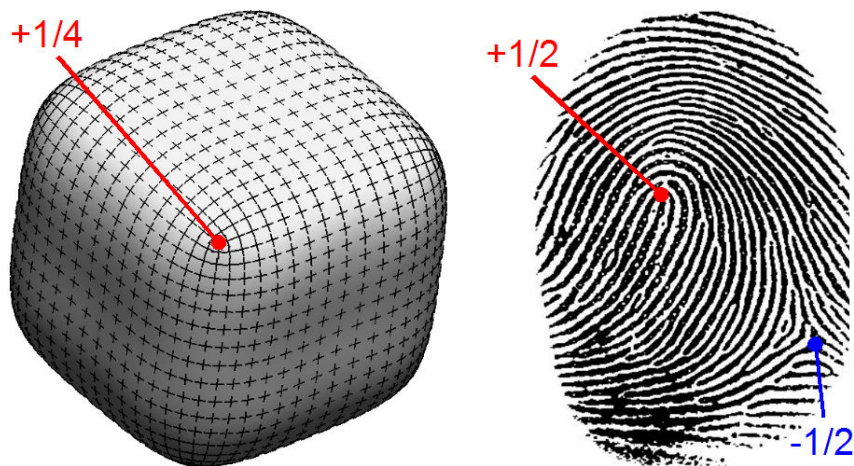


FIG. 7.1 – Exemples de singularités d'indice fractionnaire. **A gauche** : Singularité d'index $1/4$ d'un champs de croix naturel sur un cube arrondi. **A droite** : Singularités d'index $1/2$ et $-1/2$ on sur une empreinte digitale.

les deux champs autour des coins du cube après une rotation de $\pi/2$. Ce problème a été soulevé pour la première fois dans le domaine de l'informatique graphique par Hertzmann et Zorin dans un article sur le rendu non-photoréaliste [Hertzmann and Zorin 2000], où la notion de "champs de croix" est introduite intuitivement. Les auteurs mentionnent que la caractérisation de ces singularités (points où la direction ne peut être définie) et le contrôle de leur nombre et de leur type est un problème clef, impliquant des concepts mathématiques intéressants à découvrir, comme une généralisation de la formule d'Euler.

Une des contributions importante de cette thèse est de proposer une solution à ce problème. L'idée centrale est de se baser sur une représentation des champs de directions où le comportement du champ entre deux points de discrétisation est contrôlé par des entiers nommés "sauts de période". Ces sauts de période caractérisent non seulement les singularités du champs, mais aussi d'autres degrés de liberté topologiques plus complexes que nous détaillerons dans ce chapitre. L'autre idée est qu'en autorisant ces "sauts de période" à n'être non plus entiers mais multiples de $1/N$, on autorise des singularités d'ordre multiple de $1/N$ (cf Figure 7.1), et que l'on peut donc manipuler des champs de directions à symétrie de cette façon.

Ces directions à symétries peuvent se définir dans le cadre continu comme des ensembles de N directions invariants par symétrie de rotation de $2\pi/N$. Nous allons nous proposer d'étudier leur topologie, c'est à dire de définir les classes de champs de directions qui peuvent se déformer continument l'un en l'autre. Cela nous permettra de construire au Chapitre 9 un algorithme qui génère des champs de directions à symétrie lisses en contrôlant complètement leur topologie (nombre, type et placement des singularités), ainsi qu'un certain nombre de directions.

Plan du chapitre

On commencera par définir les champs de directions à N symétries dans la Section 7.1. Nous allons alors proposer un certain nombre d'outils théoriques pour les étudier. En particulier, on introduira dans la Section 7.2 la notion simple mais utile de *turning number* d'un champs le long d'un cycle, et nous nous en servirons pour donner une nouvelle définition des singularités d'un champ de directions. Nous donnerons alors deux théorèmes qui expliquent la topologie des champs de directions en se basant sur

les *turning numbers*. Dans la Section 7.3 nous proposerons une définition des singularités et de leurs indexes basée sur les *turning numbers*. Cela nous permettra une démonstration simple du théorème de Poincaré-Hopf (qui explique les relations entre les indices des singularités et la topologie de la surface) dans le cas des champs de directions à symétries sur les surfaces.

Enfin, nous exposerons dans Section 7.4 une structure de données pour représenter les champs de directions à symétries qui permet de contrôler leur comportement entre les points de discrétisation par des entiers appelés "sauts de périodes". Nous montrerons que ces "sauts de périodes" sont nécessaires pour caractériser la topologie du champs.

Contributions

- Ce chapitre présente une étude détaillée de la topologie des champs de directions à symétrie, et en particulier introduit un outil théorique, le *turning number* pour généraliser la notion d'*index* de singularité (cf Section 7.2 et Section 7.3);
- Nous prouvons un analogue du théorème de Poincaré-Hopf basé sur ces *turning numbers* en l'adaptant aux champs de directions à symétries, et en le généralisant aux variétés à bords ;
- Nous prouvons aussi un théorème reliant les *turning numbers* à la topologie du champ de directions. En se basant sur ces deux théorèmes, on explique complètement les liens entre la topologie d'une surface et la topologie d'un champ de directions défini sur elle ;
- Nous présentons une structure discrète basée sur les sauts de périodes pour représenter les champs de directions à symétries (Section 7.4).
- En autorisant les sauts de périodes à être multiples de $1/N$, on pourra manipuler des champs de directions à symétries et donc des singularités d'indices fractionnaires (cf Figure 7.1) de façon très directe.

7.1 *N*-symmetry direction fields on surfaces with boundary

This section presents the fundamental tools for studying direction fields defined over surfaces with boundary, and especially to study their *topology*. Topology is the study of properties which are invariant by continuous deformations (without cutting or gluing anything), called homotopies. In other words, topology tries to answer the question : under what condition are two objects homotopic (i.e. can be continuously transformed one into another) ? For oriented surfaces with boundary, the answer is that they need to have the same genus g (number of handles) and number of borders (connected components of the boundary) b . In other terms, if \mathcal{S}_1 and \mathcal{S}_2 are two surfaces with boundary, we have $\mathcal{S}_1 \equiv_t \mathcal{S}_2 \Leftrightarrow g(\mathcal{S}_1) = g(\mathcal{S}_2)$ and $b(\mathcal{S}_1) = b(\mathcal{S}_2)$, where \equiv_t denotes the homotopic equivalence (t stands for topology). What is even more interesting is the structure of the set of homotopy classes (classes of all objects with same topology). For oriented surfaces with boundary again, this set is isomorphic to \mathbb{N}^2 , since any pair of non negative integers (g, b) can be associated to the class of all surfaces with genus g and b borders. For this reason, we say that these two integers are the two *topological degrees of freedom* (TDoF) of surfaces. This section addresses the same questions for *N*-symmetry direction fields defined over a 2-manifold, which means we aim at studying at which conditions are two *N*-symmetry direction fields homotopic (see Figure 7.2). In other words, we want to exhibit the TDoF of these direction fields. To answer these questions, we will introduce the concept of turning numbers of cycles in direction fields. As we will show, the TDoF of a direction field are the turning numbers of a homology basis of cycles, such that these turning numbers hold all the field topology. Moreover, turning numbers generalize singularities which alone do not control all of the field topology.



FIG. 7.2 – Homotopy for fields : the middle field is homotopic to the one on the right but not to the one on the left (they can not be transformed continuously one into another)

The intuitive idea behind turning numbers is that when following a closed cycle, a N -symmetry direction might do an arbitrary number of N^{th} of turns before coming back to its starting point. Imagine you are traveling on earth along a cycle with a compass giving you the north direction, then you can count the number of turns of the compass while following the cycle. If you turn around a tree (there are no trees at earth poles...) you will get 1 turn, but if you follow the equator or turned around a pole, you will get 0 turn. We can do the same on any surface, and with any direction field defined over it, and we call this quantity the “turning number” of the field along the cycle (it depends on both the cycle and the direction field). We will show that turning numbers capture the topology of the direction field, because two direction fields are homotopic if and only if they have the same turning numbers along the $2g + b - 1$ cycles of a homology basis (a basis for cycles on a 2 manifold). This shows that homotopy classes of direction fields are isomorphic to \mathbb{Z}^{2g+b-1} , or in other words that the topology of a direction field is entirely defined by $2g + b - 1$ integers, which are the TDoF (topological degrees of freedom) of direction fields.

In this section, we will first provide the reader with the definitions of the objects that we will be handling throughout this chapter : surfaces (Section 7.1.1), curves and cycles (Section 7.1.2) and direction fields (Section 7.1.3). Then we define the curvature for cycles and direction fields. We use these curvatures to define formally turning numbers, and exhibit their fundamental properties. We finally explain how turning numbers relate to field singularities.

7.1.1 Surfaces with boundary

In this chapter we call surface (or 2-manifold) \mathcal{S} a topological space where each point has a neighborhood homeomorphic to the plane or half plane. The surface \mathcal{S} considered here is compact, connected and oriented, so that each point has a unique unit normal vector \mathbf{n} . \mathcal{S} has g handles and a boundary $\partial\mathcal{S}$ which is the collection of its b borders, for some (non-negative) integers g called the *genus* and b called the *number of borders* of \mathcal{S} .

7.1.2 Curves and Cycles on \mathcal{S}

We will call *curve* γ on \mathcal{S} an oriented 1-manifold embedded in \mathcal{S} ($\gamma \subset \mathcal{S}$). Because γ is oriented, it has a unique tangent vector \mathbf{t}_γ in each point which is also tangent to \mathcal{S} . Using this tangent vector, along with the surface normal \mathbf{n} , we can define a unique conormal vector $\mathbf{n}_\gamma = \mathbf{n} \times \mathbf{t}_\gamma$ on the curve, which ensures that $(\mathbf{t}_\gamma, \mathbf{n}_\gamma, \mathbf{n})$ form a natural local orthonormal basis called the *Darboux frame* (see Figure 7.3 left). Notice that curves are not necessarily connected, so the term "set of curves" would be more appropriate (but more cumbersome). We define the following notions on curves (see Figure 7.4 left) :

- The number of connected components of a cycle γ is called its cardinal, and denoted $|\gamma|$.

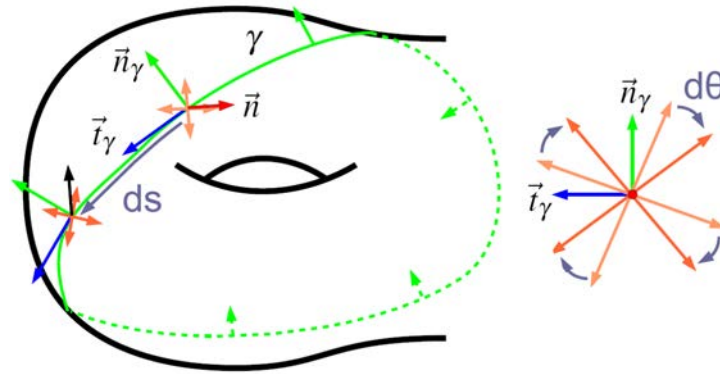


FIG. 7.3 – A Darboux frame on a chain (γ , green) consists of the tangent (\mathbf{t}_γ , red), conormal (\mathbf{n}_γ , blue) and normal (\mathbf{n} , red). When moving along an infinitesimal portion ds of γ , the direction turns of an infinitesimal angle $d\theta$ in the Darboux Frame. The *curvature* of the direction field along γ can then be defined as $\kappa_{\vec{d}}(\gamma) = d\theta/ds$

- A connected cycle γ is called a *single cycle*. It satisfies $|\gamma| = 1$.
- The *tangent* to γ is the vector $\mathbf{t}_\gamma = \mathbf{n} \times \mathbf{n}_\gamma$. Its orientation depends both on the orientations of the surface and cycle. In the following, integrals along cycles will always be done in the direction of the tangent.
- The *reversal* $-\gamma$ of a curve γ is the curve with opposite orientation : $\mathbf{t}_{-\gamma} = -\mathbf{t}_\gamma$. We use the notations $\gamma_0 + \gamma_1$ and $\gamma_0 - \gamma_1$ in place of $\gamma_0 \cup \gamma_1$ and $\gamma_0 \cup -\gamma_1$ as it is more practical to handle unions of cycles with various orientations.
- A *cycle* on \mathcal{S} is a curve without endpoints ($\partial\gamma = 0$).
- We call ∂ the boundary operator, such that $\partial\mathcal{S}$ is the subset of points of \mathcal{S} with neighborhood homeomorphic to the half plane. This subset is a cycle for which we can choose an orientation by requiring its conormal to point outwards \mathcal{S} (this is why we represent the conormal instead of tangent in our figures).
- A curve γ is called *exact* if there exists a sub-manifold S of \mathcal{S} such that $\gamma = \partial S$. An exact curve is necessarily a cycle (a boundary cannot have endpoints).
- Two cycles are *homotopic* if one can be continuously deformed into another. More formally, $\gamma_0 \equiv_t \gamma_1 \Leftrightarrow$ there exists a continuous function $\Gamma : [0, 1] \rightarrow \mathcal{C}(\mathcal{S})$ such that $\Gamma(0) = \gamma_0$, $\Gamma(1) = \gamma_1$ ($\mathcal{C}(\mathcal{S})$ is the set of all cycles on \mathcal{S}).

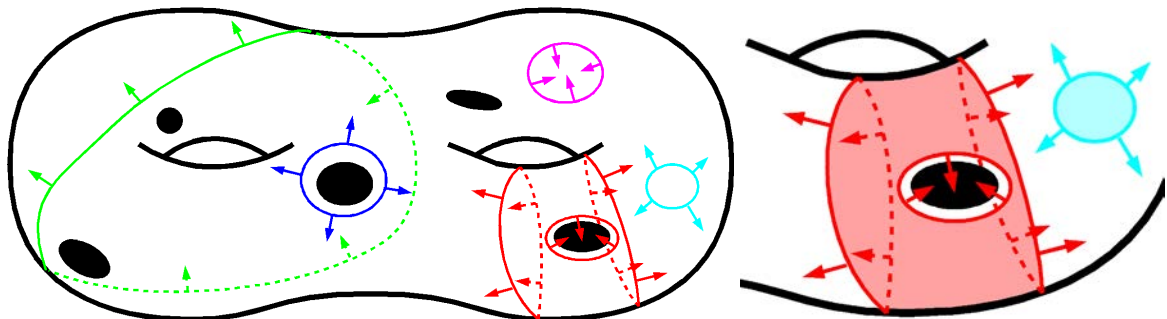


FIG. 7.4 – **Left** : Cycles on a surface with boundary (black areas are holes). Only the cycles shown on **Right** are exact because they are the boundaries (boundary with conormal pointing outwards) of submanifolds of \mathcal{S} (in light color)

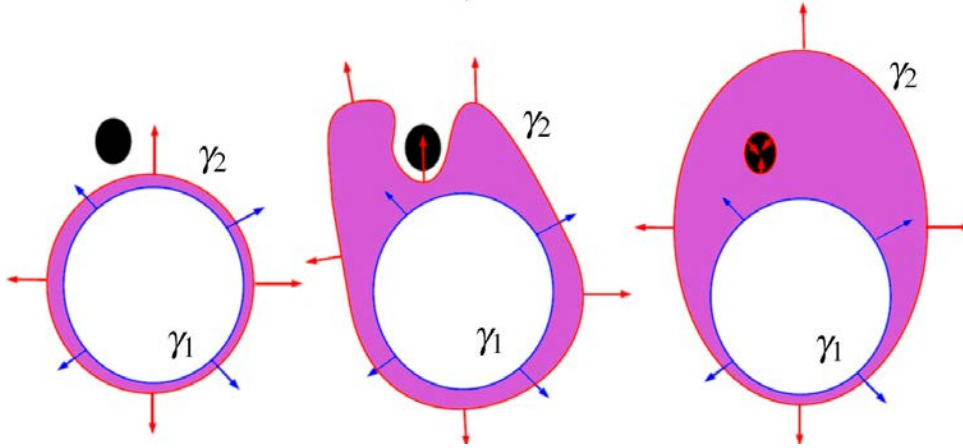


FIG. 7.5 – γ_2 stays homological to γ_1 because $\gamma_2 - \gamma_1$ is exact(it is the boundary of a 2-manifold). Note that homological cycles might have a different number of connected components.

- A cycle γ is called contractible if it is the boundary of a topological disk D ($\gamma = \partial D$). This definition of contractibility for cycles adds a notion of orientation to its definition for loops (homotopic to a null loop) as the reversal of a contractible boundary is not necessarily contractible.
- Two cycles γ_0 and γ_1 are *homological* iff $\gamma_0 - \gamma_1$ is exact (see Figure 7.5). We will denote \equiv_l the homologic equivalence (for cycles), when \equiv_s is the homotopic equivalence (for surfaces and direction fields). More formally, the homology for cycles is defined as the quotient set of closed curves (cycles) over exact curves. In particular the 0 of homology is the class of exact cycles, so γ is exact $\Leftrightarrow \gamma \equiv_l \emptyset$
- $H(\mathcal{S}) = \{\gamma_i^H\}_{i=1..n}$ is called a homology basis on \mathcal{S} if the basis cycles γ_i^H are independent (there are no $a \in \mathbb{Z}^n$ such that $\sum a_i \gamma_i^H \equiv_l \emptyset$), and if any cycle γ on \mathcal{S} is homological to a formal sum of the basis cycles ($\exists a \in \mathbb{Z}^n$ such that $\gamma \equiv_l \sum_{i=1}^n a_i \gamma_i^H$). An important result from manifold topology is that an homology basis on \mathcal{S} has $2g + b - 1$ basis vectors, where g is the genus, and b the number of borders of \mathcal{S} .

Homology is more flexible than homotopy : homology allows a cycle to split into two or two cycles to merge into one (see Figure 7.5), whereas homotopy does not.

7.1.3 Direction field

A unit tangent vector \mathbf{d} on \mathcal{S} is a vector that satisfies $\|\mathbf{d}\| = 1$ and $\mathbf{d} \cdot \mathbf{n} = 0$. We call N -symmetry direction on \mathcal{S} a set of N unit tangent vectors on \mathcal{S} invariant by rotation of $2\pi/N$ around the normal. Hence, based on a unit tangent vector \mathbf{d}_0 we can build a N -symmetry direction $\vec{d} = \{\mathbf{d}_k = R(\mathbf{d}_0, 2k\pi/N) | k \in \mathbb{Z}\}$ where $R(\cdot, \theta)$ is the rotation of angle θ in the tangent plane $T\mathcal{S}$. We call direction field \vec{d} on \mathcal{S} a mapping which associates a N -symmetry direction $\vec{d}(\mathbf{x})$ to each point $\mathbf{x} \in \mathcal{S}$. In the following, we will omit the term N -symmetry for brevity. We will call $\mathcal{D}_N(\mathcal{S})$ the set of derivable direction fields, which we define as the direction fields with bounded curvature (see definition below).

Two direction fields $\vec{d}_0, \vec{d}_1 \in \mathcal{D}_N(\mathcal{S})$ are called homotopic if there exists a continuous function $\Gamma : [0, 1] \rightarrow \mathcal{D}_N(\mathcal{S})$ such that $\Gamma(0) = \vec{d}_0, \Gamma(1) = \vec{d}_1$. In other terms, two direction fields are homotopic (or have the same topology) if they can be continuously transformed one into another. As this is a relation of equivalence, we can define homotopy classes as the sets of all direction fields equivalent to a given one. Based on the intuition of what is a direction field in the plane, it would seem that all direction fields

are homotopic, such that there is only one homotopy class. However, if the direction field is defined on a surface of arbitrary topology, we will show it is not the case.

Homotopy classes of direction fields can be characterized by the *turning numbers* of the field along some cycles. They correspond intuitively to the number of times the direction turns in a local Darboux frame while moving along the cycle. We are now going to define the curvature of both cycles and direction fields, which will be used for a rigorous definition of the turning number.

7.1.4 Curvature

The curvature of a curve or a surface expresses the angular variation of their (unit) tangent. The derivative of a unit vector is an orthogonal vector, so it is defined uniquely by its projection on this orthogonal direction. If we call s the arclength on a curve γ , we can define the curvature of γ using the decomposition :

$$\frac{\partial \mathbf{t}_\gamma}{\partial s} = \kappa_\gamma \mathbf{n}_\gamma + \kappa_{\mathcal{S}} \mathbf{n} \quad \kappa_{\mathcal{S}} = \frac{\partial \mathbf{t}_\gamma}{\partial s} \cdot \mathbf{n} ds \quad \kappa_\gamma = \frac{\partial \mathbf{t}_\gamma}{\partial s} \cdot \mathbf{n}_\gamma \quad (7.1)$$

- $\kappa_{\mathcal{S}}$ measures the normal curvature of \mathcal{S} in direction \mathbf{t}_γ .
- κ_γ measures the curvature of γ in the tangent plane of \mathcal{S} .
- $\partial \mathbf{t}_\gamma / \partial s \cdot \mathbf{t}_\gamma = 0$ by derivation of $\mathbf{t}_\gamma \cdot \mathbf{t}_\gamma = 1$ (\mathbf{t}_γ is a unit vector)

Boundaries verify the Gauss-Bonnet formula :

$$\oint_{\partial S} \kappa_\gamma ds = 2\pi \chi(S) - \int_S K dA \quad (7.2)$$

Where K is the Gaussian curvature, and $\chi(S)$ the Euler characteristic of S .

We can similarly define the curvature $\kappa_{\vec{d}}(\mathbf{t}_\gamma)$ for the direction field $\vec{d} = \{\mathbf{d}_k\}$ in direction \mathbf{t}_γ as :

$$\kappa_{\vec{d}}(\mathbf{t}_\gamma) = \frac{\partial \mathbf{d}_k}{\partial s} \cdot \mathbf{d}_k^\perp \quad (7.3)$$

where $\mathbf{d}_k^\perp = \mathbf{n} \times \mathbf{d}_k$ is a unit vector orthogonal to \mathbf{d}_k in the tangent plane, such that $(\mathbf{d}_k, \mathbf{d}_k^\perp, \mathbf{n})$ is an orthonormal basis (it is the Darboux frame of the streamlines of \mathbf{d}_k). The curvature $\kappa_{\vec{d}}$ is the same for all $k \in \mathbb{N}$ so it can be called the curvature of \vec{d} in direction \mathbf{t}_γ .

$\kappa_{\vec{d}} ds$ is a 1-forms, or an object to be integrated on curves. This is exactly what we are going to do to define the turning numbers characterizing homotopy classes of direction fields. From this point on, we will assume that our direction fields are continuous, such that their curvature is defined almost everywhere.

7.2 Turning numbers and their fundamental properties

Based on direction field curvature, we are now going to define the notion of turning number. We will then prove two important theorems involving them.

7.2.1 Definition

The turning numbers of a direction field along a cycle corresponds to the number of rotations of the field along this cycle, and are characteristic of homotopy classes of direction fields, hence of their topology.

Let γ_{01} be a curve with endpoints \mathbf{x}_0 and \mathbf{x}_1 , and $\theta_\gamma(\mathbf{x}_0)$ be an angle (in \mathbb{R}) between $\vec{d}(\mathbf{x}_0)$ and $\mathbf{t}_\gamma(\mathbf{x}_0)$. Then by definition of the curvature :

$$\theta_\gamma(\mathbf{x}_1) = \theta_\gamma(\mathbf{x}_0) + \int_{\gamma_{01}} (\kappa_{\vec{d}} - \kappa_\gamma) ds \quad (7.4)$$

is an angle (in \mathbb{R}) between $\vec{d}(\mathbf{x}_1)$ and $\mathbf{t}_\gamma(\mathbf{x}_1)$. As \vec{d} is an N -symmetry direction field, we can define an equivalence for angles : $\theta =_N \theta' \leftrightarrow \exists k \in \mathbb{Z}$ such that $\theta' - \theta = 2k\pi/N$. Especially, for a cycle γ , we have $\mathbf{x}_0 = \mathbf{x}_1$, such that the quantity

$$T_{\vec{d}}(\gamma) = \frac{1}{2\pi} \oint_\gamma (\kappa_{\vec{d}} - \kappa_\gamma) ds \quad (7.5)$$

is always an integer multiple of $1/N$ which we call the *turning number* of \vec{d} along γ , and corresponds intuitively to the number of N^{th} of turns \vec{d} does in the Darboux frame along γ (see Figure 7.7).

In the case $T_{\vec{d}}(\gamma) = 0$, the 1-form $(\kappa_{\vec{d}} - \kappa_\gamma)ds$ is called exact on γ , which means that there exists a 0-form (scalar field) θ on γ such that $(\kappa_{\vec{d}} - \kappa_\gamma)ds = d\theta$. θ is defined up to a constant which can be chosen such that θ is an angle between \mathbf{t}_γ and \vec{d} . This shows that only when the turning number of \vec{d} is 0 along γ can the angle θ be represented as a continuous real function (else we need $2\pi/N$ jumps).

Turning numbers have two fundamental properties which make them useful for studying direction field topology :

Théorème 7.2.1 (Boundary turning number) *Let S be a surface (2-manifold with boundary) embedded in \mathbb{R}^3 , then :*

$$T_{\vec{d}}(\partial S) = -\chi(S) \quad (7.6)$$

where $\chi(S) = 2 - 2g(S) - b(S)$ is the Euler characteristic of S .

Théorème 7.2.1 (Boundary turning number) is equivalent to the Poincaré Hopf theorem with a proper definition for the index of a singularity, which will be developed in next subsection. It links the turning numbers of homological cycles as $\gamma_1 \equiv_l \gamma_2 \leftrightarrow \exists S \gamma_1 - \gamma_2 = \partial S$

Théorème 7.2.2 (Topological equivalence) *Two direction fields defined over a surface \mathcal{S} are homotopic if and only if they have the same turning numbers along the cycles of an homology basis $H(\mathcal{S})$ of \mathcal{S} :*

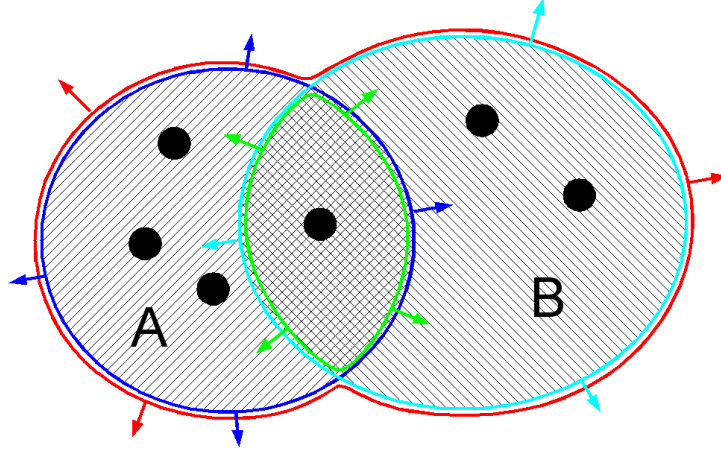
$$\vec{d}_1 \equiv_t \vec{d}_2 \Leftrightarrow \forall \gamma \in H(\mathcal{S}), T_{\vec{d}_1}(\gamma) = T_{\vec{d}_2}(\gamma)$$

Théorème 7.2.2 shows that direction fields, have homotopy classes isomorphic to \mathbb{Z}^{2g+b-1} . This comes from the fact that the $2g$ so-called generators of the surface and $b - 1$ surface borders define a homology basis and that all basis have the same number of cycles. Hence a direction field has $2g + b - 1 = 1 - \chi(\mathcal{S})$ topological degrees of freedom (TDoF) on a genus g surface with b borders. In the case $\chi(\mathcal{S}) = 2$ (\mathcal{S} is a topological sphere), this number is negative because it is impossible to build a continuous direction field without introducing at least a singularity (a hole), which brings $\chi(\mathcal{S})$ to 1.

We are now going to prove these two theorems, then explain how turning numbers relate to direction field singularities.

7.2.2 Boundary property

The boundary property states that for any direction field on a 2-manifold S , we have $T(\partial S) + \chi(S) = 0$ which generalizes the Poincaré Hopf theorem to N -symmetry direction fields. We first establish three simple results, then prove the equation by structural induction.


 FIG. 7.6 – Equivalence between $\partial A \cup \partial B$ and $\partial(A \cup B) \cup \partial(A \cap B)$

Lemme 7.2.3 *The reversal of a cycle has opposite turning number :*

$$T(-\gamma) = -T(\gamma) \quad (7.7)$$

Proof : integrating in opposite direction changes ds in $-ds$ \square

Lemme 7.2.4 *If we call A and B two sub-manifolds of \mathcal{S}_h (see Figure 7.6), the turning numbers of ∂A and ∂B are linked by the equation :*

$$T(\partial A) + T(\partial B) = T(\partial(A \cup B)) + T(\partial(A \cap B)) \quad (7.8)$$

Proof : The term in $\kappa_{\vec{d}}$ is preserved because integrated on the same set $\partial A \cup \partial B = \partial(A \cup B) \cup \partial(A \cap B)$. The term in κ_{γ} , is preserved by application of the Gauss-Bonnet formula to the equality $\int_A + \int_B = \int_{A \cup B} + \int_{A \cap B}$. \square

Lemme 7.2.5 *Boundaries of topological disks have a turning number of -1.*

Proof : Let γ^c be the boundary of a topological disk D . By definition, there exists a continuous bijective mapping $m : D \rightarrow \mathcal{D}_2$ where \mathcal{D}_2 is the unit disk in \mathbb{R}^2 . Using this application, we can define the image field $\vec{d}' = m(\vec{d})$ of the direction field on \mathcal{D}_2 .

$$T_{\vec{d}'}(\partial \mathcal{D}_2) = \frac{1}{2\pi} \oint_{\partial \mathcal{D}_2} (\kappa_{\vec{d}'} - \kappa_{\gamma}) ds = \frac{1}{2\pi} \int_{s=0}^{2\pi} \left(\frac{\partial \vec{d}'}{\partial s} \cdot \vec{d}'^{\perp} - \frac{\partial \mathbf{t}_{\gamma}}{\partial s} \cdot \mathbf{n}_{\gamma} \right) ds$$

Because we are on the unit disk, the arclength s is equivalent to the angle on the disk boundary, so we have $\mathbf{t}_{\gamma} = (\cos(s), \sin(s))^T$ and $\mathbf{n}_{\gamma} = (-\sin(s), \cos(s))^T$, which gives :

$$\kappa_{\gamma} = \frac{\partial \mathbf{t}_{\gamma}}{\partial s} \cdot \mathbf{n}_{\gamma} = 1$$

Let $\mathbf{x} = (x = \cos(s), y = \sin(s))^T$ be the position vector on the disk boundary. We have :

$$\kappa_{\vec{d}'} ds = \frac{\partial \vec{d}'}{\partial s} \cdot \vec{d}'^{\perp} ds = \left(\frac{\partial \vec{d}'}{\partial x} \cdot \vec{d}'^{\perp}, \frac{\partial \vec{d}'}{\partial y} \cdot \vec{d}'^{\perp} \right)^T d\mathbf{x}$$

we can verify that $\kappa_{\vec{d}} ds$ is closed, so it is exact on a topological disk ($\exists \theta$ such that $\kappa_{\vec{d}} ds = d\theta$). We can now compute the turning number :

$$T_{\vec{d}'}(\partial \mathcal{D}_2) = T_{\vec{d}_p}(\partial \mathcal{D}_2) = \frac{1}{2\pi} \oint_{\partial \mathcal{D}_2} (\kappa_{\vec{d}'} - \kappa_{\gamma}) ds = \frac{1}{2\pi} \oint_{\partial \mathcal{D}_2} d\theta - \frac{1}{2\pi} \int_{s=0}^{2\pi} 1 \cdot ds = -1 \square$$

These lemmas allow us to give a proof of Théorème 7.2.1 :

$$T_{\vec{d}}(\partial S) = -\chi(S) \quad (7.9)$$

Proof : It is well known in topology that any orientable surface (with boundary) can be cut along g cycles to obtain a sphere (with boundary). Hence we start by proving (7.9) for a sphere with $b > 0$ borders.

If S is a topological disk ($b = 1, \chi = 1$), its boundary is contractible so its turning number is -1 by Lemma 7.2.5 which proves (7.9) for $b = 1$.

If S is a topological cylinder ($b = 2, \chi = 0$), its boundary is composed of 2 borders : $\partial S = \gamma_1 + \gamma_2$. γ_1 and $-\gamma_2$ are trivially homotopic so they have the same turning numbers, so by (7.7) we have :

$$T(\partial S) = T(\gamma_1) + T(\gamma_2) = T(\gamma_1) - T(-\gamma_2) = 0$$

which proves (7.9) for $b = 2$.

For higher numbers of borders, we prove the property recursively : assume (7.9) is true $\forall b \leq B$, and let S have $B + 1$ borders $\gamma_1 \dots \gamma_{B+1}$. Let A_B and A_{B+1} be two connected submanifolds which boundary contains γ_B and γ_{B+1} (respectively) and intersect in a topological disk. $S \setminus \{A_B \cup A_{B+1}\}$ has at most B borders, so it verifies (7.9) :

$$\begin{aligned} T(\partial(S \setminus \{A_B \cup A_{B+1}\})) &= \sum_{i=1}^{B-1} T(\gamma_i) + T(\partial\{A_B \cup A_{B+1}\}) = \\ &= \sum_{i=1}^{B-1} T(\gamma_i) + T(\partial A_B) + T(\partial A_{B+1}) - T(\partial\{A_B \cap A_{B+1}\}) = \\ &= \sum_{i=1}^{B+1} T(\gamma_i) + 1 = \chi(S \setminus \{A_B \cup A_{B+1}\}) = -\chi(S) + 1 \end{aligned}$$

which proves (7.9) recursively.

We can now get to the most general case where S has genus g and b borders $\gamma_1 \dots \gamma_b$. By definition of the genus, there exists a family $\gamma_1^G, \dots, \gamma_g^G$ of single cycles such that $S \setminus \{\gamma_1^G, \dots, \gamma_g^G\}$ is connected. This operation does not change the Euler characteristic of the surface ($b \leftarrow b + 2g, g \leftarrow 0$ preserves $\chi = 2 - 2g - b$). The result is a sphere S_{cut} with $b + 2g$ borders $\gamma_1, \dots, \gamma_b, \gamma_1^G, -\gamma_1^G, \dots, \gamma_g^G, -\gamma_g^G$, so we can apply (7.9) :

$$T(\partial S_{cut}) = \sum_{i=1}^n T(\gamma_i) + \sum_{i=1}^g T(\gamma_i^G) + T(-\gamma_i^G) = 2 - 2g - b$$

as the second sum is null by (7.7). This finally proves Theorem 7.2.1 in the general case \square

7.2.3 Topological equivalence

We will need a simple lemma to prove the topological equivalence :

Lemme 7.2.6

$$T_{\vec{d}_1}(\gamma) = T_{\vec{d}_2}(\gamma) \quad \forall \text{ cycle } \gamma \subset \mathcal{S} \Leftrightarrow T_{\vec{d}_1}(\gamma) = T_{\vec{d}_2}(\gamma) \quad \forall \gamma \in H(\mathcal{S})$$

where $H(\mathcal{S})$ is a homology basis on \mathcal{S} . In other words, the turning numbers of all cycles on \mathcal{S} depend only on the turning numbers of a homology basis.

Proof : By definition of homology basis, any cycle γ is homological to a cycle $\sum_i a_i \gamma_i^B$ where $\gamma_i^B \in H(\mathcal{S})$ are basis cycles, such that $\sum_i a_i \gamma_i^B - \gamma$ is a boundary. Hence, by Théorème 7.2.1, we have :

$$T_{\vec{d}_0}(\gamma) = \sum_i a_i T_{\vec{d}_0}(\gamma_i^B) + \chi(\mathcal{S}) = \sum_i a_i T_{\vec{d}_1}(\gamma_i^B) + \chi(\mathcal{S}) = T_{\vec{d}_1}(\gamma)$$

because the turning numbers are equal along the cycles of the homology basis. \square

We can now prove the Theorem 7.2.2 which states that two direction fields are homotopic if and only if they have the same turning numbers along the cycles of a homology basis of \mathcal{S} .

Proof : The direct implication is immediate because turning numbers are preserved by homotopy. For the converse, let \vec{d}_0 and \vec{d}_1 be the two fields with the same turning numbers along the cycles of a homotopy basis. By Lemma 7.2.6, \vec{d}_0 and \vec{d}_1 have the same turning numbers along any cycle γ :

$$T_{\vec{d}_0}(\gamma) - T_{\vec{d}_1}(\gamma) = \oint_{\gamma} \kappa_{\gamma} - \kappa_{\vec{d}_0} - \oint_{\gamma} \kappa_{\gamma} - \kappa_{\vec{d}_1} = \oint_{\gamma} \kappa_{\vec{d}_1} - \kappa_{\vec{d}_0} = 0$$

In terms of exterior calculus, this means that the one-form (linear application) $\kappa_{\vec{d}_1} - \kappa_{\vec{d}_0}$ is exact, so there exists a 0-form (scalar field) θ such that $\kappa_{\vec{d}_1} - \kappa_{\vec{d}_0} = d\theta$. As θ is defined up to a constant, we can chose it to match the angle (in $[0, 2\pi)$ for instance) between \vec{d}_0 and \vec{d}_1 . By definition, θ verifies $\vec{d}_1 = R(\vec{d}_0, \theta)$ where R is the rotation around the surface normal. This allows us to build a continuous function $D : [0, 1] \rightarrow \mathcal{D}_N(\mathcal{S})$ defined by $D(t) = R(\vec{d}_1, t\theta)$, such that $D(0) = \vec{d}_0$ and $D(1) = \vec{d}_1$ which proves that \vec{d}_0 and \vec{d}_1 are homotopic. \square

With these two theorems, one can exhibit, understand and control all the TDoF of a direction field.

Finally, note that since the turning number takes only discrete value, and that its definition makes it continuous with respect to continuous transforms of both the field and the cycle, it is invariant by homotopy of the field and cycle.

These two theorems show that the entire direction field topology is given by the turning numbers along the cycles of a homology basis. Introducing a puncture in the surface adds a cycle to the homology basis, thus a new topological degree of freedom. We will now show that such a puncture corresponds to the usual notion of field singularity, such that singularities can be seen as the limit case of a boundary whose length tends toward zero.

7.3 Singularities

Vector field singularities are important vector field topology features, that have an important visual impact. Non degenerate singularities cannot be smoothed out, and the Poincaré-Hopf theorem implies that

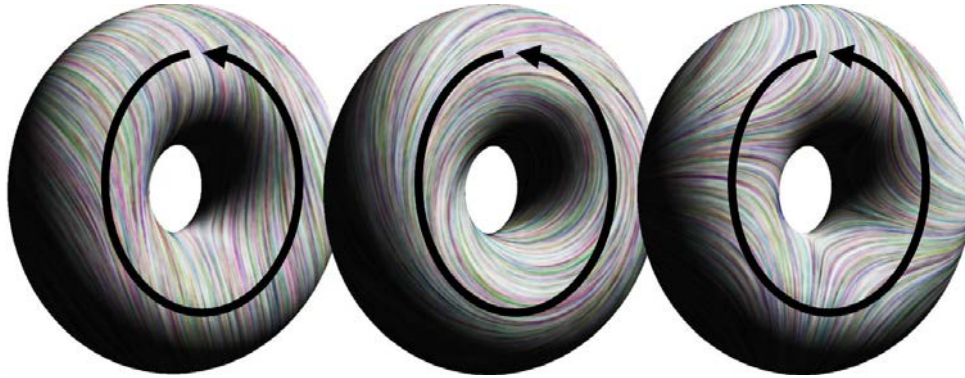


FIG. 7.7 – The turning number associated to a generator defines topology that cannot be captured by singularity indexes. The difference of topology of these direction fields without any singular points is defined by the turning numbers of the generator (black cycle) which are respectively -1, 0 and 1

they necessarily appear (except on genus 1 surfaces). Thus it is important to understand and control them. In this section, we will start by presenting their usual classification through a singularity index. We will start by the classical 2D definition of the index, then explain how this definition extends to higher order singularities and to singularities of vector field defined on a surface. We will see that singularity index does not depend on the vector norm, such that it is enough to work with direction fields. We will also see that N -symmetry direction fields allow for a wider class of singularity with fractional indexes. Finally, we will exhibit the relations between singularity indexes and turning numbers, that will allow for a very simple demonstration of the Poincaré-Hopf theorem in the setting of N -symmetry direction fields, seen as a simple consequence of Théorème 7.2.1.

7.3.1 2D vector field Singularity Classification

Linear singularities

Let $\mathbf{v} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ be a vector field. It is usually assumed that the zero set of $\mathbf{v} : \{x \in \mathbb{R}^2 | \mathbf{v}(x) = 0\}$ consists of a finite number of distinct points \mathbf{p}_i which are called the singularities of \mathbf{v} . Historically, the first classification of singularities was done using the Jacobian matrix J of \mathbf{v} :

$$J(\mathbf{v}) = \begin{pmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} \end{pmatrix} \quad (7.10)$$

The singularities are then classified according to the two eigenvalues [Tricoche 2002] of J at the critical points \mathbf{p}_i . If one of them is zero, \mathbf{p}_i is called a *degenerate* singularity, which means that it can be removed without creating another singularity. If they are non-zero, \mathbf{p}_i is called a *linear* or *first-order* singularity :

- If the eigenvalues are real with opposite signs, \mathbf{p}_i is called a *saddle*
- If the eigenvalues have negative real parts, \mathbf{p}_i is called a *sink*
- If the eigenvalues have positive real parts, \mathbf{p}_i is called a *source*
- If the eigenvalues are purely imaginary (and conjugate), \mathbf{p}_i is called a *vortex* or a *center*.

A finer classification can be made according to the imaginary parts in the source and sink cases.

Index and Higher order singularities

The Jacobian matrix is not always well behaved around \mathbf{p}_i , and does not allow to classify higher order singularities. For instance, a dipole generated by a magnetic field has a singularity which “looks” different from the singularities classified above. To classify a wider class of singularities, the *index* of a singularity was then defined as :

$$I_{\vec{d}}(\mathbf{p}_i) = \frac{1}{2\pi} \int_{\partial\Omega(\mathbf{p}_i)} \frac{v_x dv_y - v_y dv_x}{v_x^2 + v_y^2} = \frac{1}{2\pi} \int_{\partial\Omega(\mathbf{p}_i)} d\theta \quad (7.11)$$

where $\Omega(\mathbf{p}_i)$ is a small neighborhood of \mathbf{p}_i containing no other singularity, and θ is the angle between the direction \mathbf{v} and a reference vector, such that $v_x = r \cos \theta$ and $v_y = r \sin \theta$. The integral is always an integer multiple of 2π , such that the index of a singularity is always an integer :

- $I_{\vec{d}}(\mathbf{p}_i) = -1$ corresponds to a saddle.
- $I_{\vec{d}}(\mathbf{p}_i) = 0$ corresponds to a degenerate singularity.
- $I_{\vec{d}}(\mathbf{p}_i) = 1$ corresponds to all kinds of sources, sinks and centers.

This definition is more general as we can define so called *non-linear* or *higher-order* singularities which have indexes $|I_{\vec{d}}(\mathbf{p}_i)| > 1$. For instance, a dipole corresponds to $I_{\vec{d}}(\mathbf{p}_i) = 2$, a monkey saddle (saddle with 6 branches instead of 4) to $I_{\vec{d}}(\mathbf{p}_i) = -2$. Such higher-order singularities cannot be classified by the Jacobian, because computing the Jacobian at \mathbf{p}_i leads to making a linear approximation of the field at \mathbf{p}_i . This also explains why classical linear interpolations do not allow to represent higher order singularities. In contrast, our framework will allow such higher order singularities to be represented very easily.

7.3.2 2D N -symmetry direction field Singularities

For many applications [[Hertzmann and Zorin 2000](#), [Wei and Levoy 2001](#), [Ray et al. 2006](#)], we are interested in directions (we do not care about the norm), and we also want to allow switching or inverting directions. We have introduced N -symmetry direction fields as a general framework to handle such generalized fields. We will now see how vector field singularity classification extends to these N -symmetry direction fields, and in particular tackle normalization and symmetry issues.

Normalization issues

For (1-symmetry) direction fields, the integral in (7.11) is still well defined, which allows for a straightforward definition of direction field singularities. More than that, this even shows that vector field singularities only depend on the vector direction and not on its norm. This alone justifies studying direction field topology, and in particular separating vector norm and direction instead of separating its components. The only point is that vector field singularities were defined as zeros of the vector field, which does not have a meaning for direction fields. Instead, we will say that direction field singularities are points where the direction is undefined. Therefore, we identify singularities of a direction field with “holes” in its domain of definition, and singularity index keeps the same definition (7.11) as it depends only on the vector direction.

This distinction is very important : instead of being a characterization of the direction field, it is a characterization of its support, of its domain of definition. We will see later the consequences of this remark.

Symmetry issues

For N -symmetry vector fields, the integral in (7.11) is not necessarily a multiple of 2π , but only of $2\pi/N$. As a result, the singularities of a N -symmetry direction field may have fractional indexes. The discretization framework that we will present in the next section will allow to handle such fractional singularities in a very simple manner. These fractional singularities are not abstract objects : if you look at the tip of your finger, you will see a $+1/2$ and a $-1/2$ singularity formed by the lines of your fingerprint, that can be seen as a 2-symmetry direction field.

7.3.3 N -symmetry direction field Singularities on surfaces

The 2D definition for direction field singularity index cannot be directly extended to surfaces embedded in \mathbb{R}^3 because it lacks a reference vector to define θ . However, it can be shown that for $\mathbf{x} \in \mathbb{R}^2$:

$$T_{\vec{d}}(\partial\Omega(\mathbf{x})) = I_{\vec{d}}(\mathbf{x}) - 1 \quad (7.12)$$

As the turning number can be extended to surfaces in \mathbb{R}^3 because the reference vector is the tangent to the cycle, this allows us to extend the definition of the index of a singularity on a surface in \mathbb{R}^3 to :

$$I_{\vec{d}}(\mathbf{x}) = 1 + T_{\vec{d}}(\partial\Omega(\mathbf{x})) \quad (7.13)$$

This allows to see a singularity as a special case of turning number along a cycle enclosing a hole of the domain of definition of the direction field. This justifies the introduction of turning numbers as more general tools to study direction field topology. Finally, this definition for singularity indexes will allow us to demonstrate the Poincaré-Hopf theorem as a simple consequence of Théorème 7.2.1.

7.3.4 Poincaré Hopf theorem

With this definition for the index, we can generalize the Poincaré-Hopf theorem to N -symmetry direction fields :

Théorème 7.3.1 (Poincaré-Hopf) *The sum of singularity indexes on a closed surface \mathcal{S} equals its Euler characteristic :*

$$\sum_{i=1}^b I_i = \chi(\mathcal{S}) = 2 - 2g$$

Proof : Notice first that this theorem is on surfaces without boundary, so the number of borders is absent in the Euler characteristic. Let us call \mathbf{p}_i the point of index I_i and $\mathcal{S}_h = \mathcal{S} \setminus \{\Omega(\mathbf{p}_i)\}$ (surface with borders in place of singularities), on which the field is continuous because it does not contain the singularities. Applying the boundary relation (7.6) to \mathcal{S}_h yields :

$$T(\partial\mathcal{S}_h) = T(\partial\mathcal{S}) - \sum_{i=1}^b T(\partial\Omega(\mathbf{p}_i)) = - \sum_{i=1}^b (I_i - 1) = -\chi(\mathcal{S}_h) = -\chi(\mathcal{S}) + b$$

we have $T(\partial\mathcal{S}) = 0$ because \mathcal{S} has no borders, and we get the Poincaré-Hopf theorem by subtracting b in these equalities \square

The concept of replacing singularities with holes allows singularities to be handled as borders. Moreover borders are topological objects which are very well understood through cycle homology. In what follows,

we will use again this idea of replacing singularities with holes, and to characterize the singularity by the behavior of the field along the boundary of the hole. The other advantage of this concept is that borders have indexes, which correspond to the index of the singularity obtained if we contract the border to a single point.

Note that Théorème 7.2.1 generalizes the Poincaré-Hopf theorem in 2 ways :

1. It generalizes it to N -symmetry directions, that is to singularities with fractional indexes.
2. It generalizes it to surfaces with a boundary, whereas Poincaré-Hopf is only valid on closed surfaces, or with very strict boundary condition. It is that central concept of boundary that enabled such generalization.

Remarque 7.3.2 (Topological degrees of freedom) *Singularities do not hold all the topological degrees of freedom, since a homology basis also contains generators of the surface, which do not enclose any singularity (they are non-disconnecting). This justifies why we need the concept of turning number to capture the direction field topology, and not only the singularity indexes.*

We now need a discretization for direction fields that can access turning numbers, in order to guarantee control over direction field topology.

7.4 Discretization of N -symmetry direction fields

We will now present a discretization for N -symmetry direction fields based on the continuous theory, and in particular allowing to have access to the continuous quantities and tools we have defined in the previous section. This will be done in two steps :

1. Define a direction on each facet
2. Define how the direction behaves between two facets

This discretization will lead us to define a *period jump* between facets removing the ambiguity that exists when defining directions pointwise. We will then explain how period jumps are related to turning numbers, thus to direction field topology. In particular, we will detail the coherence between this discretization and the continuous theory presented in Section 7.1 and Section 7.2. Finally, we will explain how this framework allows to handle arbitrary singularities.

In this section, we assume that our mesh is oriented i.e., facets have coherent normals \mathbf{n} , and edges have an orientation. This allows to define a unique orientation for dual edges (between facets). We use notations from Discrete Exterior Calculus [Desbrun et al. 2005a] on the dual mesh as it makes the exposition clearer and the proofs easier. In this setting, 0-, 1- and 2-forms are scalars on facets, oriented dual edges, and vertices respectively. The choice of the dual graph comes from the fact that we need a well defined tangent plane to define a direction. Whereas it is well defined at dual vertices $\star f$ (plane of facet f), it is not at primal vertices. Most of our considerations will not depend on the embedding $[\star M]$ of the dual mesh. However, to visualize a direction field, we will need to interpolate it (see Chapitre 8), in which case we chose the barycentric dual which is always well defined (in particular the embedding of a dual vertex $[\star f] = \mathbf{g}(f)$ always belongs to $[f]$).

The structure presented in this section requires particular attention as it will serve as a basis for the following chapters. Chapitre 8 will define an interpolation scheme for direction fields based on this structure. This interpolation will be used to devise a visualization scheme allowing for the visualization of our direction fields, and in particular their arbitrary singularities. Chapitre 9 will present two algorithms

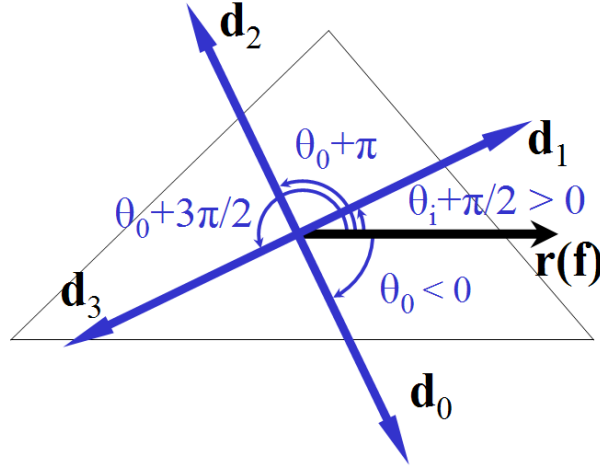


FIG. 7.8 – A N -symmetry direction (here $N = 4$) on a triangle t_i is a set of vectors \mathbf{d}_k , $k \in \{0..N-1\}$ defined as the images of a reference vector \mathbf{r}_i by rotations of $\theta_i + k(2\pi/N)$. By convention, negative angles correspond to clockwise orientation (here $\theta_i < 0$ but $\theta_i + \pi/2 > 0$).

whose combination allow for a topologically controlled direction field design tool. Chapitre 10 will not use explicitly this structure but the continuous concepts of Section 7.1.

7.4.1 N -symmetry directions on facets

A N -symmetry direction \vec{d} on a surface is defined as a set of N unit vectors lying in the tangent plane of the surface that is preserved by rotations of $2\pi/N$ around the normal. In other terms, its N representatives $\{\mathbf{d}_i\}_{i=1..N}$ all verify $\mathbf{d}_i \cdot \mathbf{n} = 0$ and $\|\mathbf{d}_i\|^2 = \mathbf{d}_i \cdot \mathbf{d}_i = 1$.

The first step of the discretization of a direction field \vec{d} is then to define one direction on each vertex of the dual mesh, that on each facet f (see Figure 7.8). This is done by choosing a reference direction $\mathbf{r}(f)$ in each facet f (for instance one of the oriented edges of f), and defining \vec{d} by an angle $\theta_0(f) \in \mathbb{R}$ such that :

$$\forall \mathbf{d}_i(f) \quad \exists k_i \text{ such that } R(\mathbf{r}(f), \theta_0(f) + k_i 2\pi/N) = \mathbf{d}_i(f)$$

where $R(\cdot, \theta)$ is the rotation of angle θ in the tangent plane $T\mathcal{S}$. As θ_0 is defined by a real on each facet, it is a dual 0-form. As this section will make use of several 0,1 and 2-forms, all the forms that we use will be indexed by the order of the form (for instance the 0 of θ_0 means that it is a 0-forms).

This representation can be used in itself. However, we have too little information to have access to the topology of the direction field. In particular we do not know how it behaves between the discretization points. To solve this problem, we introduce the notion of period jumps, that explicits the behavior of the direction field between discretization points (facet centers). Based on period jumps, we will be able to recover and control all the topology of the field based on the continuous theory, and in particular to have access to the turning numbers of the fields.

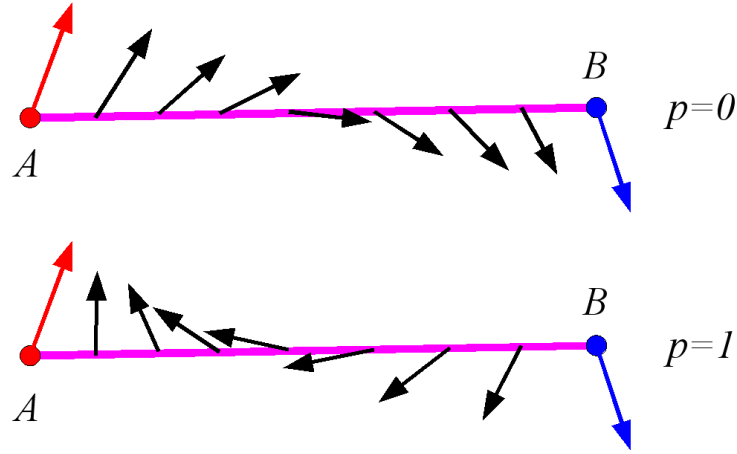


FIG. 7.9 – Direction interpolation ambiguity : between two directions given at points A and B, we have different possible interpolations, which can be chosen using the variable p .

7.4.2 Curvature and period jumps

Let us start with a simple example to explain direction interpolation ambiguity. Assume that we want to interpolate a direction \vec{d} along an edge $[AB] \subset \mathbb{R}^2$ knowing $\vec{d}(A)$ and $\vec{d}(B)$. If \vec{d} is continuous, the angular variation along $[AB]$ verifies :

$$\kappa(A \rightarrow B) = \int_A^B d\theta = \angle(\vec{d}(B), \vec{d}(A)) + 2p\pi \quad (7.14)$$

where we call $p \in \mathbb{Z}$ the period jump, and $\angle(.,.)$ is the angle in $(-\pi, \pi]$. The angular variation has infinitely many possible values for all the possible values of p (see Figure 7.9). Hence to interpolate the direction field on $[AB]$, we need both the values at A and B and an integer p (period jump), such that the linearly interpolated value at a point $P = (1-t)A + tB$ is written :

$$\theta(P) = \theta(A) + \kappa(A \rightarrow B)t \quad (7.15)$$

We can take $\theta(A) \in (-\pi, \pi]$, but according to the choice for p , $\theta(B)$ can be outside this interval. We will now extend this concept to a mesh. The main difficulty is then that there is no "universal" reference vector from which to compute the angle θ , but only one reference vector local to each facet. We are now going to solve this by bringing adjacent facets in the same plane, and extending (7.14) in this common plane.

Let f_1 and f_2 be two adjacent triangles such that the dual edge $f_1 f_2$ is oriented from f_1 to f_2 . We can isometrically bring f_1 and f_2 to be coplanar, and define the *curvature* 1-form κ_1 of the direction field \vec{d} along $f_1 f_2$ as the angle of rotation that brings \vec{d}_1 to \vec{d}_2 . Simple geometry (see Figure 7.10) shows that θ_0 only defines the curvature 1-form up to integer multiples of $2\pi/N$:

$$\kappa_1(\theta_0, p_1) = \kappa_1^r + d_0\theta_0 + 2\pi p_1/N \quad (7.16)$$

where :

- $\kappa_1^r(ij)$ is the angle of a rotation that brings \mathbf{r}_i to \mathbf{r}_j (see Figure 7.10), given by : $\kappa_1^r(ij) = \angle(\mathbf{r}_i, ij) + \angle(ij, \mathbf{r}_j)$ where \angle is the angle oriented by the triangle normal. Defining $\kappa_1^r(ji) = -\kappa_1^r(ij)$ makes κ_1^r a 1-form. Note that κ_1^r depends only on the mesh and on the (arbitrary) choice of a reference vector on each facet. It is independent of the direction field that we are defining ;

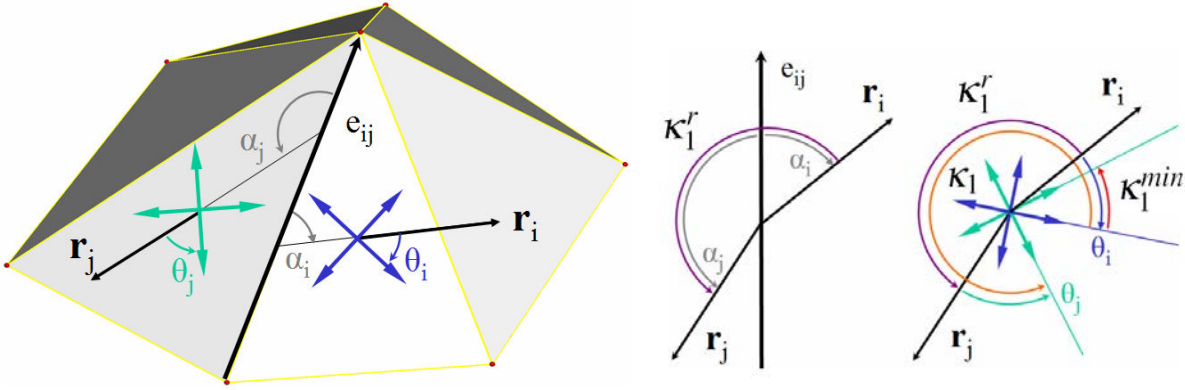


FIG. 7.10 – **Left** : 4-symmetry directions on adjacent facets with common edge e_{ij} . **Middle** : The rotation angle between reference vectors \mathbf{r}_i and \mathbf{r}_j is $\kappa_1^r(ij) = \alpha_j - \alpha_i$. **Right** : $\kappa_1 = \kappa_1^r(ij) + \theta_0(j) - \theta_0(i)$ is a rotation angle between the directions on t_i and t_j , and so is any angle $\kappa_1 + p_1(ij)(2\pi/N)$. We represent here κ_1 and the minimum angle κ_1^{\min} corresponding to $p_1(ij) = 0$ and $p_1^{\min}(ij) = -3$.

- d_0 is the exterior derivative for 0-forms given by $(d_0\theta_0)(ij) = \theta_0(j) - \theta_0(i)$;
- p_1 is an integer 1-form. As both smoothness and singularity indexes will depend on the curvature, we need to introduce this p_1 to remove the ambiguity (as in Figure 7.9 but along a dual edge). Considering these integers associated to dual edges as an integer 1-form makes it simpler to prove the validity of our method (see 10.3.1).

As explained in [Ray et al. 2008b], because a direction field has unit norm, its covariant derivative is in the orthogonal direction, so it is uniquely defined by its projection in this direction. We call this projection curvature, as this is the way the curvature of a curve is defined (projection of the derivative of a unit tangent vector in the orthogonal direction). Integrating such a curvature gives a difference of angles, which explains why the definition for discrete curvature is based on differences of angles.

The couple (θ_0, p_1) along with the choice of reference vectors defines a unique direction field with a well defined curvature :

1. The 0-form θ_0 represents the angle between \vec{d} and \mathbf{r} on each facet, such that $\vec{d} = R(\vec{d}_0, \theta_0 + 2k\pi/N)$ on dual vertices.
2. The integer 1-form p_1 called period jump represents the behavior of \vec{d} between discretization points, thus defines its curvature and its topology.

This will allow us to compute its turning number along any cycle, thus its singularities.

7.4.3 Turning numbers

In the discrete setting, our continuous notion of curve matches partially with 1-chains : a curve embedded in a triangulated mesh can be approximated by a set of edges or dual edges, which is a 1-chain with integer values (the cycle can pass through an edge more than once, and in any direction). The 1-chains we use in this section are of this form. However, a more modern approach consists in allowing the 1-chain to have real values, and seeing it as a "rasterization" of a continuous cycle on primal or dual edges. Using the definition of the turning number (7.5) and the discrete curvature 1-form (7.16), we derive an expression for the turning number of a (discrete) direction field \vec{d} along a cycle (closed 1-chain) γ in the

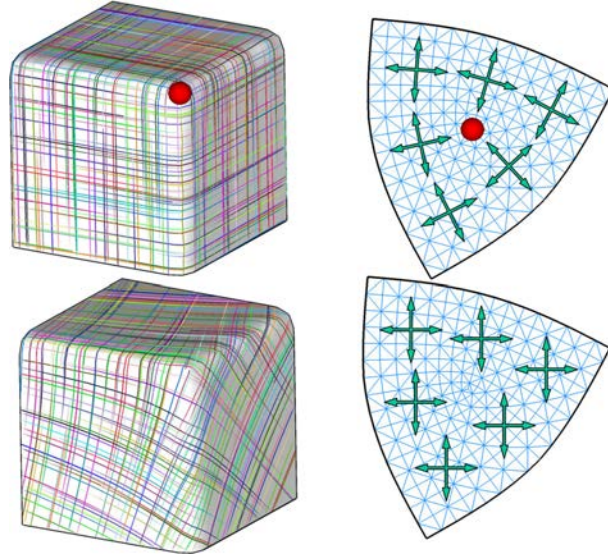


FIG. 7.11 – 4-symmetry direction field on a cube corner. **Top row** : a singularity of index $1/4$ placed at the corner exactly balances the angle defect, so the field is not distorted on the cube (left) but distorted in the map (right). **Bottom row** : removing the singularity allows to define a direction field that is constant in the map (right) but necessarily distorted on the surface (left).

discrete setting :

$$T_{\vec{d}}(\gamma) = \frac{1}{2\pi} \int_{\gamma} \kappa_{\vec{d}} - \kappa_{\gamma} = \frac{1}{2\pi} \langle \kappa_1^r - \kappa_{\gamma} | \gamma \rangle + \frac{1}{N} \langle p_1 | \gamma \rangle \quad (7.17)$$

where $\langle \omega | \gamma \rangle$ is the discrete equivalent of $\int_{\gamma} \omega$. Numerically, this is simply obtained by a dot product between the vectors corresponding to the 1-forms ω and 1-chain γ . We used the fact that $\langle d\theta | \gamma \rangle = \langle \vec{\theta} | \partial\gamma \rangle = 0$ because γ is a cycle. As the first term is independent from \vec{d} , this means that the topology of \vec{d} is defined entirely by the period jumps p_1 and does not depend on its geometry θ_0 . Finally, $\langle \kappa_{\gamma} | \gamma \rangle$ can be computed as the sum of oriented angles between input and output edges along all triangles crossed by γ .

7.4.4 Singularity index

In Section 7.3, we have defined singularities by the turning number of a small cycle around the singularity. On the dual graph, the smallest such cycle is the boundary of a dual cell $\star v$, such that we cannot distinguish singularities within such a cell. Hence we will naturally assume that the singularity lies at the center v of the cell.

To compute the singularity index at a primal vertex v , we can simply choose $\gamma = \partial \star v$ ($\star v$ is a dual facet), in which case $\langle \kappa_{\gamma} | \gamma \rangle$ is simply the sum of angles incident to v . Using (7.13), the index of the singularities of a N -symmetry direction field is a 2-form :

$$I_2(\theta_0, p_1) = \frac{d_1 \kappa_1(\theta_0, p_1) + K_2}{2\pi} = \frac{d_1 \kappa_1^r + K_2}{2\pi} + \frac{d_1 p_1}{N} = I_2^r + \frac{d_1 p_1}{N} \quad (7.18)$$

where :

- d_1 is the exterior derivative for 1-forms given by $(d_1 f_1)(v) = \sum_{\star e \in \partial \star v} f_1(\star e)$ where $\partial \star v$ denotes the oriented boundary of the dual cell relative to vertex v

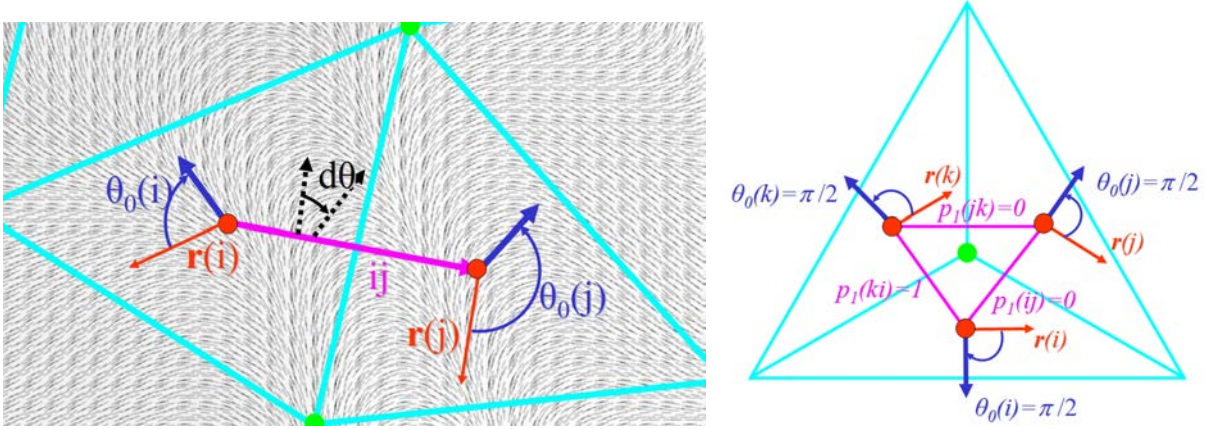


FIG. 7.12 – **Left** : the period jump $p_1(ij)$ must be set according to Equation (7.19) to ensure that the field rotation along the dual edge ij is the same as in the original vector field (visualized by the gray integral lines). **Right** : Converting a source singularity. One can check that the index computed using Equation (7.18) equals to 1.

- K_2 is the angle defect 2-form. $K_2(v)$ is 2π minus the sum of angles of triangle corners adjacent to v , which corresponds to the integrated Gaussian curvature over the dual cell $\star v$. The average Gaussian curvature over $\star v$ is then given by $K_2^*(v) = K_2(v)/\|\star v\|$ where $\|\star v\|$ is area of $\star v$. It can be approximated by (a third of) the 1-ring area of v .
- $I_2' = d_1 \kappa_1' + K_2/2\pi$ is the index of the reference field (field of reference vectors). It is necessarily an integer that depends only on the choice of the reference vectors.

Indexes are multiples of $1/N$ and a zero index corresponds to no singularity. On the figures, singularities with positive (resp. negative) indexes are marked by small red (resp. blue) spheres. Note that the indexes of singularities are controlled by p_1 alone, whatever choice for θ_0 .

We will end this section on the discretization of direction fields by explaining how to convert an existing vector field in any kind of representation into our structure. This is possible as long as there exists an interpolation such that the vector field is defined everywhere on the surface.

7.4.5 Converting vector fields into our representation

To manipulate existing vector fields under any representation, it is necessary to convert them into our structure. This is possible as long as the vector field $\mathbf{v}(\mathbf{x})$ to convert is known at each point \mathbf{x} of the mesh M . This is the case in all FEM modeling where the vector field is described as a sum of basis functions defined on the whole mesh, and in the case the vector field is interpolated (linearly or not) between samples. We can also use this framework to convert 1-forms from the DEC setting into our representation by using their interpolation with Whitney forms (see Chapitre 3). If the vector field we wish to convert is only sampled, we suggest to simply interpolate it linearly between the samples.

The conversion can be done in 4 steps :

1. Singularity placement : To preserve the exact location of singularities, we need to add a vertex at its exact position, which leads to splitting in 3 the corresponding facet.
2. Choose a reference vector $\mathbf{r}(f)$ on each facet

3. Compute the angle $\theta_0(f) = \angle(\mathbf{v}(\mathbf{g}(f)), \mathbf{r}(f))$ and norm $r(f) = \|\mathbf{v}(\mathbf{g}(f))\|$ at each facet barycenter $\mathbf{g}(f)$.
4. Compute the period jumps such that the field rotation along dual edges is preserved. This rotation is given by the integral of the angular variation of the vector field along the dual edge :

$$\kappa_1(ij) = \int_{*ij} d\theta = \int_{*ij} \frac{v_x dv_y - v_y dv_x}{v_x^2 + v_y^2}$$

This value can be evaluated using the analytic representation of the field. Once the $\kappa_1(ij)$ is evaluated from the original field, one can compute the period jump of the dual edge using Equation (8.2) :

$$p_1(ij) = \frac{1}{2\pi} (\kappa_1(ij) - \angle(\mathbf{r}(jil), \mathbf{r}(ijk)) + \theta(jil) - \theta(ijk)) \quad (7.19)$$

In Figure 7.12, we show an example of how to compute $\theta_0(f)$ and period jump $p_1(ij)$. We have also illustrated a concrete example where we convert an existing field with a singularity of the type source into our representation.

7.4.6 Conclusion

This chapter proposed an in depth study of direction fields and of their topology in the continuous case. It also proposed a discretization coherent with this study, and allowing to control the direction but also its curvature. This control of the curvature gives us explicit control over the topology. The algorithm presented and commented in Chapitre 9 will take advantage of that to propose a topologically controlled direction field design method. This chapter also provided the theory to understand the important relations between the curvatures of the direction field and of the surface it is defined on. We will take advantage of them in Chapitre 10 to propose a direction field processing method with a geometry aware direction field processing. In both cases, we will need to visualize our results. The next chapter focuses on this point, and on the interpolation of direction field discretized with our period jumps based structure.

8

Visualisation

Sommaire

8.1 Previous work	132
8.1.1 Representing Higher-order Singularities in Vector Fields	132
8.1.2 Visualization of Vector Fields on Surfaces	136
8.2 Interpolation	136
8.2.1 Step 1 : 0D	136
8.2.2 Step 2 : 1D	137
8.2.3 Step 3 : 2D	138
8.3 Corrected interpolation	139
8.4 The GPU-based Visualization Algorithm	141
8.4.1 Pass 1 : Depth Value for Geometric Discontinuities	141
8.4.2 Pass 2 : Object to Image Space Projection	141
8.4.3 Pass 3 : LIC in the Image Space	142
8.5 Results, Discussions and Conclusions	143

Introduction

Dans les deux prochains chapitres, nous allons travailler sur des outils de manipulation de champs de directions et il nous sera donc nécessaire de disposer d'un outil de visualisation efficace pour ces champs. Comme nous allons utiliser la structure de données présentée au Chapitre 7, nous voulons que cet outil de visualisation tire profit de cette structure. En particulier, on a vu que les "sauts de périodes" déterminent le comportement du champs entre les points de discrétisation, et on va donc pouvoir construire une interpolation (nécessaire à la visualisation) qui ne fera qu'explicitement ce comportement. De plus, cette interpolation basée sur les "sauts de périodes" va nous permettre de représenter des singularités d'ordre arbitraire, s'affranchissant ainsi des contraintes habituelles sur la complexité des singularités représentées. Ainsi, cette technique présente un intérêt majeur par rapport à l'état de l'art, au delà d'être un simple outil pour visualiser "nos" champs de directions.

La différence majeure avec les approches classiques de visualisation de champs de vecteur est que notre interpolation sera, dans un sens, en coordonnées polaires, puisque l'on interpolera la direction et la norme séparément (la norme ne sera s'ailleurs pas utilisée pour la visualisation).

Plan du chapitre

Dans ce chapitre, nous commencerons par discuter l'état de l'art dans le domaine de la visualisation de champs de vecteurs sur les surfaces (Section 8.1). Nous n'avons pas inclus cet état de l'art dans celui du Chapitre 6 puisqu'il s'intéresse à un problème bien séparé (la visualisation au lieu de la manipulation et du traitement). Nous présenterons ensuite un premier schéma d'interpolation de champs de directions (Section 8.2) puis une correction proposée pour assurer plus de régularité (Section 8.3). Nous utiliserons ensuite ces interpolations pour construire à la Section 8.4 une méthode de visualisation basée sur la méthode LIC (Line Integral Convolution) et accélérée par la carte graphique. Cette méthode de visualisation de champs de directions basés sur notre discrétisation opère dans l'espace image. Nous finirons par présenter des résultats de cette méthode à la Section 8.5 appliquée à différents champs de directions sur des surfaces triangulées.

Contributions

- Nous proposons deux schémas d'interpolation basés sur la discrétisation des champs de directions proposée au Chapitre 7.
- La non linéarité de ces interpolations, couplée avec la notion de saut de période nous permettent de représenter des singularités d'ordre arbitraire, s'affranchissant ainsi des limites de l'état de l'art. Ces singularités ne peuvent apparaître que sur les sommets, et leurs indices sont déterminés de façon combinatoire.
- Nous démontrons l'efficacité de cette représentation en proposant une visualisation basée sur la méthode LIC, accélérée par la carte graphique et qui opère dans l'espace image.

Comme rappelé dans l'introduction de cette partie, le contenu de ce chapitre correspond à l'article [Li et al. 2006b] et a été laissé en anglais.

8.1 Previous work

Vector fields on surfaces are important objects which appear frequently in scientific simulation in Computational Fluid Dynamics (CFD) or in Finite Elements Modeling (FEM) [Schroeder et al. 2005]. To be visualized, such vector fields are usually linearly interpolated for the sake of simplicity and performance considerations. Namely, the vector field is sampled at each vertex of the underlying piecewise linear surface and interpolated linearly, similarly to what is done for the geometry. However, the price that one pays for the simplicity that comes with the linearity is that only linear (or first-order) singularities can be represented in the triangles [Tricoche 2002]. This leads to the mis-representation of some vector fields where higher-order singularities are present (e.g. a dipole of a magnetic field). This mis-representation of the singularities changes the perception of the topology of the vector fields and hence may lead to a bad interpretation of the simulation results. Therefore, finding a simple representation that enables the meaningful information of the higher-order singularities to be preserved is a research direction worth exploring.

8.1.1 Representing Higher-order Singularities in Vector Fields

Different methods have been proposed to encode vector fields on surfaces, using either linear schemes, or more sophisticated higher-order schemes. In all these methods, the vector fields are supposed to be

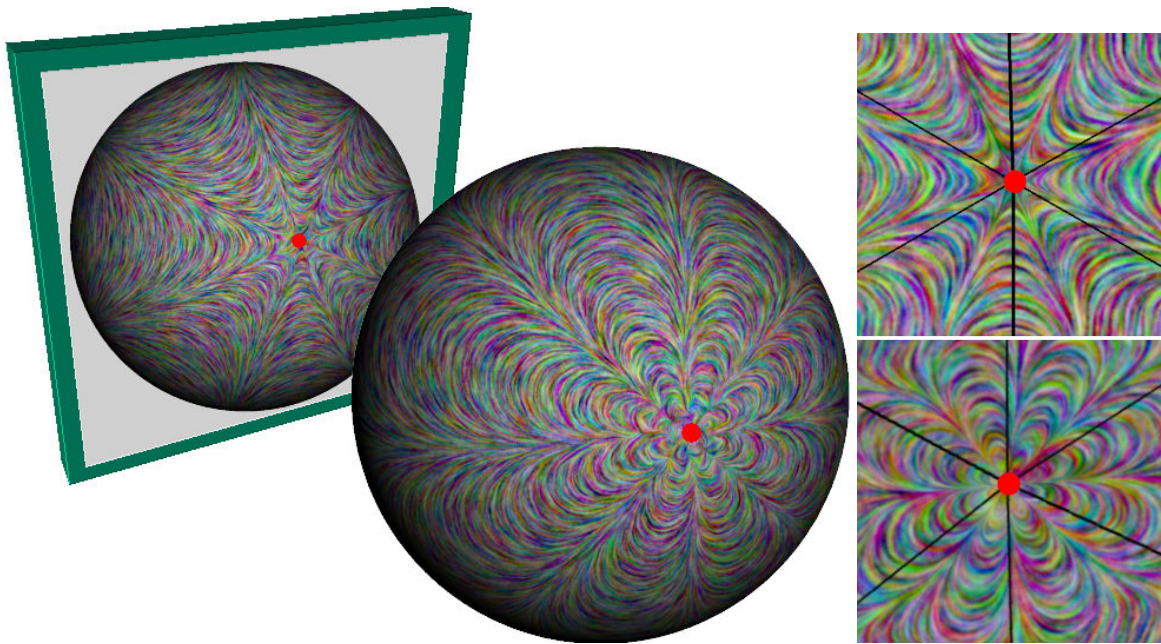


FIG. 8.1 – **Left** : a sphere with only two higher-order singularities of index +5 (front) and -3 (back), the sum of which equals the Euler characteristic (2 for a sphere). **Right** : zooming in the vicinity of the singularities. One can observe that the hyperbolic and elliptic structure of the curvilinear sectors, which have been described in Tricoche *et al.*'s work (see Figure 8.3). Note that using our approach, the separatrices of the curvilinear sectors do not need to appear on the incident edges (black lines) of the vertex.

sampled on the vertices of the piecewise linear domains.

Linear schemes

Among the pioneers in this direction, Tricoche *et al.* [Tricoche *et al.* 2000a] explored the possibility of representing higher-order singularities using only piecewise linear 2D vector field. Their method is based on the observation that the vicinity of a critical point in a 2D vector field can be characterized as a sequence of these three basic sector types [Andronov *et al.* 1973, Firby and Gardiner 1982]. In fact, they can be understood as three possible interpolations of two radial vectors originated from the vertex in the counter-clockwise direction. In the parabolic and hyperbolic sectors, the angle of rotation is defined as $\angle(\mathbf{v}(q_2), \mathbf{v}(q_1))$, where $\angle(\cdot, \cdot)$ is the natural rotation angle which is always in $(-\pi, \pi]$. This angle is always positive (resp. negative) in parabolic (resp. hyperbolic) sectors. In the elliptic sector, the angle of rotation defined as $2\pi + \angle(\mathbf{v}(q_2), \mathbf{v}(q_1))$. Note that since the elliptic sector is non-linear, it has to be split into two triangles when being modeled by linear triangles.

Based on this observation, Tricoche *et al.* proposed a way to model singularities with complex topologies on a vertex and conversely, given a vertex, to classify the possible types of singularities encountered on that vertex. Their idea is to “synthesize” non-linear singularities on a vertex by using a sequence of parabolic, hyperbolic and elliptic sectors (see Figure 8.3). The technique has been applied to simplify the topology of 2D vector fields [Tricoche *et al.* 2000b]. The main limitation of the purely piecewise linear schemes is that the complexity of singularities that a vertex can represent is limited by its connectivity. Moreover, the elliptic sectors need to be split in two due to their non-linearity.

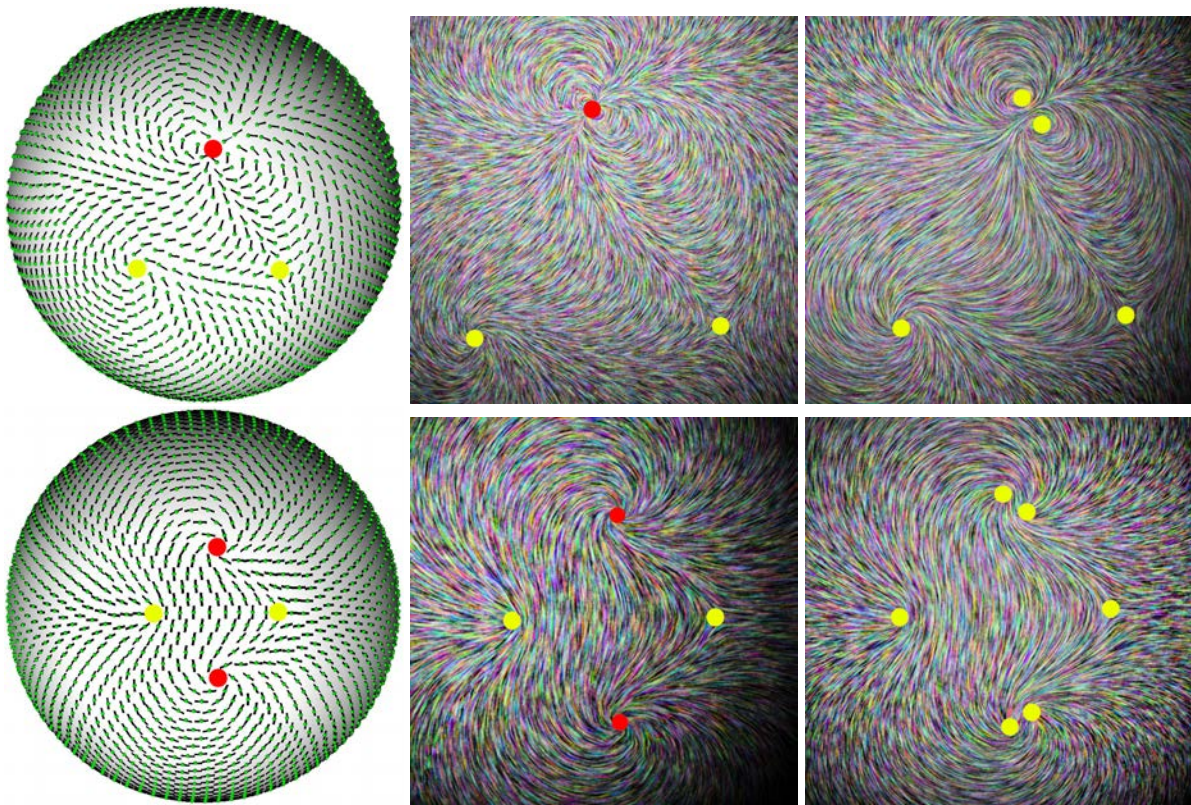


FIG. 8.2 – Two examples of non-linear vector fields on the sphere represented by our facet-based data structure (left). The top one is a vector field. The bottom one is a vector field with sign ambiguity (for instance, an eigenvector field of a second-order tensor field). First-order (yellow) and higher-order (red) singularities are defined on vertices. The vector fields are visualized using our LIC-based method (middle). On the right, we show the same two vector fields represented by the classic vertex-based piecewise linear representation. Note that the higher-order singularities are split into combinations of first-order singularities.

In [Tricoche et al. 2000b], the above scheme was used to simplify 2D vector fields. First, the singularities to be merged are clustered. Then, each cluster is represented by a piecewise linear vector field with only one higher-order singularity. As with the method in [Tricoche et al. 2000a], higher-order singularities are “synthesized” on vertices with the three basic sector types. Thus, as far as higher-order singularities are concerned, this method also suffers from the fact that the maximum complexity of a singularity at a vertex is bounded by the number of triangles meeting at that vertex.

In [Theisel 2002], Theisel introduced a method to design piecewise linear vector fields with prescribed topology. With his approach, all types of critical points of a 2D vector field can be represented. Theisel et al. [Weinkauff et al. 2004] extended the idea of [Theisel 2002] to 3D vector fields. Similarly to the 2D case, the vicinity of a critical point is segmented into sectors with a regular behavior of the flow. They achieved the construction of higher-order 3D vector fields by designing their topological skeletons.

In [Weinkauff et al. 2005], an approach is proposed to extract and classify higher-order singularities of 3D vector fields. This is achieved by extracting the topological skeleton of a 2D vector field on a convex surface around the area of interest. They demonstrated vector field simplification as an application for this method.

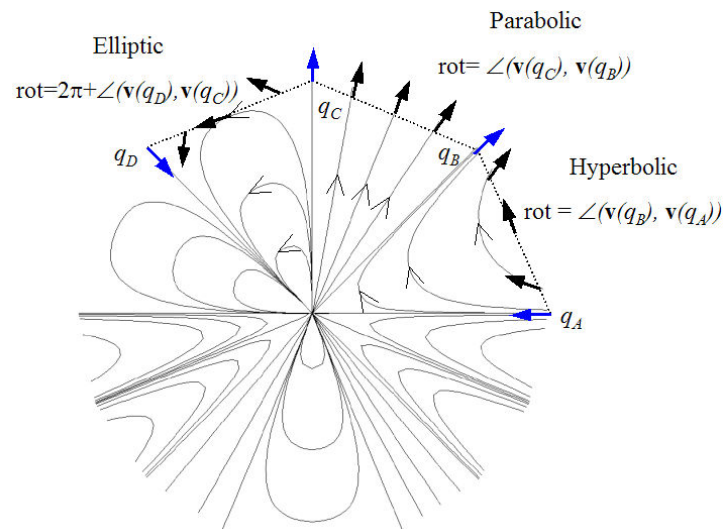


FIG. 8.3 – Modeling a higher order singularity with the three types of curvilinear sectors by Tricoche *et al.*'s method

Mixed schemes and non-linear schemes

In [Scheuermann *et al.* 1998], a mixed higher-order/linear vector field scheme is proposed to visualize non-linear vector field topology. Some non-linear schemes, for instance, [Scheuermann *et al.* 1999, Zhang *et al.* 2004] were also proposed to represent higher-order singularities.

In [Scheuermann *et al.* 1998] Scheuermann *et al.* proposed a method to first detect higher-order singularities by computing the indexes of the triangles in the non-linear singular regions and then approximating the singularities by using polynomials of high degrees. In the non-singular regions, they use piecewise linear interpolation to represent the field.

To improve the *approximation* of higher-order singularities in a given non-linear vector field, Scheuermann *et al.* [Scheuermann *et al.* 1999] proposed a C^1 interpolation scheme (based on Powell-Sabin and Nielson's interpolants) for 2D vector field defined on a piecewise linear planar domain. Although higher-order singularities are still split into combinations of first-order ones, the overall topology of the vector field is better reflected by their C^1 interpolation scheme as compared to previous work.

In [Zhang *et al.* 2004], Zhang *et al.* presented a non-linear scheme to represent non-linear singularities through the use of exponential maps (around the one-ring of a vertex) and parallel transport. The non-linear property makes the interpolation scheme quite complicated and computationally expensive. More specifically, the construction of the exponential map, which is basically a local parameterization of the one-ring of a vertex, impedes the acceleration by graphics hardware. They extended the definitions to tensor fields in [Zhang *et al.* 2005].

Unfortunately, mixed and non-linear schemes impede the acceleration by graphics hardware. It is therefore difficult to achieve an interactive framerate with those approaches.

8.1.2 Visualization of Vector Fields on Surfaces

Since the introduction of LIC (Line Integral Convolution) by Cabral and Leedom in 1993, the method has been widely used for the visualization of 2D vector fields. In this thesis, we have also chosen to use the concept of LIC to visualize our vector fields. After the invention of this method, much work has been done to accelerate the original algorithm. For instance, in [Stalling and Hege 1995] Stalling and Hege presented the FastLIC method, which employs simple box filters. This method minimizes the total number of streamlines and hence accelerates the original LIC by an order of magnitude.

Later, work has been done to extend the 2D LIC technique to visualize vector fields on surfaces. As said in [Laramee et al. 2004], these methods can be classified into mainly three types according to the space in which LIC operates :

- *Parametric space* : Forssell and Cohen [Forssell 1994, Forssell and Cohen 1995] proposed a method that allows the operation of LIC through a parameterization [Floater and Hormann 2004] of the surface. However, a global parameterization of a general surface may not always be obtained easily. Battke et al. [Battke et al. 1997] and Carr et al. [Carr and Hart 2002] proposed methods that allow the operation of LIC in parametric space per triangle packing algorithms. Triangle packing can be considered as the simplest local parameterization of the surface by triangle. Nevertheless, the drawback of the triangle packing is that it is quite sensitive to the quality of the mesh.
- *Object space* : Immersing the vector field on the surface into a 3D volume enables the field to be treated as 3D, which means that streamline tracing can be performed [Rezk-Salama et al. 1999, Interrante 1997].
- *Image space* : Two methods, ISA (Image Space Advection) [Laramee et al. 2003] and IBFVS (Image Based Flow Visualization for Curved Surfaces), [van Wijk 2003] have been proposed to enable high-performance visualization of flow on surfaces. A side-by-side comparison of the two methods can be found in [Laramee et al. 2004]. Working in this space, no parameterization of the surface is needed. Moreover, LIC can be accelerated by taking advantage of graphics hardware [Heidrich et al. 1999]. Seeing the advantages of the image space based methods, we have chosen to carry out LIC in image space as well.

Let us now enter the heart of the matter, starting by an interpolation scheme for direction fields in our structure.

8.2 Interpolation

Using the idea of integer period jumps to remove the direction interpolation ambiguity, we will start by explaining how to interpolate a vector field on the mesh M in three steps of increasing dimension : facet centers (0D), dual edges (1D), mesh (2D).

8.2.1 Step 1 : 0D

We remind the reader that we call $[ij]$ and $[ijk]$ the edge and triangle based on vertices i , j and k , and $\mathbf{g}(s)$ the gravity center of a simplex, s . In our structure, we have a direction sample $\vec{d}(f)$ on each facet f . The first step of the interpolation is to place this direction $\vec{d}(f)$ at the center of mass $\mathbf{g}(f)$ of f . In our framework, remember that $\vec{d}(f)$ is defined by its angle $\theta(f) \in \mathbb{R}$, which is the angle between $\mathbf{d}(f)$ and the facet reference vector, $\mathbf{r}(f)$. In the case we want to interpolate a vector field, we can also introduce a scalar $r(f)$ such that $\mathbf{v}(f) = r(f)\mathbf{d}(f)$ is a N -symmetry vector. As θ , r is also a dual 0-form simply a

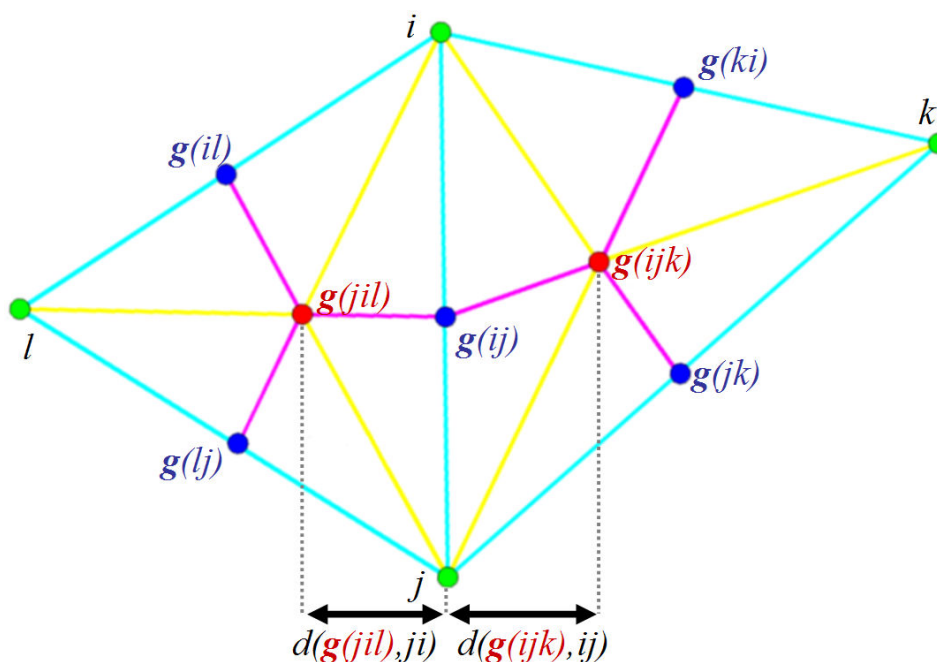


FIG. 8.4 – Illustration on two adjacent (primal) triangles of a primal mesh (light blue) and its barycentric dual (magenta). Subdivision simplices are triangles based on a primal vertex (green), a dual vertex (red), and an edge middle (dark blue). Their edges are based on primal and dual edges, and edges between primal and dual vertices (yellow).

scalar. Even if our visualization scheme does not take a norm into account, we introduce this scale r in case the interpolation scheme we describe is to be used with other purposes.

Notice that $\vec{d}(f)$, $r(f)$ and $\theta(f)$ are given at $\mathbf{g}(f)$. They will be interpolated, so they are not constant on the whole facet, and we only have $\vec{d}(f) = \vec{d}(\mathbf{g}(f))$. but we omit the G for the sake of brevity and because it is not ambiguous. Moreover, it corresponds to the implementation where a θ and a r are stored for each facet.

8.2.2 Step 2 : 1D

The main idea of our period jumps based structure is to consider a vector by its direction and norm, and not by its coordinates (as in the standard representation) or by its integral along an edge (as in DEC representation). We will preserve this idea when choosing an interpolation for direction fields under such representation. Namely, we will interpolate the polar coordinates of the vector. Thus this second interpolation step consists in interpolating separately the norm and direction of \mathbf{d} along the barycentric dual edges.

For an edge ij between a pair of adjacent triangles ijk and jil , we can define geometrically the barycentric dual edge (see Figure 8.4) :

$$[*ij] = [\mathbf{g}(ijk)\mathbf{g}(ij)] \cup [\mathbf{g}(ji)\mathbf{g}(jil)] \quad (8.1)$$

Given a period jump $p_1(ij)$, the curvature along $\star ij$ is given by :

$$\begin{aligned} \kappa_1(ij) &= \int_{\star ij} d\theta = \\ &\angle(\mathbf{r}(ijk), \mathbf{r}(jil)) + \theta(jil) - \theta(ijk) + 2p_1(ij)\pi \end{aligned} \quad (8.2)$$

where the angle $\angle(\cdot, \cdot)$ is the angle measured after flattening the pair of triangles ijk, jil apart their common edge ij . The main difficulty of this step is to evenly split this curvature along the two parts of the dual edge. We chose to split according to the height ratio above the common edge :

$$\begin{aligned} \alpha(\mathbf{g}(ijk), \mathbf{g}(ij)) &= \frac{d(\mathbf{g}(ijk), ij)}{d(\mathbf{g}(ijk), ij) + d(\mathbf{g}(jil), ji)} = \\ &\frac{\|[\mathbf{ig}(ijk)] \times [ij]\|}{\|[\mathbf{g}(jil)\mathbf{g}(jik)] \times [ij]\|} = \frac{\|[\mathbf{ik}] \times [ij]\|}{\|[\mathbf{kl}] \times [ij]\|} \end{aligned} \quad (8.3)$$

where $d(P, e)$ is the distance from point P to edge e . This gives a natural linear interpolation for $P \in [\mathbf{g}(ijk)\mathbf{g}(ij)]$ given in barycentric coordinates $P = (1-t)\mathbf{g}(ijk) + t\mathbf{g}(ij)$:

$$\theta(P) = \theta(f) + \Delta\theta(e)\alpha t \quad (8.4)$$

On the other hand, the direction interpolation will be more tricky. \mathbf{d} is parameterized by its angle $\theta \in \mathbb{R}$. As we said earlier, the fact that we take the angle in \mathbb{R} and not in $(-\pi, \pi]$ is very important. It allows the continuity of a direction on a triangle to be defined, even if the direction rotates by more than 2π between two discretization points (facet centers). This choice gives us more freedom for the interpolation, but generates an ambiguity. The period jump solves for this ambiguity by making explicit the quantity of rotation of the direction between discretization point.

For the norm, the same considerations lead to a linear interpolation of the form :

$$r(t) = r(f) + \Delta r(e)\alpha t \quad (8.5)$$

with $\Delta r(\star ij) = r(jil) - r(ijk)$

8.2.3 Step 3 : 2D

In this last step, we interpolate the vector field over the whole surface mesh using a variant of *side-vertex* interpolation [Nielson 1979]. The interpolation is defined piecewise on the subdivision simplices $S_{i,j,k}$ of the mesh M . The subdivision simplices are simply defined as the triangles

$$S_{i,j,k} = [\mathbf{g}(ij)\mathbf{g}(ijk)] \quad \forall (i, j, k) | ijk \in \mathcal{F} \quad (8.6)$$

Notice that because of the possible permutations for (i, j, k) , there are six subdivision simplices per facet of the mesh (see Figure 8.4). On a subdivision simplex $S_{i,j,k}$, the field is only known on the edge $[\mathbf{g}(ij)\mathbf{g}(ijk)]$. It is then interpolated on the whole $S_{i,j,k}$ such that the vector is constant along each segment between a point of $[\mathbf{g}(ij)\mathbf{g}(ijk)]$ and the primal vertex $[i]$. Another way of saying that is that we obtain the vector at a point \mathbf{x} as the vector at the intersection \mathbf{x}' between $([i], \mathbf{x})$ and $[\mathbf{g}(ij)\mathbf{g}(ijk)]$ (see Figure 8.5). This interpolation is a variant of “side-vertex” interpolation scheme [Nielson 1979]. In our case, the value along the side is interpolated linearly while it is constant (identical to the side value) along the side-vertex paths. This has a very simple expression in barycentric coordinates. If we write :

$$\mathbf{x} = (1-t')[i] + t'((1-t)\mathbf{g}(ij) + t\mathbf{g}(ijk))$$

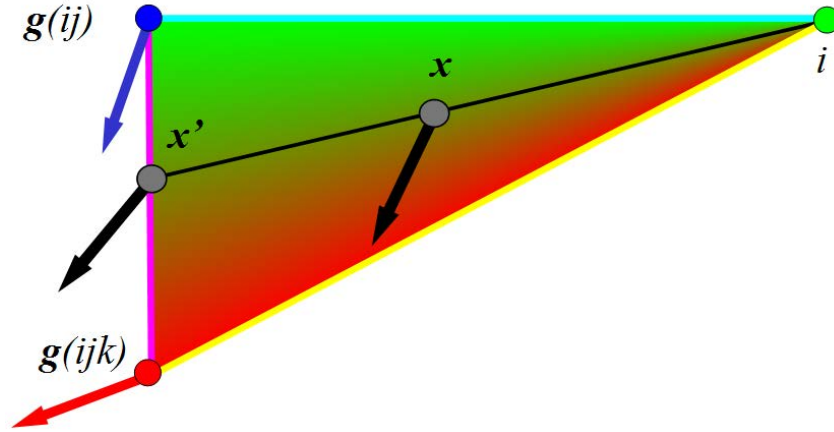


FIG. 8.5 – Our “side-vertex” interpolation over a subdivision simplex : we obtain the interpolation value at \mathbf{x} as the value at its projection \mathbf{x}' on the (magenta) dual edge.

then we have :

$$\mathbf{x}' = (1-t)\mathbf{g}(ij) + t\mathbf{g}(ijk) \quad (8.7)$$

and we obtain the vector direction and norm by the interpolation described in Equations (8.4) and (8.5).

This interpolation scheme has two advantages :

1. It is straightforward to implement on a GPU (see Section 8.4).
2. It allows arbitrary singularities to be represented on the vertices of the mesh.

However, it has a major drawback : it is discontinuous at vertices (curvature is unbounded near vertices). In the following section, we propose a modified interpolation scheme to correct this issue.

8.3 Corrected interpolation

This section proposes a correction for the interpolation scheme of the previous section. We chose not to present this scheme alone as it is more complex than the previous one, and we needed to present important notions on direction field interpolation in a simpler context to be more clear.

The previous scheme has the important limitation of presenting high distortions at vertices even in the absence of singularities. Because of angle defect, continuity of a direction field at a vertex cannot be defined directly on the mesh, but on an exponential map of the neighborhood of each vertex.

Définition 23 (Exponential map) *Let i be a vertex and ij a reference edge emanating from i . Any point \mathbf{x} in the vicinity of i can be parameterized by the distance $r_i(\mathbf{x})$ between i and \mathbf{x} the angle $\theta_i(\mathbf{x})$ between $[ij]$ and $[i]\mathbf{x}$ (measured on an isometric flattening). Then the exponential map of i relative to $[ij]$ is the mapping :*

$$Exp_i(\mathbf{x}) = \begin{pmatrix} u = r \cos(2\pi\theta_i(\mathbf{x})/\theta_i^{sum}) \\ v = r \sin(2\pi\theta_i(\mathbf{x})/\theta_i^{sum}) \end{pmatrix}$$

where θ_i^{sum} is the sum of angles adjacent to i .

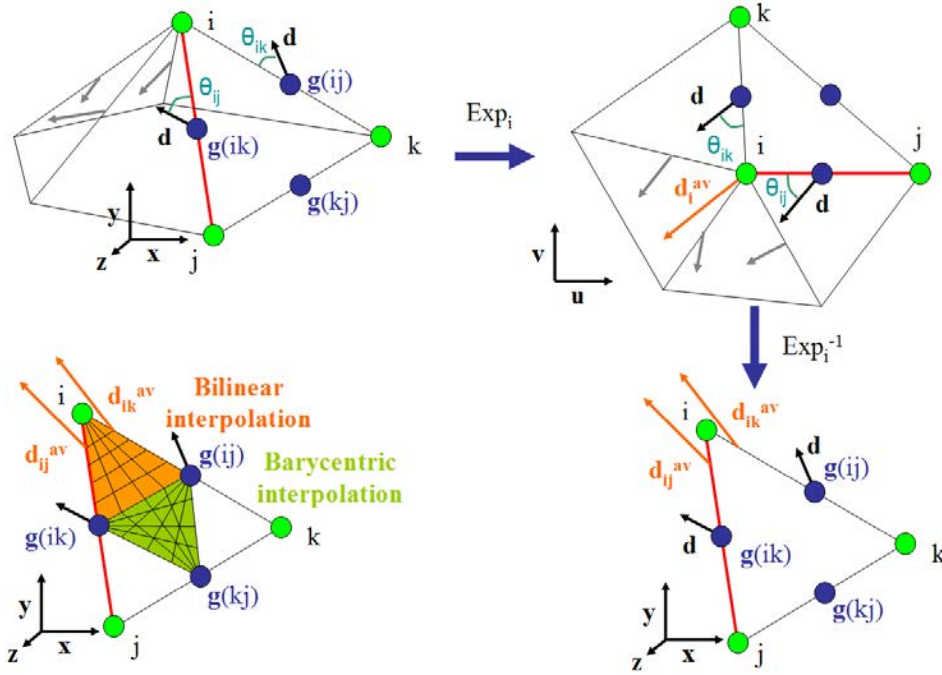


FIG. 8.6 – Corrected interpolation scheme : find an average direction \vec{d}_i^{av} in an exponential map, then interpolate bilinearly in a corner triangle. Two corners of the bilinear patch are mapped at the same vertex i , but correspond to two (different) inverse images of \vec{d}_i^{av} .

An exponential map is not linear, however it keeps the edges emanating from i straight. Thus the image of a direction defined on an edge is well defined in the exponential map by the same angle it had with the original edge (see Figure 8.6). On each edge ik , we know the angle θ_{ik} between the direction \vec{d} and $[ik]$ at the edge middle $g(ik)$ by applying (8.4). Its image by the exponential map is defined based on $Exp_i(ik)$ by the same angle. We can then compute the angle θ_{ik}^u between $Exp_i(\vec{d})$ and the u axis (image of ij) by adding the angle between ik and ij in the exponential map. We can then define the average direction \vec{d}_i^{av} at i in the exponential map by averaging the θ_{ik}^u . This \vec{d}_i^{av} form an angle θ'_{ik} with each edge ik .

The backwards image of \vec{d}_i^{av} by the exponential map is then defined for each edge ik by the angle θ'_{ik} taken on the mesh. Finally, the interpolation is defined on triangles $[ig(ij)g(ik)]$ by bilinear interpolation of the angle, where i is taken twice, once with θ'_{ij} and once with θ'_{ik} . The direction is interpolated using barycentric coordinates on the final triangle $[g(ij)g(jk)g(ki)]$

Instead of the previous interpolation scheme, this corrected scheme requires to cut the triangles in four instead of six. On the central triangle, the interpolation is simply barycentric, and on the three outer triangles, its is bilinear, with two different values at the vertex. For the interpolation of the norm, we simply do a barycentric interpolation on each of the four triangles, with the value at the vertex defined as the average values on edges. As illustrated in Figure 8.7, this scheme offers much more regularity, especially in regions where the field is highly curved, such as in the vicinity of high order singularities.

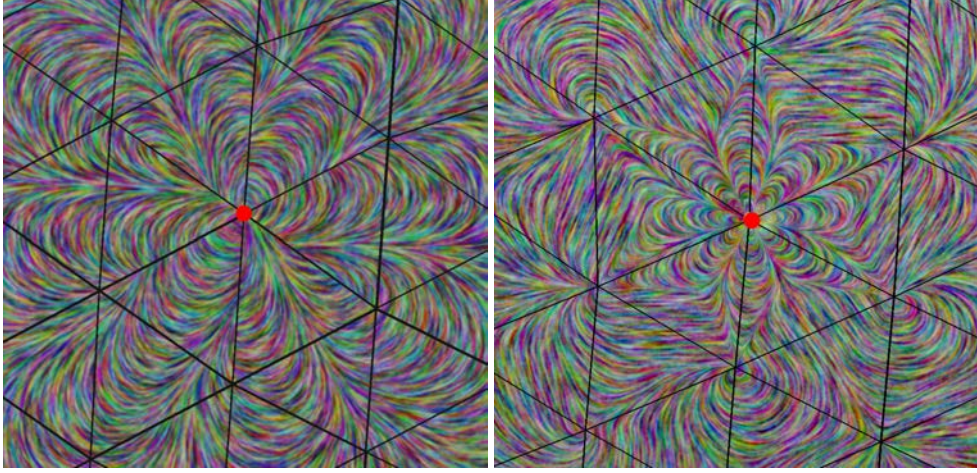


FIG. 8.7 – Corrected interpolation scheme (left) offers much more regularity than the side-vertex interpolation (right).

8.4 The GPU-based Visualization Algorithm

In this section we present an algorithm to visualize our vector fields. We adopted a GPU-accelerated LIC-based approach working in image space, which is inspired by the Image Space Advection method proposed in [Laramee et al. 2003]. The algorithm is done in three passes (see Figure 8.8).

8.4.1 Pass 1 : Depth Value for Geometric Discontinuities

One of the disadvantages of decoupling the LIC process with the 3D surface geometry is the undesired visual continuity across the geometric discontinuities. This problem is due to the projection of a self-occluding surface onto the image plane (as pointed out in [Laramee et al. 2003]). We adopted the criteria to distinguish the geometric discontinuities proposed in [Laramee et al. 2003] as follows.

$$\|z_{i+1} - z_i\| > \varepsilon \|\mathbf{p}_{i+1} - \mathbf{p}_i\| \quad (8.8)$$

where ε is a threshold chosen to be 0.15 for the results shown in this chapter. The test compares the depth values in the object space z_i and z_{i+1} of two consecutive points \mathbf{p}_i and \mathbf{p}_{i+1} along the integral path in the image plane. A positive result of the test identifies a geometric discontinuity. To allow performing the above test in the LIC pass, the Z-buffer is rendered to the frame buffer and stored as a texture, T_{depth} .

8.4.2 Pass 2 : Object to Image Space Projection

Since the LIC is performed in the image space, the vectors at the points along the forward and backward line integral path need to be known in advance. We achieved this by storing the vector field of the surface as follows. First, for each visible point of the surface, we calculate its vector using the interpolation scheme as described in Section 7.4. We only interpolate the direction since we are only using the directional component of the vector for the LIC pass. The norm of the vectors can also be interpolated to give more effects of the visualization as done in [Cabral and Leedom 1993]. More specifically, in our GPU implementation, we associate (1,0,0), (0,1,0) and (0,0,1) to the vertices of the subdivision simplex

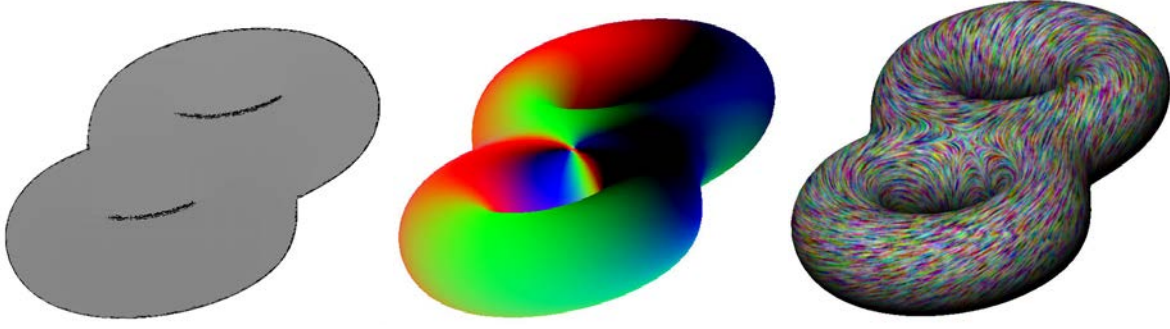


FIG. 8.8 – The three passes of the LIC on GPU. **Left** : the geometric discontinuities identified. **Middle** : vectors on the surface encoded in colors. **Right** : result of the LIC process on a vector field with a singularity of index -2.

$\mathbf{g}(ijk)$, $\mathbf{g}(ij)$ and $[i]$ respectively. The direction at a point of barycentric coordinates $(1 - \lambda_1 - \lambda_2, \lambda_1, \lambda_2)$ in the subdivision simplex (except at $[i]$, where it is undefined) is given by the angle of rotation, θ measured with respect to the base direction $\mathbf{r}(f)$. Using Equations (8.7) and (8.4), the θ value is given by $\theta(f) + \kappa_1 \alpha \lambda_1 / (1 - \lambda_2)$. Then, the obtained vector in the object space is projected onto the image plane, then normalized. The vector in the image space is rendered to the frame buffer as color, which will be resampled as a texture, T_{vector} , in the third pass.

8.4.3 Pass 3 : LIC in the Image Space

Brief review : LIC (Line Integral Convolution) was proposed by Cabral and Leedom [Cabral and Leedom 1993] to perform a texture synthesis for the visualization of 2D vector fields. The basic idea is that the local behavior or the vector field can be approximated by tracing a streamline starting from a point in its positive and negative directions. By convolving an input white noise texture T_{noise} with a low-pass filter $\tau(w)$ along this streamline, the pixel intensity is highly correlated along individual streamlines but independent in the perpendicular direction. Denoting a streamline $\gamma(s)$, line integral convolution gives the intensity for a pixel $x_0 = \gamma(s_0)$

$$Intensity(x_0) = \int_{s_0-L}^{s_0+L} \tau(s - s_0) T_{noise}(\gamma(s)) ds \quad (8.9)$$

Now, we present how we carry out LIC in the image space using a programmable GPU as follows : starting from the projection \mathbf{p}_0 onto image space of a visible point of the surface, a streamline is traced in the positive and negative directions by coordinate advection using Equation 8.10 :

$$\begin{aligned} \mathbf{p}_i &= \mathbf{p}_{i-1} + \vec{d}_{img}(\mathbf{p}_{i-1}) \Delta s \\ \mathbf{p}'_i &= \mathbf{p}'_{i-1} - \vec{d}_{img}(\mathbf{p}'_{i-1}) \Delta s \text{ where } \mathbf{p}'_0 = \mathbf{p}_0 \end{aligned} \quad (8.10)$$

where $\vec{d}_{img}(q)$ is the vector in the image plane obtained by resampling the vector image T_{vector} obtained in the first pass.

The time step Δs is defined such that $n \cdot \Delta s = L$, where n is the number of steps and $L = 1/10$, in both directions. We use only the direction of the vector as done in [Cabral and Leedom 1993] and we found that the results given are satisfactory. As for the convolution kernel, we chose to use a rather simple low-pass filter, which is of value $1/2L$ in $[-L, L]$ and zero elsewhere. Namely, the average of

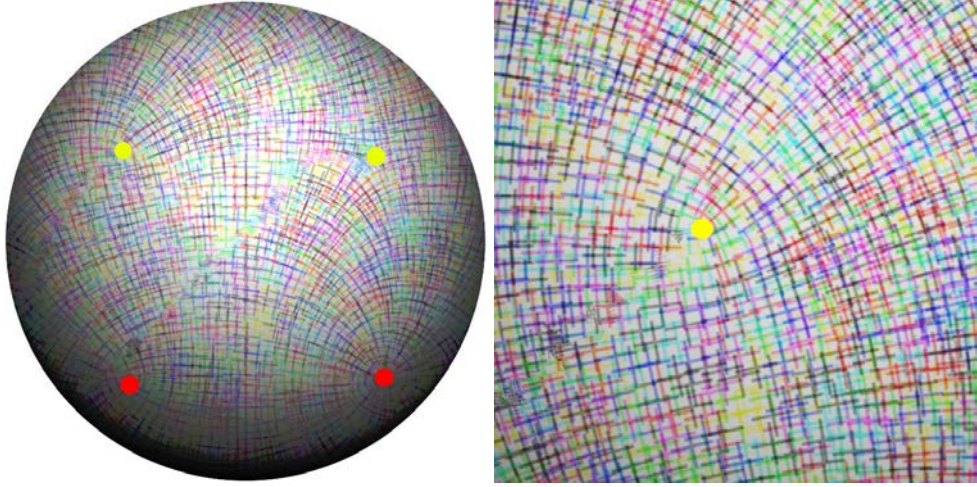


FIG. 8.9 – **Left** : indexes of the singularities from left to right and from top to bottom are $1/4$, $-1/4$, $1/2$ and 1 . **Right** : a zoom in on the singularity of index $1/4$.

the $2n + 1$ sample of the T_{noise} along the streamline. We chose this box filter basically for the reason of performance. To implement more complicated kernels such as the Hanning filter as done in [Cabral and Leedom 1993], one needs more precomputation and extra storage of a one-dimensional texture (as discussed in [Grabner and Laramée 2005]).

The integral in a direction (positive or negative) stops at q_i whenever the position q_i gives a positive result to Equation (8.8) by resampling the depth information, T_{depth} , stored in the second pass. Random noise value is assigned to the unsampled points on the rest of the streamline. This generates visual discontinuities at geometric discontinuities due to self occlusion.

Before presenting the results, we feel it is important for the practicality of our structure to explain how to convert between existing representations for vector field and ours. Note that most existing representations are less general than ours as they do not handle field symmetries. Our method is also specialized to handle directions, even if the extension to tangent vectors is straightforward through the introduction of the scalar $r(f)$ on each facet representing the vector norm.

8.5 Results, Discussions and Conclusions

We have shown examples of vector fields that are represented by our construction on several types of surface. All the examples are visualized by the LIC-based algorithm that we presented, which is implemented on a machine with a 1.7 GHz Pentium processor and 1 GB of RAM equipped with a GeForce FX5650 graphics card. We achieved 15 *frames/s* to visualize the Statue (50k facets) at resolution 800×600 .

In every example that we have shown, using our representation, singularities can only appear on vertices. Figure 8.2 shows the comparison of our representation and the *classic* piecewise linear vector field. Note that our interpolation scheme is C^0 smooth except at vertices. It is not C^1 , but our choices lead to continuity of the derivative across edges in direction perpendicular to the crossed edge.

We also show examples of vector fields defined on surfaces of various genus. Figure 8.2, Figure 8.1 and Figure 8.9 show genus-0 surfaces, Figure 8.8 shows a genus-2 surface and Figure 8.10 shows a genus-3

surface with more geometric detail. Among the examples, Figure 8.1 is especially interesting. The reason is that one can see that curvilinear sectors in the vicinity of a higher-order singularity correspond to the three basic sector types. Moreover, it shows that the complexity of the singularities that a vertex can represent is not limited by its connectivity. Finally, the index that we have defined in Equation (7.18) verifies the Poincaré-Hopf index theorem as we have proved in [Ray et al. 2008b]. One can check for instance in Figure 8.1 that the indexes of the two singularities of the sphere sum to $5 - 3 = 2 = \chi(g = 0)$, where g is the genus and χ is the Euler characteristic of the surface.

Our representation also enables singularities with fractional indexes to be represented. This is achieved by using period jumps multiple of $1/2$ or $1/4$ instead of integers. Examples are shown in Figure 8.1, Figure 8.2-bottom and Figure 8.9 for the three cases respectively.

In term of memory consumption, vertex-based vector field encoding schemes on surfaces require three real numbers (4 bytes) to be stored per vertex. Our facet-based encoding requires two real numbers to be stored per facet corresponding to the angle and norm, and one integer (1 byte) per edge corresponding to the period jump. If we denote V the number of vertices, and considering that the number of facets (resp. edges) of a mesh is around 2 (resp. 3) times the number of vertices, our approach requires around $(2 \times 4 \times 2 + 2 \times 3)V = 22V$ bytes to be stored while it needs around $(3 \times 4)V = 12V$ bytes in the classic vertex-based encoding scheme. This difference is explained by the fact that our sampling is on facet barycenters, which are roughly twice as dense as vertices.

Limitations

We have adopted the side-vertex interpolation scheme for efficiency and simplicity considerations, but also because it allows a nice visualization of singularities of arbitrary index. The drawback of this choice is that field discontinuities may appear on non-singular vertices. However, the visual quality is affected only when one zooms in very closely to a vertex or on a coarse mesh where a vector field with high variations is defined. In such cases, one can use other types of interpolation at non-singular vertices to make the field smoother. This will be one of the directions in the future work.

With our representation, singularities can only occur at vertices, whereas in piecewise linear vector fields, first-order singularities may appear anywhere on the facets. This is more an advantage than a flaw since it gives direct control over the singularity placement. Moreover, as stated in the previous section, we can allow a singularity anywhere on the mesh by simply creating a new vertex at the desired singularity position and splitting the corresponding facet in 3.

Conclusions and Future Work

In this chapter, we have introduced a concise yet complete representation of vector fields on triangulated surfaces with arbitrary topology. The representation enables us to capture or represent higher-order singularities in a given continuous vector field, for example using a FEM function basis. The vector field is described in polar coordinates on each facet, with a facet edge being chosen as reference to define the angle. We have introduced an integer called the period jump, which is associated to each edge of the triangulation to remove the ambiguity in interpolating the angle between two facets sharing an edge. The field is interpolated over the whole surface by using a variant of side-vertex interpolation, which combined with the notion of period jump removes the bound on the complexity of singularities that a vertex can represent imposed by its connectivity. Since the singularities can only be located on vertices and their indexes can be found combinatorily, the tracking of singularities over time becomes easier as

compared to traditional methods that use numerical detection. To visualize the so-defined vector field at interactive frame rate, we have presented a LIC-based GPU-accelerated algorithm operating in image space.

There are several directions for the future work. Most naturally, the representation may be extended to represent 3D vector fields with higher-order singularities. Besides, by adopting our representation, one can carry out visualization of dynamic flow on the surfaces. Finally, as discussed in the previous section, a different interpolation scheme can be studied for non-singular vertices.

In the next chapter, we will present a direction field design method based on the "period jumps" based structure introduced in Chapitre 7 and used in this chapter. Thus the visualization method proposed here is used to get feedback on the design.



FIG. 8.10 – A vector field defined using our representation on a genus-3 surface. Yellow dots are first-order singularities and red dots are higher-order singularities.

Manipulation de champs de directions avec contrôle topologique

Sommaire

9.1 Zipping algorithm : Constraining the topology	148
9.1.1 Problem setting	148
9.1.2 Edge classification	149
9.1.3 Topological issues	150
9.2 Constrained direction field smoothing	152
9.2.1 Problem formulation	152
9.2.2 Algorithm	152
9.3 Results and conclusion	153
9.3.1 Results	153
9.3.2 Conclusion	154

Introduction

Nous présentons dans ce chapitre un algorithme qui permet de contrôler explicitement toute la topologie d'un champ de directions discret dans la structure proposée au Chapitre 7. A partir d'un ensemble de singularités définies par l'utilisateur, nous proposons un algorithme qui construit un champ de directions lisse ayant exactement ces singularités. Les directions de ce champs peuvent alors être modifiées pour rajouter des contraintes géométriques. Si les indices des singularités définies par l'utilisateur vérifient le théorème de Poincaré-Hopf (leur somme égale la caractéristique d'Euler $(2g - 2)$, ou si la surface a au moins un bord libre, la construction est garantie de ne générer aucune autre singularité (Section 9.1). Cet algorithme est le seul à notre connaissance à garantir un tel contrôle de façon prouvable.

Plan du chapitre

Nous commencerons par décrire l'algorithme du "Zipping" (Section 9.1) qui contraint la topologie du champ de directions en fixant les "sauts de périodes". Ensuite, on présentera dans la Section 9.3 l'algo-

rithme de lissage qui construit un champs lisse à topologie contrainte, et permet l'édition de ce champs à topologie fixée.

Contributions

- Nous proposons un algorithme appelé "Zipping" qui calcule les "sauts de période" permettant de contrôler les singularités d'un champ de directions. Les autre degrés de liberté topologiques que nous avons exhibés dans le Chapitre 7 sont laissés libres.
- Nous proposons un algorithme de lissage de champs de directions à topologie fixée. L'utilisateur peut ensuite le manipuler en imposant certaines directions.
- La combinaison de ces algorithmes fournit un outil de manipulation de champs de directions qui offre à la fois un contrôle topologique prouvé et un contrôle géométrique.

Les algorithmes décrits dans ce chapitre apparaissent dans l'article "*N*-Symmetry Direction Fields on Surfaces of Arbitrary Genus" [Ray et al. 2008b] qui est paru dans "ACM Transactions On Graphics". Ils ont été conservé dans leur langue originale, mais on a appliqué le formalisme plus concis du calcul extérieur discret pour les formules, et assuré la cohérence avec le reste de cette thèse (et en particulier l'étude théorique du Chapitre 7 qui figure aussi dans [Ray et al. 2008b]).

9.1 Zipping algorithm : Constraining the topology

We propose in this section an algorithm called Zipping which computes the period jumps such that the singularities and their indexes are exactly the ones defined by a user. Recall that in our direction field representation, we can choose the period jumps $p_1(e)$ on each dual edge, which will set the topology of the represented direction field. However, the period jumps control all Topological Degrees of Freedom (TDoFs), including those not corresponding to singularities, such as turning numbers of generators (see Figure 7.7 right). We choose to let these TDoFs free such that an optimization can play on these additional TDoF to find a smoother solution. As a result the algorithm will not compute explicitly all period jumps, but find some free period jumps and express all the other period jumps as a combination of the free ones.

9.1.1 Problem setting

The problem we tackle in this section is the following :

Given constrained singularity indexes $I_2^c(v)$ at each interior vertex v of the mesh, find the integer 1-form p_1 such that :

$$I_2(v) = I_2^c(v) \quad \forall v \quad (9.1)$$

Using (7.18), the above expression can be rewritten as :

$$dp_1 = N(I_2^c - I_2) = \Delta I_2 \quad (9.2)$$

in other words, we are trying to integrate an integer 2-form ΔI_2 into an integer 1-form p_1 . We could simply do that by inverting the matrix of d in a DEC setting. However the kernel of d is of high dimension, which means that there are many solutions to (9.2), such that this problem is not well defined. Moreover, we would have no guarantee that the solutions are integer. Hence, we rather propose a greedy algorithm to compute a solution to (9.2) in linear time, which works by classifying edges into 3 sets.

9.1.2 Edge classification

As in the continuous setting, the homology basis $H([M])$ of $[M]$ consists of $2g$ generators and $b - 1$ borders (we will discuss later the case $b = 0$). As the turning numbers of the cycles in $H([M])$ do not correspond to singularities, we will leave the turning numbers of these cycles free. As the direction cannot be defined at vertices, the direction field is defined over $[M]_h$, which is the embedding $[M]$ of M with holes at every interior vertices v allowing a singularity to occur at v . Hence $[M]_h$ has a homology basis $H([M]_h) = H([M]) \cup \bigcup_{v \neq v_{last}} \{\partial v^*\}$. Each removed interior vertex v adds a cycle ∂v^* around it to the homology basis, except one of them that can be chosen arbitrarily (we call it v_{last}). The turning numbers along these additional cycles in $H([M]_h)$ correspond to singularities located on interior vertices, so we want to control them explicitly. Hence, the basic idea in Zipping is to progressively classify the dual edges in three sets :

1. The set \mathcal{E}_0^* of *null* edges, whose associated period jumps we can set to 0 without constraining any turning number. Hence we have $p_1(e_0^*) = 0$
2. The set \mathcal{E}_{free}^* of *free* edges, whose associated period jumps correspond to the turning number of a cycle homological to a cycle in $H([M])$, hence corresponding to a TDoF but not to a singularity (see Figure 9.3). We choose the name *free* because these other TDoF will be left free in an optimization.
3. The set \mathcal{E}_{dep}^* of *dependent* edges (they may depend on free edges), whose associated period jumps constrain the turning numbers of cycles ∂v^* around a single vertex v , hence constraining the index of a the singularity at v .

Zipping is a greedy algorithm which makes this iterative classification, and computes an expression of the period jumps of edges in \mathcal{E}_{dep}^* as a function of the free period jumps of edges in \mathcal{E}_{free}^* , such that the indexes have their constrained value. This way, period jumps in \mathcal{E}_{dep}^* take the general form :

$$p_1(e_{dep}^*) = p_1^0(e_{dep}^*) + \sum_{e_{free}^* \in \mathcal{E}_{free}^*} c(e_{dep}^*, e_{free}^*) \quad (9.3)$$

where p_1^0 is a 1-form that is non-trivial only on \mathcal{E}_{dep}^* (it is 0 on \mathcal{E}_0^* and \mathcal{E}_{free}^*). c can be represented as a $(|\mathcal{E}_{dep}^*|, |\mathcal{E}_{free}^*|)$ matrix of the following form (with the ordering $\mathcal{E}_0^*, \mathcal{E}_{dep}^*, \mathcal{E}_{free}^*$) :

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & C \\ 0 & 0 & Id \end{pmatrix}$$

- Period jumps in \mathcal{E}_0^* are zero so they depend on nothing and nothing depends on them.
- Period jumps in \mathcal{E}_{dep}^* depend only on free period jumps.
- Period jumps in \mathcal{E}_{free}^* depend only on themselves.

where only C is non trivial. Hence, the Zipping algorithm proceeds in the following way.

1. Fill \mathcal{E}_0^* : While it is possible, add to \mathcal{E}_0^* edges e_0^* that do not close any cycle, and set their period jumps to 0. This is equivalent to building a spanning tree of the dual graph G^* .
2. Fill \mathcal{E}_{dep}^* : While it is possible, add to \mathcal{E}_{dep}^* edges that close dual cells v^* (cycles around a single primal vertex v) and computes the corresponding period jumps such that they verify (9.2) :

$$p_1^0(e_{dep}^*) = \Delta I(v) - \sum_{e^* \in \partial v^* \setminus e_{dep}^*} p_1^0(e^*), \quad c(e_{dep}^*, e_{free}^*) = - \sum_{e^* \in \partial v^* \setminus e_{dep}^*} c(e^*, e_{free}^*) \quad (9.4)$$


```

Build a recovering tree  $\mathcal{E}_0^*$  of  $G^*$  // grow black edges
 $\forall e_0^* \in \mathcal{E}_0^*$  set  $p_1^0(e_0^*) \leftarrow 0, c(e_0^*) \leftarrow 0$ 
 $\mathcal{E}_{dof}^* \leftarrow \mathcal{E}^* \setminus \mathcal{E}_0^* \quad \mathcal{V}_{zip} \leftarrow \mathcal{V}$ 
 $i_{free} \leftarrow 0$  // Number of free variables found
while  $\mathcal{E}_{dof}^* \neq \emptyset$  do
  while  $\mathcal{V}_{zip} \neq \emptyset$  do
    Take  $v_z \in \mathcal{V}_{zip}$  and remove it from  $\mathcal{V}_{zip}$ 
    if  $v_z \notin \partial M$  and  $\exists!$  unset edge  $e_z^* \in \partial v_z^*$  then
      // zip red (then blue) edges
      Move  $e_z^*$  from  $\mathcal{E}_{dof}^*$  to  $\mathcal{E}_{dep}^*$ 
      Set  $p_1^0(e_z^*)$  using (9.4)
       $\forall e_{free}^* \in \mathcal{E}_{free}^*$  set  $c(e_z^*, e_{free}^*)$  using (9.4)
      Add the face opposite to  $v_z^*$  across  $e_z^*$  to  $\mathcal{V}_{zip}$ 
    end if
  end while
  // free green edge
   $i_{free} \leftarrow i_{free} + 1$ 
  Take  $e_{free}^* \in \mathcal{E}_{dof}^*$  and move it to  $\mathcal{E}_{free}^*$ 
  Set  $p_1^0(e_{free}^*) \leftarrow 0, c(e_{free}^*, e_{free}^*) \leftarrow 1$ 
  Add the 2 faces adjacent to  $e_{free}^*$  to  $\mathcal{V}_{zip}$ 
end while

```

FIG. 9.1 – Zipping algorithm (see Figure 9.2)

3. Add a remaining edge to \mathcal{E}_{free}^* : any remaining edge necessarily closes a cycle (else it would have been added to \mathcal{E}_0^*) which does not enclose a single vertex (else it would have been added to \mathcal{E}_{dep}^*). Hence this cycle is homological to a border or to a generator of the mesh, which turning numbers we want to keep free, so any such edge can be added to \mathcal{E}_{free}^* .
4. After freeing an edge, it becomes possible again to find edges that close dual cells, such that steps 2 and 3 may be iterated until no edge remain. The period jumps computed in step 2 will depend on the period jumps of the edges that have been freed in a step 3.

Please see Figure 9.1 for the whole Zipping algorithm.

9.1.3 Topological issues

If M has no boundary, we have :

$$\sum_{\mathcal{V}} \Delta I_2(v) = \int_M \Delta I_2 = \langle \Delta I_2 | M \rangle = \langle dp | M \rangle = \langle p | \partial M \rangle = 0$$

As base field singularities verify the Poincaré-Hopf theorem 7.3.1 ($\sum_{\mathcal{V}} I_{\vec{d}_0}(v) = 2 - 2g$), this means that on v_{last} , we have :

$$\Delta I_2(v_{last}) = - \sum_{\mathcal{V} \setminus v_{last}} \Delta I_2(v) \quad , \quad I_c(v_{last}) = 2 - 2g - \sum_{\mathcal{V} \setminus v_{last}} I_c(v) \quad (9.5)$$

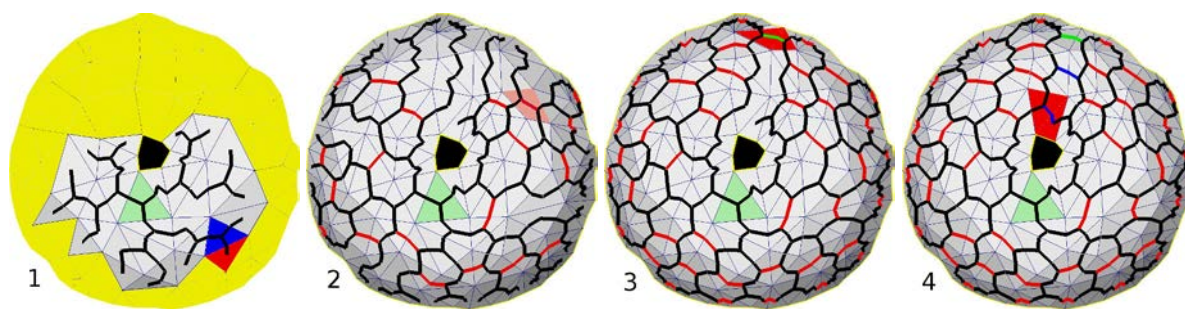


FIG. 9.2 – Zipping : 1-**Grow** black edges (width first search) 2-**Zip** red edges 3-**Free** green edge 4-**Zip** blue edges (blue edges depend on the freed green edge)

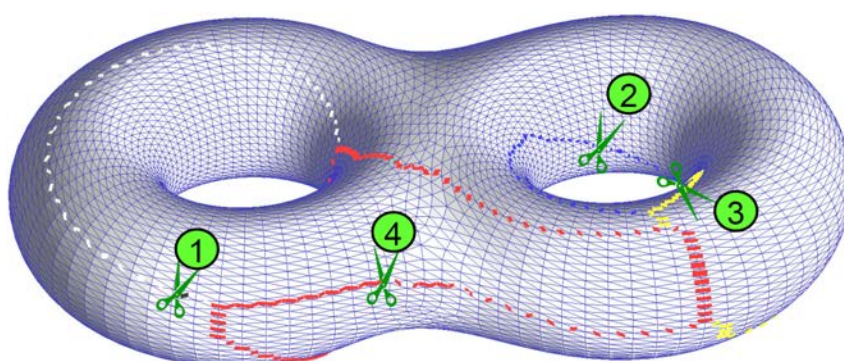


FIG. 9.3 – A Genus g surface without boundary requires $2g$ edges to be freed. The image shows in different colors the edges which period jumps effectively depend on free period jumps (scissors).

Hence the index of v_{last} adapts to ensure the Poincaré-Hopf theorem is verified. If the indexes have been constrained such that :

$$\sum_{\gamma} I_c(v) = 2 - 2g$$

then this last index will be 0. As the position of v_{last}^* depends on some choices made arbitrarily in the greedy algorithm, it cannot be easily determined, hence it is highly recommended to run the algorithm with constrained indexes which sum up to $2 - 2g$ to avoid the appearance of a random (but necessary) singularity.

If M has boundary, they are also handled by the Zipping algorithm, but their turning numbers are not constrained. In fact, we can also leave some vertices unconstrained by simply declaring them as border vertices before Zipping. However, in this case we cannot guarantee that no undesired singularity will appear, but only authorize where they can appear. A simple way to ensure that no singularity appear when M has a boundary, is to run our Zipping algorithm on M with its holes triangulated. The index of a border on such a field will be equal to the sum of the indexes of the border vertices. In this case again, no singularity will appear if and only if the sum of indexes of constrained vertices (including vertices) add up to $2 - 2g$.

9.2 Constrained direction field smoothing

Based on the "Zipping" algorithm, we now show a simple use of our structure to design smooth direction fields with strong topological constraints.

9.2.1 Problem formulation

Using the notions above, the problem we want to solve is :

Given :

- a mesh $M = \langle \mathcal{V}, \mathcal{E}, \mathcal{F} \rangle$
- a constrained singularity index $I_c(v)/N$ given on each vertex $v \in \mathcal{V}$.
- a constrained direction $\vec{d}_c(f)$ given on each facet of a subset $\mathcal{F}_c \subset \mathcal{F}$.

Interpolate the constraints in the smoothest way possible. More formally, minimize :

$$E_{G^*}(\theta_0, p_1) = \|\kappa_1\|^2 = \sum_{e^* \in \mathcal{E}^*} \kappa_1^2(e^*) = \sum_{e^* \in \mathcal{E}^*} \left(d\theta_0(e^*) + \kappa_1^r(e^*) + \frac{2\pi p_1(e^*)}{N} \right)^2 \quad (9.6)$$

subject to the constraints :

$$I_{\vec{d}}(v) = I_{\vec{d}_0}(v) + \sum_{e^* \in \partial v^*} (e^*) = I_c(v) \quad \forall v \in \mathcal{V} \quad (9.7)$$

$$\vec{d}(f) = \vec{d}_c(f) \quad \forall f \in \mathcal{F}_c \quad (9.8)$$

As told in previous section, we build a representation for the direction field which implicitly enforces the topological constraints, by expressing all the period jumps $p(e^*)$ as a function of a limited number of free period jumps. We now describe a design algorithm which uses this structure to reduce the problem to a simple quadratic minimization procedure.

9.2.2 Algorithm

The design algorithm links between user inputs and Zipping inputs by computing the indexes of the base field and making the difference with user constrained indexes, and uses the output of the Zipping algorithm to ensure that the field topology is constrained during the creation of the direction field. The minimization problem is then solved in two passes :

1. Minimize E_{G^*} with respect to θ and p by assuming that p is continuous
2. Minimize with respect to θ only, with the p being set to their rounded value of the first pass

We use for both passes a standard formula to solve the problem of minimizing $([A_f, A_l][x_f, x_l] - B)^2$ where x_f are variables and x_l are set :

$$x_f = (A_f^t A_f)^{-1} A_f^t (B - A_l x_l) \quad (9.9)$$

This method is not guaranteed to find the global minimum with respect to the discrete variables $p_1(e^*)$, but offers good results in practice that verify all the constraints. For the continuous variables $\theta_0(f)$, if at least one directional constraint is set, A_f is of maximal rank, therefore, by Gramm's theorem, $A_f^t A_f$ is non-degenerate and our algorithm finds the unique minimum. We also noticed that we could improve the visual aspect of the direction field near the constrained directions by adding a Laplacian smoothing term to the energy (9.6).

Algorithm 1 Design algorithm

1. Set $I_2^c(v) = 0$ on $\mathcal{V} \setminus \mathcal{V}_c$ // no singularities outside \mathcal{V}_c
2. Choose a direction $\mathbf{r}(f)$ on each facet // base field
3. Compute $\kappa_1^r(e^*)$ as the angles between $\mathbf{r}(f_1)$ and $\mathbf{r}(f_0)$ for each dual edge f_0f_1 .
4. Compute the angle defects $K(v)$ at all vertices.
5. Compute the base field indexes $I_2^r(v)$ using (7.18)
6. Compute $\Delta I_2(v) = N(I_2^c(v) - I_2^r(v))$
7. Apply the Zipping Algorithm
8. Build the linear system $[A_f, A_l, C][\theta_f, \theta_l, p_{free}]^t = B$ corresponding to (9.6). Each line of the system corresponds to an edge $e^* : [A_f, A_l]$ is the signed adjacency matrix with a proper ordering (it contains +1 and -1 at the indexes corresponding to the dual vertices at the 2 extremities of e^*), C contains $c(e^*, e_{free}^*)$. B contains the $\kappa_1^r(e^*) - p_1^0(e^*)$.
9. Pass 1 :

$$[\theta_f^1, p_1^{free}]^t = ([A_f, C]^t [A_f, C])^{-1} [A_f, C]^t (B - A_l \theta_l)$$
10. Pass 2 :

$$[\theta_f^2]^t = (A_f^t A_f)^{-1} A_f^t (B - [C, A_l][rnd(p_1^{free}), \theta_l]^t)$$

where *rnd* is the rounding to the nearest integer value.
11. Apply rotations θ^2 to \mathbf{r} to get a direction \vec{d} on each facet of M .

9.3 Results and conclusion

9.3.1 Results

Our framework allows direction fields on a surface to be created and edited via topological and geometric constraints. As illustrated in Figure 9.4, a few constraints only are required to create the desired field. The same picture also demonstrates that our method is not affected by the complex geometry of the hair of the David, whereas the previous methods only based on relaxation ([Ray et al. 2006] or [Hertzmann and Zorin 2000]) are trapped by this geometry and generate many singularities in this zone.

We have also tested this algorithm on large models from the Stanford 3D Scanning Repository called the statue and Lucy. These two models are treated in less than 4 minutes, and smooth fields are obtained (see Figure 9.5). Note that these two models have a complex topology (large genus g and number of borders b). The turning number around each border is used to counter the effect of each handle (a small handle is equivalent to a singular point of index 2).

	Lucy	The statue
<i>genus</i>	45	13
<i>#borders</i>	47	9
<i>#triangles</i>	125000	300000
<i>time</i>	2min 47s	3min 36s

TAB. 9.1 – Timings obtained on a Pentium IV 1.7Ghz

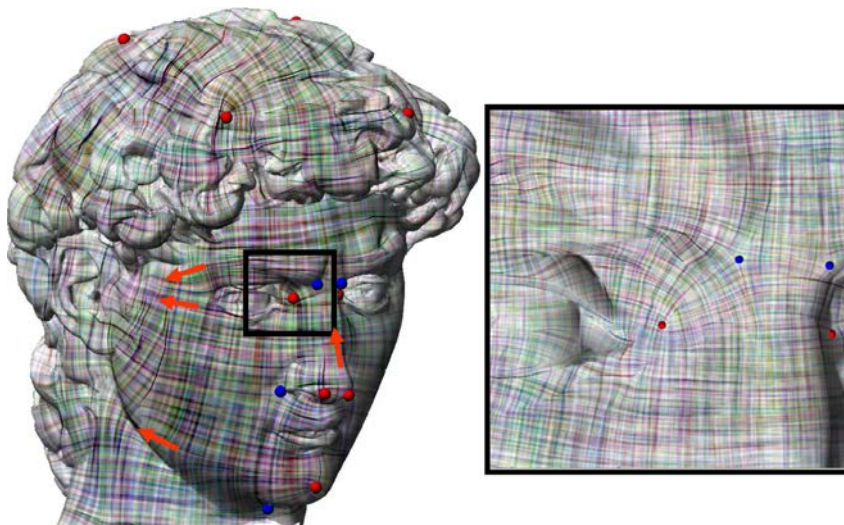


FIG. 9.4 – Direction field edited by topological and geometrical constraints. Red (resp. blue) dots are constrained singularities with positive (resp. negative) index. Orange arrows are directional constraints.



FIG. 9.5 – Large models with many borders and high genus efficiently processed by our framework.

Our algorithm also deals nicely with important constraints that can be applied to geometry (see the rotation constraints in Figure 9.6) and on topology (see Figure 7.7).

Finally, we compare with recent work in direction field design. Figure 9.7 shows that starting from a user defined set of constraints (singularities and directional constraints), our method generates a direction field with no other singularity (right). To our knowledge, this is the first algorithm that achieves direction field design with provably correct singularity control. Figure 9.8 shows a more difficult configuration, successfully handled by our algorithm.

9.3.2 Conclusion

We have introduced a generalization of direction field that is general enough to manipulate direction fields with different kind of symmetries, whose applications range from texture synthesis to quad remeshing.

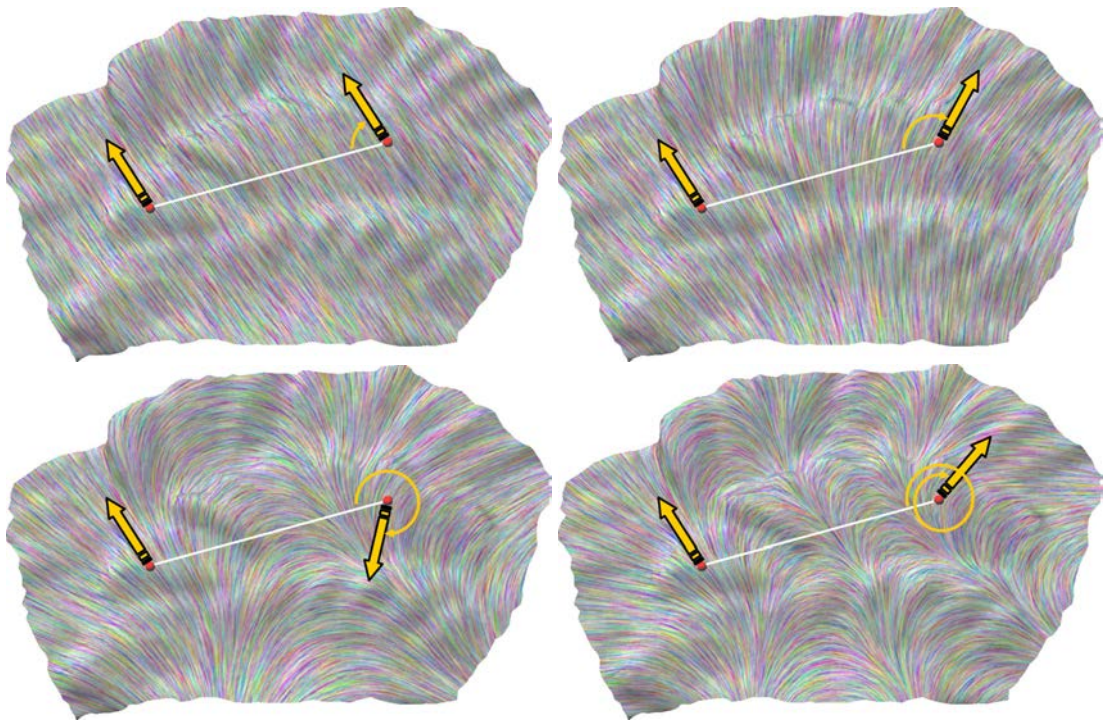


FIG. 9.6 – Direction constraints are applied to a direction field. Notice that the direction is given by a rotation of the base field that can be greater than 2π .

Our representation separates the topology and the geometry of the direction fields in order to simplify their manipulation : the singularities can be constrained through the topology by using only greedy algorithms, and the geometry can be smoothed by a simple quadratic form. This has been made possible by a complete analysis of the direction field topology including the characterization of the field behavior along cycles, and an extension of the Poincaré-Hopf theorem to rational indexes. A new direction field smoothing algorithm has been presented to demonstrate the benefits of our structure. With our algorithm, one can for the first time provably control the placement of singularities. We are confident this new way to consider and manipulate direction fields will set a clean and practical basis for a broad range of applications in geometry processing.

In this approach, all the topology must be specified (manually) by the user. Most of the time however, we want the topology to reflect the shape of the surface at a certain scale, which might require many singularities depending on the complexity of the shape and the desired level of resolution. In that case, it becomes tedious to place all singularities manually. In that case, a singularity placement heuristic such as the one of [Ben-Chen et al. 2008] might be more appropriate. Classic smoothing algorithms have the opposite problem : singularities emerge spontaneously, but too many of them appear if the surface presents small details or noise. Most of the time, no control is provided, except in [] where some local pair cancellations can be applied. This is extremely problematic if we want to produce a quad parameterization with few singular vertices. Therefore, we have worked on providing a weaker control over singularities where they emerge only to capture the global shape of the object at a given resolution, and not at the finest scales. This will be presented in the next Chapter.

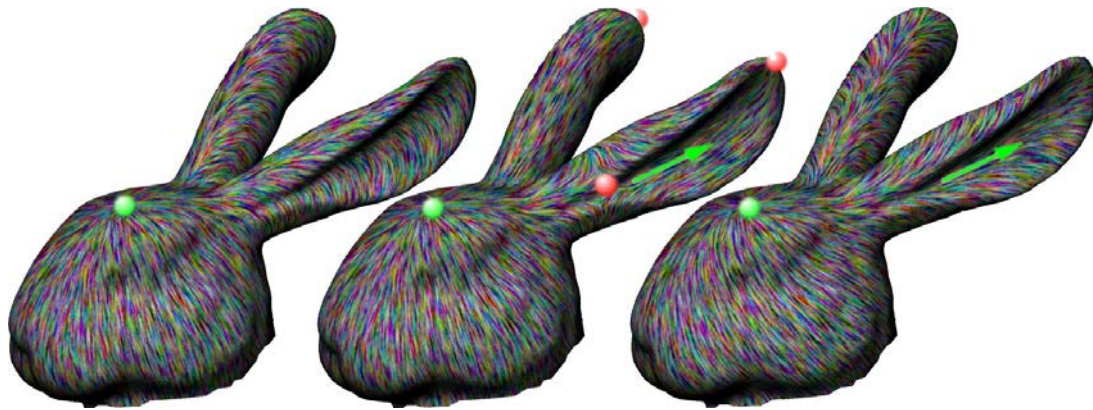


FIG. 9.7 – **Left** : result of [Fischer et.al 07] with a single singularity constraint (green dot); **Center** : result of [Fischer et.al 07] with the same singularity constraint and directional constraint (green arrow), additional singularities appear (red dots); **Right** : our result with the same set of constraints. As predicted by our theorem, no additional singularity appears.

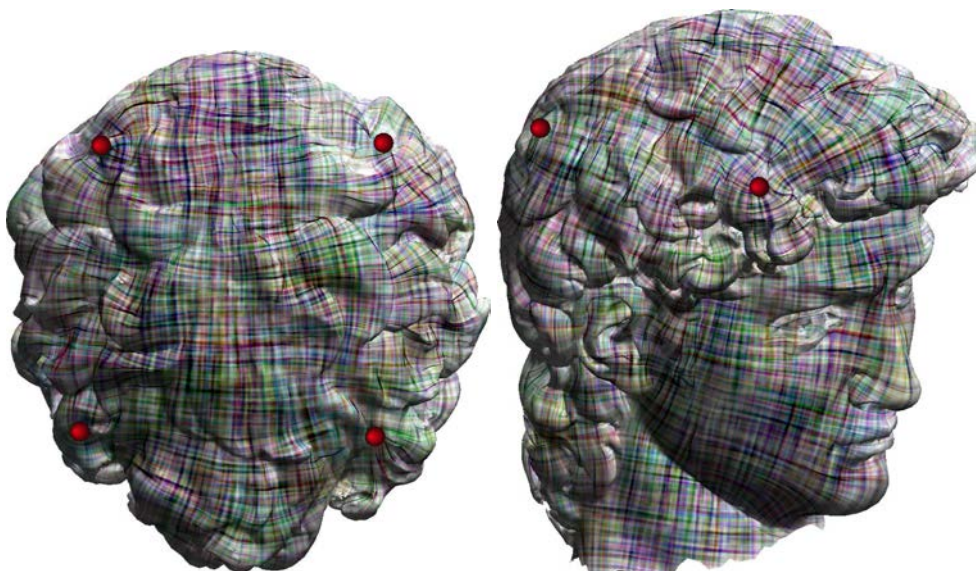


FIG. 9.8 – A particularly challenging data set for direction field design algorithms. Our algorithm successfully verifies the four user-defined singularities (green dots) and does not generate any additional singularity. From a topological point of view, this is equivalent to a cube (minus its bottom face).

Manipulation de champs de directions avec contrôle de l'influence géométrique

Sommaire

10.1 Problem statement	159
10.2 Geometric smoothing algorithm	160
10.3 Canceling the influence of the geometry	160
10.3.1 Topological disks	161
10.3.2 Extension to surfaces of arbitrary genus	162
10.4 Filtering the geometric influence	162
10.4.1 Filtering K	163
10.4.2 Computing singularity indexes	163
10.4.3 Editing the direction field topology	163
10.5 Results and Applications	164
10.6 Conclusion	166

Introduction

Comme on l'a expliqué dans l'introduction de cette partie, les champs de directions sont nécessaires à de nombreuses applications en informatique graphique pour orienter des objets sur les surfaces triangulées. Comme nous l'avons vu dans le Chapitre 6, de tels champs sont généralement construits en extrapolant des contraintes utilisateur en minimisant une énergie composé d'un terme de régularité et d'un terme d'attache aux données. Plus récemment, des structures spécialisées (les N -Rosy présentés au Chapitre 6 et les champs de directions à N symétries que nous avons introduits au Chapitre 7) ont été inventées afin d'unifier la manipulation de ces champs, et de fournir un contrôle topologique (singularités) et géométrique sur ces champs. Le contrôle topologique total présenté au chapitre précédent permet de ne générer que très peu de singularités, même en présence de hautes fréquences (détails) dans la géométrie de surface. Cependant, l'utilisateur doit explicitement spécifier toutes les singularités au lieu de les laisser émerger naturellement du lissage et de l'extrapolation. Sur des modèles compliqués, il peut être nécessaire de placer de nombreuses singularités ce qui peut être une tâche fastidieuse. En outre, la simplicité topologique peut nécessiter de trop fortes distortions du champs, comme illustré dans la Figure 10.1. Cet

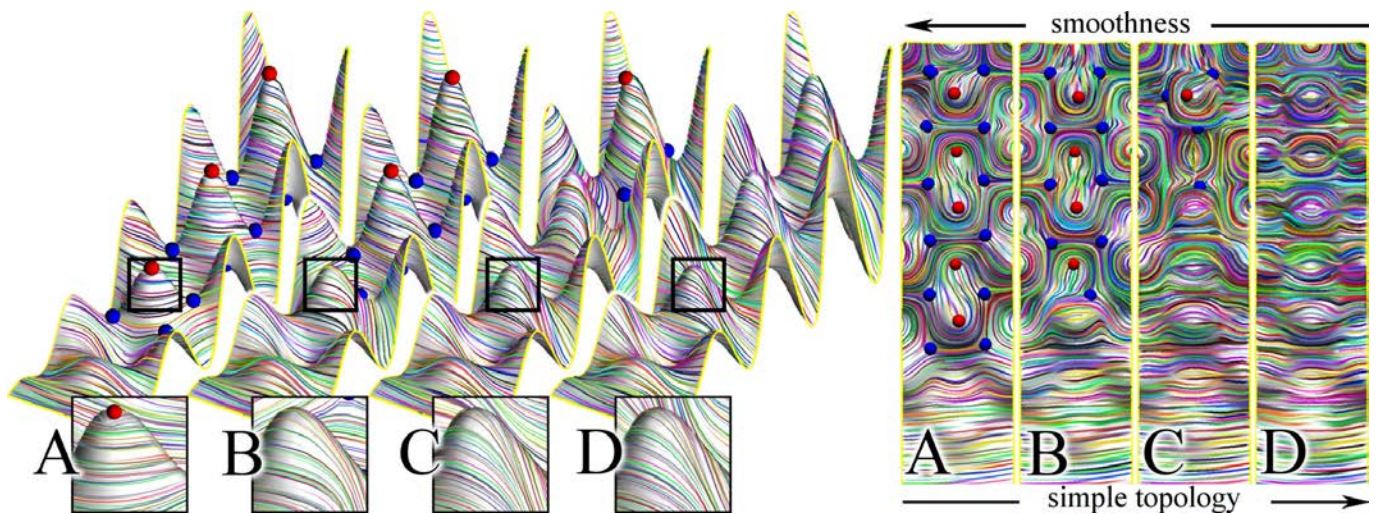


FIG. 10.1 – Notre nouvelle méthode de manipulation de champs de directions propose un compromis entre deux objectifs : la régularité et la réduction du nombre de singularités (de gauche à droite). Les points rouges et bleus signalent la position de singularités d'indices positifs et négatifs.

antagonisme entre la régularité et la simplicité topologique peut sembler contre intuitive à première vue. Toutefois, nous allons l'expliquer sur la base des résultats du Chapitre 7.

Ce chapitre présente une représentation intermédiaire qui garde des opérations intuitive telles que le lissage et l'extrapolation de contraintes directionnelles, mais modifie l'objectif de régularité d'une manière qui évite l'apparition de singularités dessous d'un niveau de détails donné. L'utilisation de cet outil de manipulation de champs de directions est intuitive, simple et permet de créer des champs avec peu de singularités, même en présence de hautes fréquences géométriques, sans devoir spécifier ces singularités manuellement.

Plan du chapitre

Ce chapitre est organisé de la façon suivante : nous commencerons par poser notre problème dans la Section 10.1. Ensuite, la Section 10.2 réexpliquera la méthode GDS dans le formalisme introduit au Chapitre 7, et présentera GDS comme la limite "géométrique" de notre algorithme (cf Figure 10.1-A). La Section 10.3 présentera la limite "topologique" de notre algorithme où le nombre de singularités est minimisé en supprimant complètement l'influence de la géométrie sur l'apparition de singularités (cf Figure 10.1-D). Enfin, la Section 10.4 présente un compromis entre ces deux limites où seule l'influence des hautes fréquences (détails fins) de la géométrie est annulée (Figure 10.1-BC).

Contributions

Notre contribution principale dans ce chapitre est de proposer un outil de manipulation de champs de directions qui présente toutes les caractéristiques souhaitables exhibées au Chapitre 6 : symétrie de rotation, régularité au sens des directions, contrôle géométrique et topologique intuitif. Il propose un compromis entre les approches purement *géométriques* qui ne cherchent qu'à maximiser la régularité du champs

sans en contrôler la topologie, et l’algorithme *topologique* présenté au chapitre précédent qui lisse un champ de directions à topologie fixée explicitement par avance.

Les algorithmes *géométriques* génèrent souvent trop de singularités dès que la géométrie devient complexe ou présente des hautes fréquences (détails fins). L’approche *topologique* ne souffre pas de ce problème mais nécessite de spécifier manuellement (ou par des heuristiques) toutes les singularités. En revanche, notre algorithme réduit de façon importante les singularités dues aux hautes fréquences géométriques, tout en laissant le lissage faire émerger les singularités de façon globale.

La limite *géométrique* de notre algorithme est équivalente à la méthode GDS (Chapitre 6 Section 6.3.1). La limite *topologique* **n’est pas** équivalente à la méthode du chapitre précédent qui fixe la topologie *a priori*. Cependant, elle assure une topologie simple indépendante de la géométrie de la surface. En particulier, sur un disque topologique et en l’absence de contraintes directionnelles, l’algorithme contrôle de façon prouvable la position et le type des singularités.

Ce chapitre fait l’objet de l’article [Ray et al. 2008a] qui est actuellement en cours de soumission. Comme les autres chapitres de cette thèse tirés directement des publications de l’auteur, le contenu technique de ce chapitre est laissé en anglais. Ces autres chapitres ont été retravaillés depuis leur publication pour prendre en compte le formalisme du calcul extérieur que nous avons adopté dans [Ray et al. 2008a] et pour simplifier et harmoniser les notations. La soumission de [Ray et al. 2008a] étant plus récente, elle utilise déjà ce formalisme, si bien que ce chapitre est très fidèle à la soumission.

10.1 Problem statement

Our objective is to design a direction field optimization algorithm that can be used to smoothe an existing direction field, supports directional constraints and automatically places singularities. We also want to control the influence of the geometry on the apparition of these singularities to avoid “noisy topology” such as in the hand model in Figure 6.3. In this chapter, we will refer to objectives (equalities) that we want to get close to, rather than energies that we want to minimize as the method used will not always be least squares minimization. Thus a natural smoothness criterion for a direction field depending only on the direction (not on a vector norm) will be the objective $\kappa_1(\theta_0, p_1) = 0$ (κ_1 is the direction field curvature defined in Equation (7.16)). We will explain in Section 10.2 how this is closely related to the GDS method (Chapitre 6, Section 6.3.1). Unfortunately, this approach generates too many singularities in the presence of small geometric details or noise. This Chapter focuses on limiting the influence of the geometry in the apparition of such singularities.

The control of the geometric influence is based on the relation (7.18) between index, direction field curvature and surface angle defect. If we generate a smooth field, it will have a small curvature $\kappa_1(\theta_0, p_1)$. Thus $d_1 \kappa(\theta_0, p_1)$ will be small too as it sums $\pm \kappa_1(\theta_0, p_1)$ over one rings. In this case, the index will be close to $K_2/2\pi$. As a consequence, smoothing algorithms introduce singularities to balance the angle defects. For instance, in Figure 7.11, introducing a singularity of index 1/4 on a cube corner of angle defect $\pi/2$ allows to cancel the curvature $\kappa_1(\theta_0, p_1)$. Conversely, removing singularities ($I(\theta_0, p_1) = 0$) leads to generate an amount of curvature that is proportional to the angle defect. We capture this curvature in a (minimal norm) target 1-form κ_1^t such that $d_1 \kappa_1^t = -K_2$. We can then modify our objective to $\kappa_1(\theta_0, p_1) = \kappa_1^t$. Indeed, if $\kappa_1(\theta_0, p_1)$ is close to κ_1^t , the index will tend to be zero even if the angle defect is high. Choosing κ_1^t to have minimal norm keeps the new objective as close as possible to the original smoothness criterion.

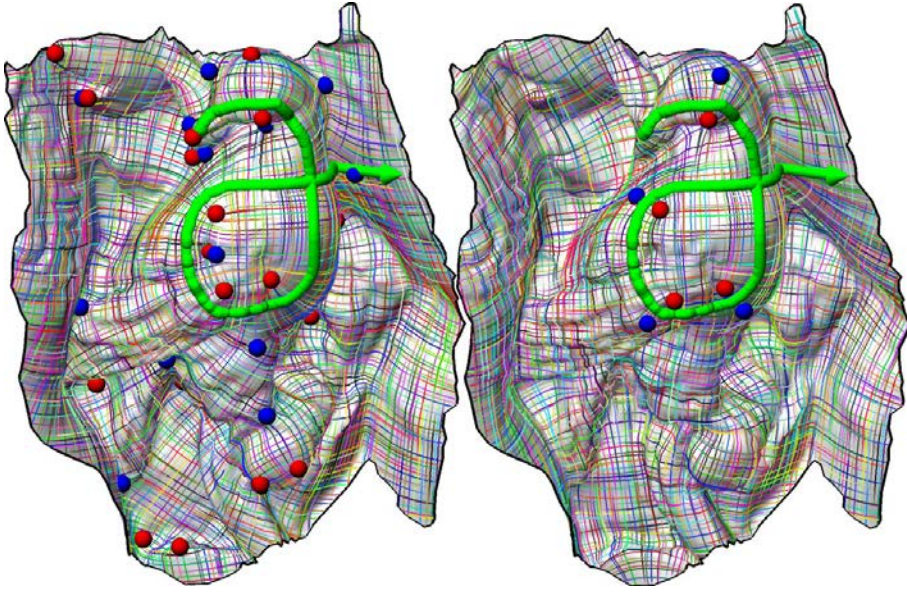


FIG. 10.2 – Directional constraints (green arrow) with filtered geometric influence (left) and without geometric influence (right).

10.2 Geometric smoothing algorithm

We start by recalling the GDS method of [Li et al. 2006a] that we have described in Chapitre 6. This method generates a smooth N -symmetry direction field given a set of soft and hard constraints. The objective of GDS is :

$$\mathbf{v}_i - R(N \cdot \kappa_1^r(ij)) \mathbf{v}_j = 0 \quad (10.1)$$

If we compare it with the zero curvature smoothness objective :

$$\kappa_1(\theta_0, p_1)(ij) = \kappa_1^r(ij) + \theta_0(j) - \theta_0(i) + 2\pi p_1(ij)/N = 0 \quad (10.2)$$

we see that they are related by the change of variables $\mathbf{v}_i = (\cos(N\theta_0(i)), \sin(N\theta_0(i)))$, and the integer variables p_1 are eliminated by taking the cos and sin of (10.2). This is coherent because we are interested in direction which are normalized ($\|\mathbf{v}_i\| = 1$). In GDS, this is enforced in the optimization.

At the end of the optimization, we can recover the angles $\theta_0(i) = \text{atan}(\mathbf{v}_i \cdot (0, 1) / \mathbf{v}_i \cdot (1, 0)) / N$ and we add π/N if $\mathbf{v}_i \cdot (1, 0) < 0$. Finally, p_1 is defined to minimize the field curvature as the closest integer to $N(-\kappa_1^r - d_0\theta_0)/2\pi$.

As explained earlier, this pure smoothness objective generates singularities on fine details. We will now explain how to cancel this geometric influence by modifying the objective (10.2).

10.3 Canceling the influence of the geometry

This section shows how to cancel the impact of the surface geometry in the generation of singularities. We will first prove a theorem stating that we can control exactly the singularities on a topological disk. In particular, we can choose to have no singularities, but this comes at the cost of smoothness. On arbitrary topologies, the Poincaré-Hopf theorem states that singularities need to appear. Instead of

[Ray et al. 2008b] where these necessary singularities are exactly controlled, we propose an algorithm to let a minimum number of them emerge naturally. The algorithm described here will be further improved in the Section 10.4 to prevent only the singularities due to high geometric frequencies from appearing.

10.3.1 Topological disks

On a topological disk, we can have exact control over the singularities by choosing a 2-form I_2^t that constrains the desired index :

Théorème 10.3.1 (Exact control) *Let \mathcal{D} be a mesh with disk topology, and I_2^t be a 2-form with values multiple of $1/N$ on \mathcal{D} . Then for any 1-form κ_1^t such that :*

$$d_1 \kappa_1^t = -K_2 + 2\pi I_2^t \quad (10.3)$$

there exists a discrete direction field (θ_0, p_1) such that $I(\theta_0, p_1) = I_2^t$ and $\kappa_1(\theta_0, p_1) = \kappa_1^t$.

PREUVE. The Discrete Poincaré Lemma [Desbrun et al. 2005b] implies that for a k -form ω_k on a topological disk, $d_k \omega_k = 0$ if and only if there exists a $(k-1)$ -form σ such that $\omega = d_{k-1} \sigma$. We will invoke this argument twice.

1. $I_0^r = I(0, 0) = (d_1 \kappa_1^r + K_2)/2\pi$ is the index of the singularities of the field defined by the reference vectors with curvature κ_1^r , such that $N(I_2^t - I_2^r)$ is an integer 2-form. $d_2 N(I_2^t - I_2^r) = 0$ by definition of d_2 so there exists an integer 1-form p_1 such that $N(I_2^t - I_2^r) = d_1 p_1$. Then by definition of the index (7.18), $I(\theta_0, p_1) = (d_1 \kappa_1^r + K_2)/2\pi + d_1 p_1/N = I_2^r + (I_2^t - I_2^r) = I_2^t$.
2. we have $d_1(\kappa_1^t - \kappa_1^r - 2\pi p_1/N) = -K_2 + 2\pi I_2^t - d_1 \kappa_1^r - 2\pi(I_2^t - I_2^r) = 0$ so there exists a 0-form θ such that $d\theta = \kappa_1^t - \kappa_1^r + 2\pi p_1/N$. Inserting these expressions into the definition of curvature (7.16) we have $\kappa_1(\theta_0, p_1) = \kappa_1^t$ □

The first part of the theorem ($I(\theta_0, p_1) = I_2^t$) shows that we can exactly control singularity indexes by setting an appropriate p_1 (independently from θ). In particular, we can remove all singularities by setting $I_2^t = 0$. The second part shows that we can control the curvature of the field as long as the target curvature κ_1^t satisfies $d_1 \kappa_1^t = -K_2 + 2\pi I_2^t$. Hence if we want to smoothe a field with topology control, we cannot ask for $\kappa_1^t = 0$ anymore, but only for the κ_1^t of minimal norm (defined in (3.10)) under this constraint. The discrete Poincaré lemma also implies that there exists at least one exact solution to $d_1 \kappa_1^t = -K_2 + 2\pi I_2^t$, and we select the one of minimal norm. These constraints are linear and can be exactly fulfilled, so we can enforce them with Lagrange multipliers. However, a simple and faster solution in practice is to enforce the constraints only approximately with a *quadratic barrier* : we add to the energy to minimize the sum of squared constraints. At the minimum, the constraints will be close to be enforced. In our context, we do not need exact enforcement of the constraints, and this leads to a smaller system, with similar results.

Once the optimal κ_1^t has been computed, the smoothing algorithm (Section 10.2) can be adapted by replacing the objective (6.1) with the new objective $\kappa_1(\theta_0, p_1) = \kappa_1^t$, which by the same transformation becomes :

$$\mathbf{v}_i = R(N(\kappa_1^r(ij) - \kappa_1^t(ij)))\mathbf{v}_j \quad (10.4)$$

If a unique hard constraint on \mathbf{v}_i is given, the system admits an exact solution that can be computed by an exact solver, and corresponds to the smoothest field with no singularity (see left model in Figure 10.3). In other terms, this corresponds to smoothing the surface with virtually zero Gaussian curvature.

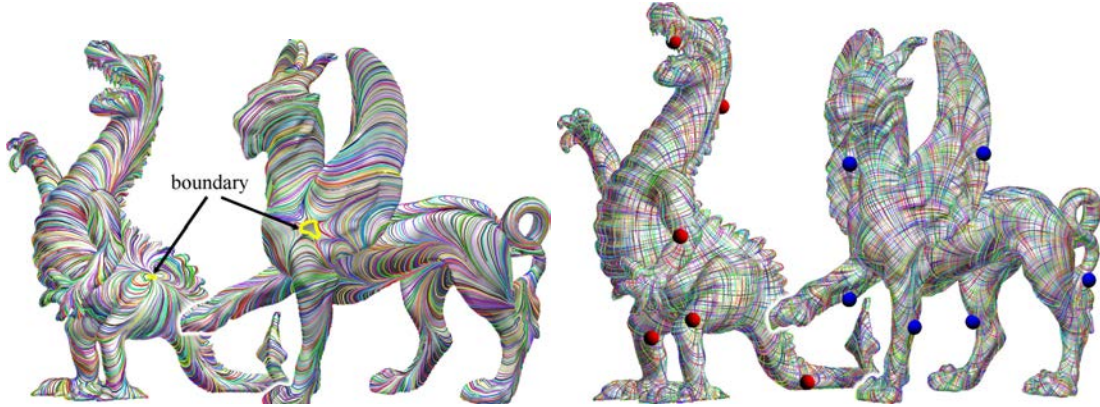


FIG. 10.3 – **Left** : removing geometric influence on a topological disk (left) and a genus 2 surface with a border (right). **Right** : if no geometric influence is wanted, the algorithm evenly distributes singularities on the surface : 8 singularities of index 1/4 on topological sphere (left), and 8 singularities of index -1/4 on genus two surface (right).

10.3.2 Extension to surfaces of arbitrary genus

For open surfaces of arbitrary genus, the same algorithm will not generate singularities in practice (see right model in Figure 10.3). For closed surfaces of arbitrary genus, we have $\sum_{v \in \mathcal{V}} d_1 \kappa_1^f(v) = 0$ and $\sum_v K_2(v) = 2\pi\chi$, where χ is the Euler characteristic of the surface. As a consequence, it is impossible to enforce $d_1 \kappa_1^f = -K_2$. The solution proposed in this section, and improved in Section 10.4 is to chose an κ_1^f satisfying $d_1 \kappa_1^f = \bar{K}_2 - K_2$ where \bar{K}_2 is defined such that \bar{K}_2/A is constant :

$$\bar{K}_2(v) = \frac{|v^*| \sum_{v'} K_2(v')}{\sum_{v'} |v'^*|} = \frac{2\pi\chi |v^*|}{\sum_{v'} |v'^*|}$$

and $|v^*|$ is the dual cell area of v . With this κ_1^f , (10.4) does not have an exact solution so we apply the iterative process described in Section 10.2 with objective curvature κ_1^f such that $d_1 \kappa_1^f = \bar{K}_2 - K_2$. This smoothing behaves as if the surface had a constant Gaussian curvature, so it evenly distributes singularities over the surface (see Figure 10.3). Obviously this is an extreme solution where the singularities appear independently from the geometry. Next section presents a trade-off where only the influence of high geometric frequencies is removed.

10.4 Filtering the geometric influence

The previous section has presented how to smoothe a direction field with singularities appearing independently from the geometry of the surface. This section focuses on removing only the effect of high frequencies of the geometry.

In Section 10.2, we have presented a geometric smoothing algorithm with objective $\kappa_1(\theta_0, p_1) = 0$. In Section 10.3, the influence of the surface's geometry on the smoothing is canceled by changing the objective to $\kappa_1(\theta_0, p_1) = \kappa_1^f$ such that $d_1 \kappa_1^f = \bar{K}_2 - K_2$ as in this case, the smoothing behaves as if the surface has a constant Gaussian curvature $K_2^* = \bar{K}_2/A$. More generally, changing the objective to $d_1 \kappa_1^f = K_2^{corr} - K_2$ will make the smoothing behave as if the surface had angle defects K_2^{corr} instead of K_2 . What most applications need is to remove only the singularities due to high geometric frequencies while

placing singularities according to the global shape of the surface. To achieve this, we chose as K^{corr} a low-pass filtered version of K_2 . As it must still be an admissible angle defect, K_2^{corr} needs to satisfy $\sum_v K_2^{corr}(v) = 2\pi\chi$.

10.4.1 Filtering K

In our context, it is important to smoothe only high frequencies and to preserve the global distribution of Gaussian curvature $K_2^* = K_2/A$. As singularities appear as ‘‘clusters’’ of K_2 , the repartition of K_2 on the surface plays an important role. We will smooth K_2^* with a Gaussian smoothing algorithm based on an raw estimation of the geodesic distance by the shortest path of edges between two points. Note that the algorithm acts on K_2 directly but it smooths K_2^* . We start by setting $K_2^{corr} = 0$ and we choose a Gaussian radius σ . Then for each vertex i , we use a Dijkstra algorithm to compute the distance D_{ij} from vertex i to vertex j , stopping when $D_{ij} > 2\sigma$. The filter can then be computed as :

$$K_2^{corr}(j) = \sum_i \frac{c_{ij}K_2(i)}{\sum_k c_{ik}} \quad c_{ij} = |v_j^*| e^{-\left(\frac{D_{ij}}{\sigma}\right)^2}$$

where A_j is the dual cell area of v_j . Notice that the smoothing ensures :

$$\sum_v K_2(v) = \sum_v K_2^{corr}(v)$$

as $\sum_j c_{ij}K_2(i)/\sum_k c_{ik} = K_2(i)$. Large values of σ will correspond to a more smooth K_2^{corr} thus to remove larger geometric details, but will require longer computation time. The geometric smoothing algorithm of Section 10.2 corresponds to the limit $\sigma \rightarrow 0$, where $c_{ij} \rightarrow 0$ except $c_{ii} \rightarrow A_i$ and $K_2^{corr} \rightarrow K_2$. The geometry canceling smoothing algorithm of Section 10.3 corresponds to the limit $\sigma \rightarrow \infty$, where $c_{ij} \rightarrow A_j$ and $K_2^{corr} \rightarrow \bar{K}_2$. Once K_2^{corr} is computed, we get κ_1^t as in Section 10.3 except that the constraint is replaced by $d_1 \kappa_1^t = K_2^{corr} - K_2$.

10.4.2 Computing singularity indexes

Depending on the application, the user might need to have access to singularity indexes. For instance, in global parameterization, they determine the type of cone singularities. The index definition requires p_1 to be known, while in our smoothing algorithm it is never explicitly defined. As our objective was $\kappa_1(\theta_0, p_1) = \kappa_1^t$, we can define a coherent p_1 as the integer minimizer of $(\kappa_1(\theta_0, p_1) - \kappa_1^t)^2$, which by (7.16) is the closest integer to $N(\kappa_1^t - \kappa_1^r - d_0\theta)/2\pi$. This expression differs from the usual way to compute the index because we take into account the geometric correction term κ_1^t .

10.4.3 Editing the direction field topology

When a user designs a direction field, the field topology emerges from the direction he constrains and the global shape of the surface. However, the user might want to control the position of the singularities accordingly to application dependent criteria. Simple examples can be to merge singularities (see Figure 10.5), or to move them to semantically meaningful positions, for instance to respect local symmetries (see Figure 10.4).

Our method is based on updating ω^t accordingly to the desired topology. Canceling a pair of singularities of indexes $+I$ at v and $-I$ at v' , or equivalently moving and index $+I$ singularity from v to v' is done

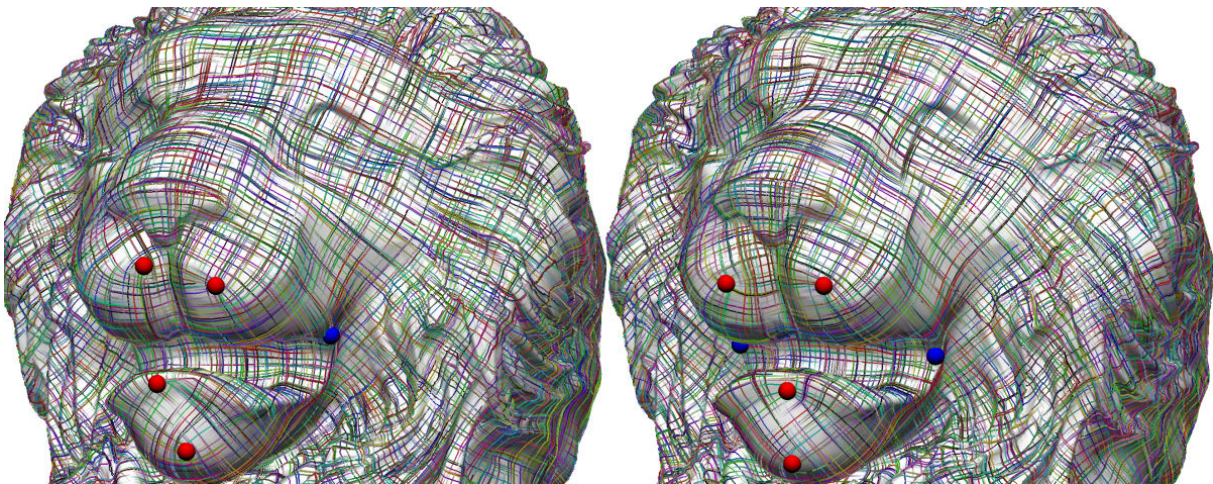


FIG. 10.4 – Topological editing is useful to adjust the topology, especially when the surface exhibits local symmetries.

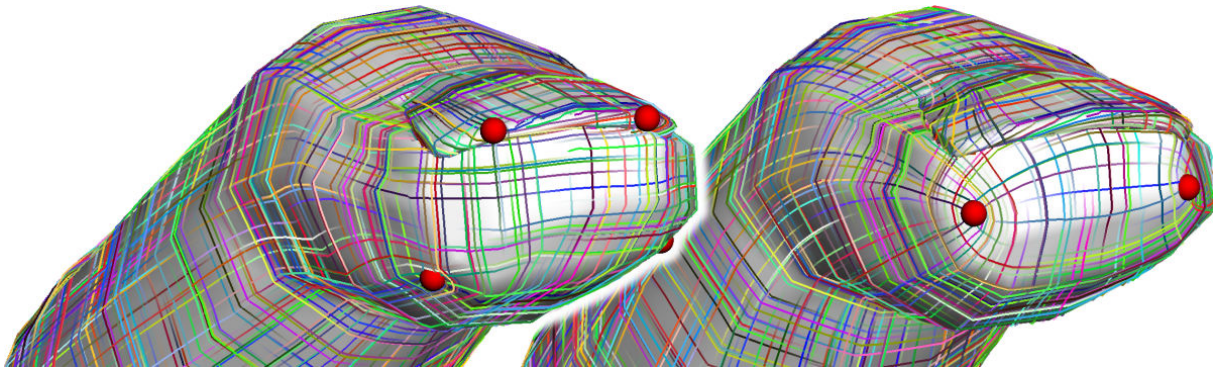


FIG. 10.5 – Merging singularities can simplify the field topology.

by adding I to $K^{corr}(v')$ and subtracting I to $K^{corr}(v)$. Then ω^t is computed as usual, and the field is smoothed again. This solution allows the user to continue adding directional constraints as before i.e., it can change the topology. In extreme cases, when the wanted singularity is close to noisy directional constraints, increasing weight of the smoothing criterion in the one-ring of singularities will ensure that the topology will be respected. Indeed, it can be considered as a constraint system solved with quadratic barrier.

10.5 Results and Applications

We recapitulate the three main steps of the final algorithm, provide some insight on the parameter tuning and timings for the Michelangelo's David statue at resolution 100K triangles :

1. Smoothing the angle defect K_2 is done by a Gaussian filter and the result is called K_2^{corr} (Section 10.4.1). The only tuning parameter is the Gaussian radius σ . Increasing σ makes it possible to “trade” some field smoothness against a simpler field topology. It takes respectively 20s, 1min 15s and 5 min 30s to smooth with $\sigma = 0.05h$, $0.1h$, and $0.2h$, where h denotes the height of the statue.

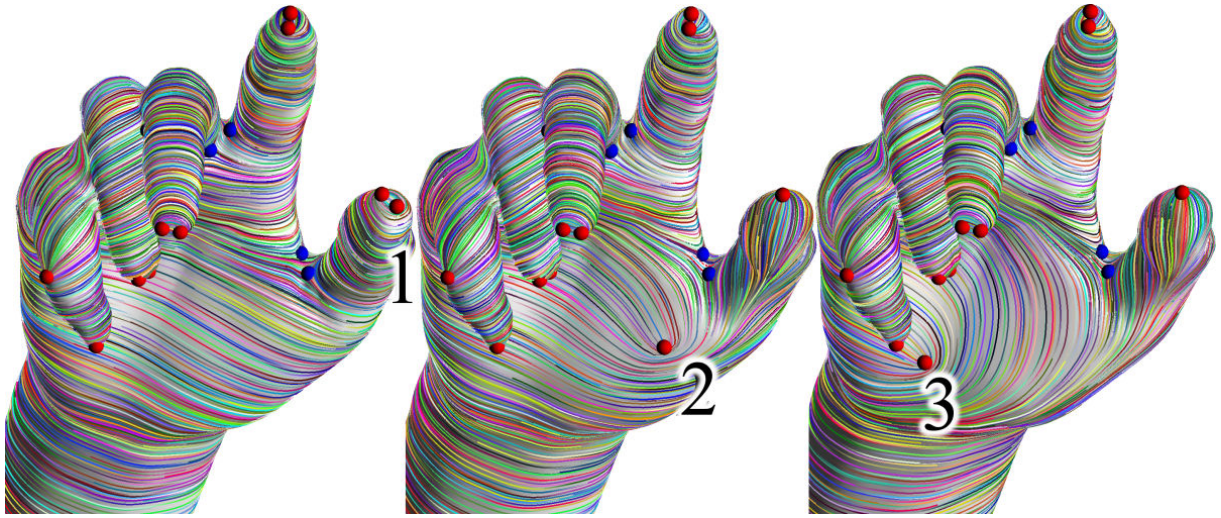


FIG. 10.6 – Moving an index from (1) to new positions (2) then (3), by a modification of ω^t is robust enough in practice.

2. Compute the target curvature κ_1^t : minimize its squared norm under constraint $d_1 \kappa_1^t = K_2^{corr} - K_2$ (Section 10.3). Computing κ_1^t from K_2^{corr} requires 2 seconds using CHOLMOD [Davis and Hager 2005] as linear solver.
3. Achieve $\kappa_1(\theta_0, p_1) = \kappa_1^t$ in the least square sense after a change of variables as in [Li et al. 2006a] (Section 10.2). It requires a series of quadratic minimizations. The tuning parameters are the trade-off λ between smoothness and data-fitting and the number of iterations. 1 to 5 iterations are usually sufficient, and each step costs 2.5 seconds with CHOLMOD on the David.

In comparison to our method :

- N-RoSy [Palacios and Zhang 2007] uses topology simplification as a post-processing step. It does not allow to preserve a simple topology while editing the direction field.
- NSDF [Ray et al. 2008b] requires manually placing singularities. It may be tedious for high genus models such as the Michelangelo’s David statue (genus 8) that requires at least 56 singularities of index $-1/4$ for a 4-symmetry direction field (see Figure 10.9).

For interactive design of direction fields, our algorithm allows to control the field by simply painting hard constraints on the surface while preserving the trade-off between simple topology and smoothness. Since new constraints are usually introduced to locally modify the field geometry, the smoothing is also performed locally. This allows real-time feedback while painting new constraints.

It has been shown in [Palacios and Zhang 2007] that complex field topology may affect the quality in texturing / hatching applications. For global parameterization based on cone singularities, that topological complexity becomes critical as it determines the complexity of the parametric domain. Our method provides direction fields suitable for this application as they only have the minimal number of singularities to capture the global shape of the object. As illustrated in Figure 10.8, the global parameterization (very coarse quad mesh) is simple enough to be used as parametric domain for geometry images [Gu et al. 2002] (increasing the geometry image resolution gives finer meshes).

Finally, the generality of the N -symmetry framework makes it eligible to triangular re-meshing based on 6-symmetry directions fields. As explained in Section 10.1, increasing the number of symmetries will also increase the number of singularities generated by high geometric frequencies. Figure 10.7 shows that

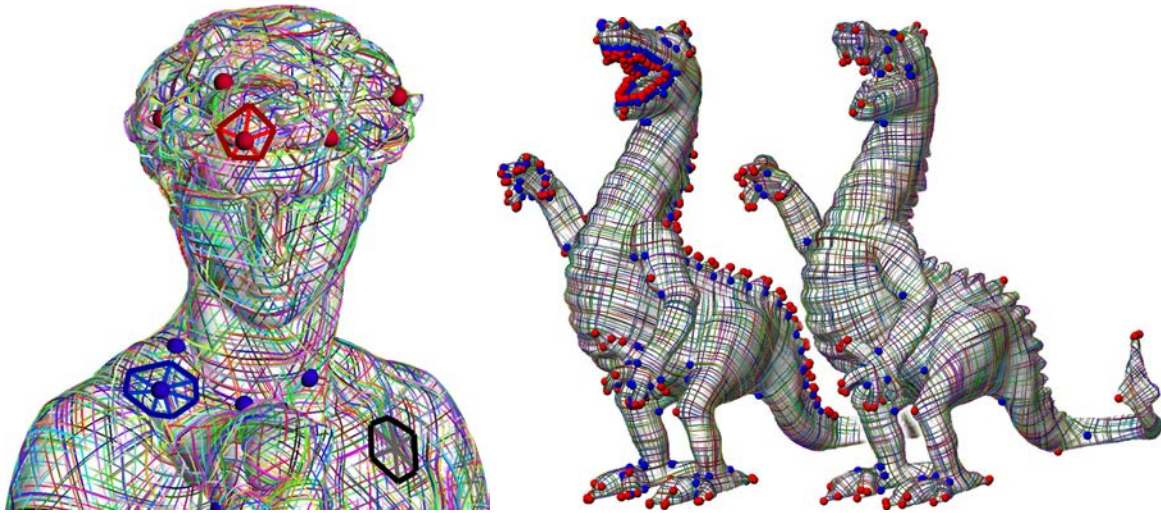


FIG. 10.7 – **Left** : A 6-symmetry direction fields generated on the head of the Michelangelo's David statue. Notice that for triangular re-meshing applications, $1/6$ indexes will become valence 5 vertices (red), $-1/6$ indexes will become valence 7 vertices (blue), and null indexes become regular (valence 6) vertices (black). **Right** : Our method (right) allows to remove singularities due to geometric high frequencies (left).

our framework allows to create 6-symmetry directions fields with few singularities even in the presence of high geometric frequencies.

10.6 Conclusion

In the past few years, the research topics of quad re-meshing and mesh parameterization have been converging toward what can be called quad parameterization. We believe that computer graphics and modeling applications would greatly benefit from quad parameterization as it allows for both seamless texture mapping and automatic conversions between surface representations. However, quad parameterization is still a theoretic concept as existing methods either fail on complex models or require too much user interaction. This work breaks through some limitations by providing a simple yet efficient tool to define quad orientation with a clean global topology.

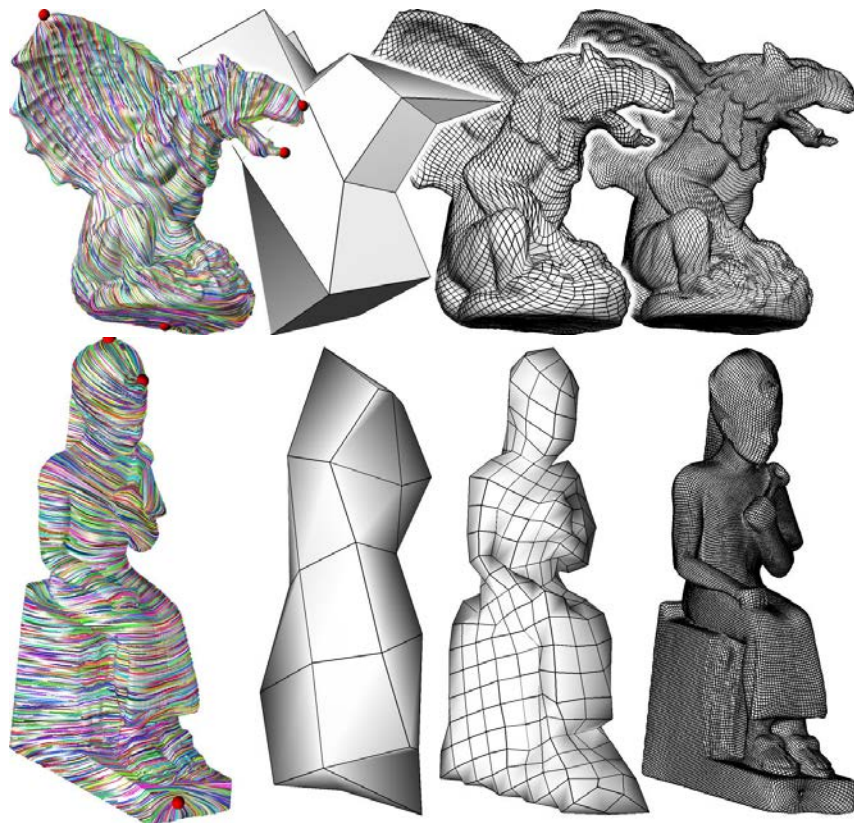


FIG. 10.8 – Producing few singularities allows to create very coarse quad mesh (right) with quad cover. This mesh can be used as base domain for geometric images with different resolution (middle).

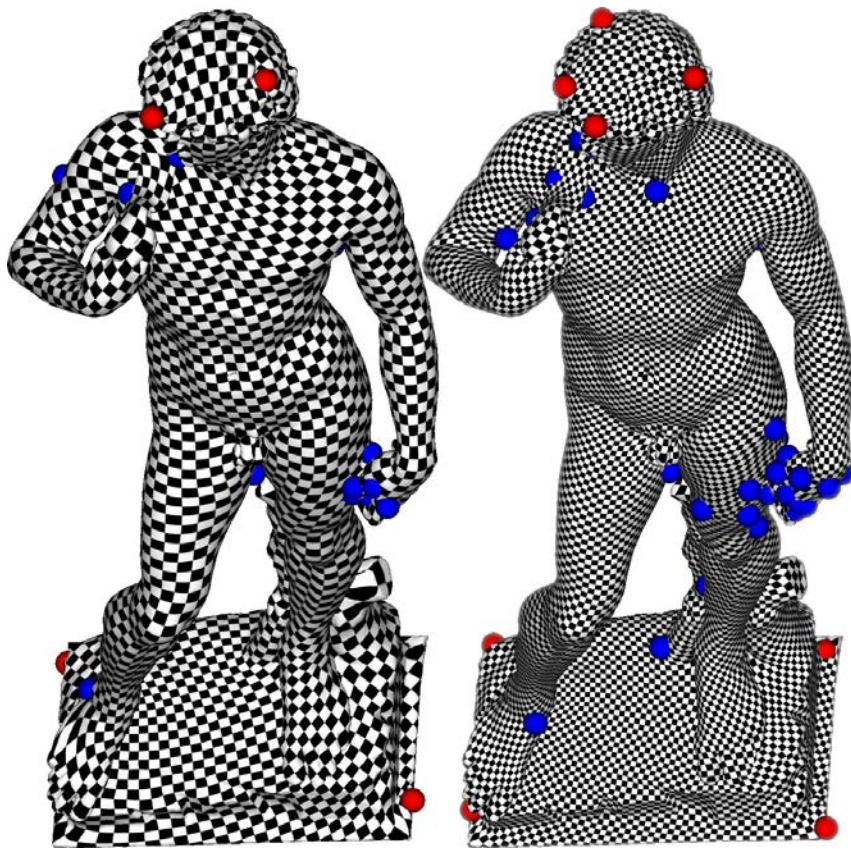


FIG. 10.9 – 2-symmetry (left) and 4-symmetry (right) direction fields generated by our method are used as input for a global parameterization (using quadcover).

Conclusion générale

Cette thèse a présenté des outils théoriques et pratiques pour étudier, comprendre et construire des bases de fonctions sur les surfaces. Ces bases de fonctions permettent d'approximer l'espace (de dimension infinie) $L^2(\mathcal{S})$ de toutes les fonctions de carré intégrable définies sur la surface \mathcal{S} . Parce que nous ne pouvons construire qu'une base finie, nous n'avons qu'un nombre limité de degrés de liberté avec lesquels nous devons essayer de paramétrer $L^2(\mathcal{S})$ aussi bien que possible. En outre, il est souhaitable que cette base ait de "bonnes" propriétés telles que l'orthonormalité. Nous avons vu que les bases de fonctions les plus largement utilisés en synthèse d'images sont les bases de fonctions interpolantes telles que les bases P_n ou bilinéaire. Ces bases de fonctions permettent de construire une approximation V_n de $L^2(\mathcal{S})$ de dimension finie n , mais elles ne sont ni normales ni orthogonales.

Nous avons présenté une méthode élégante pour construire une base orthogonale de V_n que nous avons appelée la "base d'Harmoniques Variétés" (cf Chapitre 5). En outre, cette base s'est révélée être localisée en fréquence, c'est à dire que chaque "Harmonique Variété" correspond à une fréquence sur la surface. Cela nous a permis de construire une méthode de filtrage spectral en temps réel. Grâce à l'orthonormalité des Harmoniques Variétés, le filtrage spectral s'exprime comme un simple produit matrice/vecteur, ce qui rend possible cette application en temps réel. Sans l'orthonormalité, nous aurions besoin d'inverser une grande matrice pour obtenir le même résultat. C'est l'une des remarques les plus importantes de cette thèse : la construction d'une bonne base de fonction rend le traitement plus facile. Toutefois, les Harmoniques Variétés perdent une propriété importante des fonctions d'interpolation : leur localité. Par analogie avec le traitement de l'image, nous pouvons dire qu'utiliser une base de fonctions d'interpolation correspond à la définir la valeur de chaque pixel, alors qu'utiliser les Harmoniques Variétés correspond à faire une transformée de Fourier sur l'ensemble de l'image. Toutefois, des méthodes efficaces en traitement d'image sont basées sur les fonctions définies à une échelle intermédiaire entre le pixel et l'image entière. En particulier, la transformée en cosinus discrets utilisée dans la compression JPEG ne calcule pas une transformée de Fourier sur toute l'image, mais seulement sur des carrés de pixels (en général 8 par 8) pavant l'image. Dans le même ordre d'idée, la compression par ondelettes est basée sur une base de fonctions à une échelle intermédiaire. Contrairement aux Harmoniques Variétés, ces fonctions sont définies sur un pavage régulier du domaine de base en carrés. Ainsi, pour étendre ces concepts aux surfaces, nous devons construire de tels domaines carrés. C'est pourquoi nous nous sommes concentrés dans la dernière partie de cette thèse sur les champs de directions à symétries qui donnent une référence locale sur la surface permettant d'orienter ces carrés. De toute évidence, la principale différence avec le cas du traitement d'image est que notre grille carrée présentera des singularités. C'est pourquoi la dernière partie de cette thèse se concentre en particulier sur le contrôle (fort ou faible) de ces singularités.

En utilisant les résultats de notre dernière partie, et les progrès récents en géométrie numérique [Kalberer et al. 2007], il est possible de générer une paramétrisation quad (découpage en carrés) d'une surface triangulée avec très peu de singularités. Ainsi nous avons proposé des méthodes de manipulation de champs de directions avec un contrôle fort (Chapitre 9) ou faible (Chapitre 10) sur ces singularités. Les paramétrisations

quad en résultant forment un espace idéal pour installer des bases de fonctions comme celle de la transformée en cosinus discrets ou des ondelettes sur la surface. C'est une idée très prometteuse qui nous paraît mériter d'être étudiée. Ces bases de fonctions pourraient avoir d'importantes applications dans tous les domaines de la géométrie numérique qui nécessitent de définir des fonctions sur les surfaces. Par exemple, elles pourraient permettre la définition d'un Laplacien avec toutes les propriétés de son homologue continue, surmontant ainsi les limitations mentionnées dans [Wardetzky et al. 2007b]. Ainsi, je pense que le travail présenté sur les champs de directions à symétries est une importante contribution de cette thèse. Contrairement aux approches habituelles basées sur les champs vecteur pour définir les directions, nous avons enquêté sur ce qui se passe lorsque nous travaillons sur les directions directement, puisque ce sont elles qui nous intéressent. En d'autres termes, au lieu de représenter des vecteurs par leurs coordonnées dans une base locale, nous avons choisi de travailler directement avec des directions, ou des vecteurs normalisés. Ce choix est très géométrique : la notion fondamentale de courbure est définie par les dérivées de telles directions (la tangente et la normale sont des vecteurs unitaires). En outre, les éléments les plus caractéristiques (géométriquement) d'un champ de vecteurs sont leurs lignes de champs, qui dépendent seulement de leur direction. Cette approche nous a été utile : elle a permis à la fois d'obtenir des résultats théoriques importants, et des progrès techniques dans les applications. Du point de vue théorique, notre étude sur les champs de directions à symétries sur les surfaces permet une bien meilleure compréhension de ces objets. En particulier, nos deux théorèmes sur la topologie des champs de vecteurs exposent très clairement les relations entre la topologie d'un champs de directions et celle de la surface sur laquelle celui-ci est défini, même lorsque cette surface a un bord. La très élégante formule sur les indices (7.18) révèle des relations très simples entre les trois notions fondamentale de courbure d'un champs, courbure Gaussienne d'une surface et indices de singularités. Sa grande simplicité en fait le principal fondement du Chapitre 9 (l'algorithme de "Zipping" utilise cette formule pour définir les sauts de période qui contrôlent la topologie) et du Chapitre 10 (imposer la nullité de l'index défini par (7.18) donne une condition que la courbure du champ doit vérifier pour éviter singularités).

Si cette thèse a donné des solutions à des problèmes importants, elle a également soulevé de nouveaux problèmes et de nouvelles questions. L'un des problèmes les plus subtils que nous avons rencontrés dans cette thèse est lié à la déformation de surfaces triangulées basé sur des opérateurs (par exemple le lissage). En fait, ces opérateurs dépendent de la métrique de la surface, qui est altérée au cours de la transformation. Ainsi, l'opérateur évolue pendant la transformation de la surface, ce qui n'est jamais pris en compte. Ce problème a des conséquences indésirables en pratique. Par exemple, le lissage de la géométrie d'un maillage peut générer des pointes (cf Figure 5.11), ce qui est très contre-intuitif puisque le lissage est censé éliminer ces pointes. Comprendre ce qui est en jeu ici, et la manière de tenir compte du fait que l'opérateur de lissage varie avec la géométrie est un problème important qui mérite d'être étudié. En particulier, nous devrions essayer de comprendre ce qui arrive à une base de fonctions lorsque la surface sur laquelle elle est définie est soumise à des déformations.

En conclusion, les bases de fonctions sont des outils fondamentaux pour définir n'importe quelle quantité sur une surface, ce qui est au coeur de nombreux algorithmes en informatique graphique, et en géométrie numérique en particulier. Comme on peut le voir dans cette thèse, les bases de fonctions sur les surfaces triangulées souffrent de diverses limitations : les fonctions de base interpolantes ne sont pas orthogonale et trop localisées. Les Harmoniques Variétés ne sont pas localisées et sont assez couteuses à calculer si l'on veut en avoir assez pour capturer les détails fins. La définition de bases de fonctions sur une paramétrisation quad sera donc sûrement la voie à suivre dans l'avenir. Cela exige de comprendre les propriétés fondamentales d'une paramétrisation quad, et en particulier des champs de directions nécessaires à leur construction. Dans ce contexte, une importante contribution de cette thèse est l'étude approfondie de ces champs de directions, qui sont nécessaires pour contrôler et construire des paramétrisations quad robustes. Dans cette thèse, je défend l'idée que les paramétrisations quad sont nécessaires pour définir

une base de fonction sur les surfaces qui ne souffrirait pas des limitations actuelles. Les paramétrisation quad ont aussi d'autres applications importantes. Elles permettent de plaquer des textures sur des surfaces sans discontinuités visibles, et plus généralement d'étendre les outils du traitement d'image (qui s'appuie sur des domaines rectangulaires) aux surfaces. Le plaquage de texture sans discontinuité est un enjeu important parce que les découpes sont fastidieuses à définir et produisent des artefacts visuels indésirables. L'extension des outils du traitement d'image à des surfaces est également désirable car il permettrait encore plus d'interaction entre les deux communautés très riches du traitement d'image et de la géométrie numérique. Ainsi, je pense que l'un des principaux objectifs de la communauté de la géométrie numérique est de concevoir des méthodes de paramétrisations quad qui seront suffisamment simples et robustes pour être utilisées en dehors du monde universitaire et à une grande échelle. Les travaux récents dans le domaine de la paramétrisation et du remaillage convergent définitivement vers cet objectif. Les travaux effectués sur la paramétrisation au cours des dernières années ont donné lieu à des méthodes maintenant largement diffusées en dehors du cercle académique (LSCM et ABF sont désormais implémentés dans la plupart des modélisateurs). Les paramétrisation quad étant une suite logique de ces travaux, je suis convaincu qu'elles rencontreront le même succès. Comprendre et contrôler les champs de directions est un élément fondamental et nécessaire pour atteindre un tel objectif. J'espère donc que cette thèse contribuera à faire de la paramétrisation quad le standard de demain dans l'industrie de l'informatique graphique.

A

Algèbre linéaire pour la géométrie

Sommaire

A.1 Algèbre linéaire géométrique	173
A.1.1 Inverse d'une matrice	174
A.1.2 Application aux calculs d'intersection	174
A.1.3 Pseudo-inverse	175
A.1.4 Factorisations de matrices	175
A.1.5 Orthonormalisation de Gram-Schmidt	176
A.2 Systèmes linéaires contraints	176
A.2.1 Minimisation de formes quadratiques en variables réelles	177
A.2.2 Minimisation de formes quadratiques en variables entières	177
A.2.3 Minimisation de formes quadratiques en variables réelles et entières	178
A.3 Minimisation contrainte	178
A.3.1 Blocage de variable	178
A.3.2 Contraintes linéaires	178
A.3.3 Contraintes quadratiques	179

Cette annexe présente des résultats d'algèbre linéaire importants dans le contexte de l'informatique graphique. Ce sont des résultats connus, mais nous avons essayé de les présenter dans une perspective plus géométrique.

Plan du chapitre

Dans la Section [A.1](#), nous montrons comment l'algèbre linéaire peut être utilisée pour faire des calculs géométriques, ainsi qu'une interprétation géométrique des opérations usuelles de l'algèbre linéaire. Dans la Section [A.2](#), l'algèbre linéaire sera utilisée comme un outil pour résoudre des systèmes d'équations, et nous nous intéresserons à la façon d'y incorporer diverses contraintes.

A.1 Algèbre linéaire géométrique

L'algèbre linéaire est une branche fondamentale des mathématiques. Elle présente un spectre d'applications extrêmement large. Nous allons présenter certains de ses outils que nous utilisons dans cette thèse,

ainsi que leur interprétation géométrique.

A.1.1 Inverse d'une matrice

L'inversion de matrice est l'un des outils les plus fondamentaux et les plus utiles de l'algèbre linéaire. Sa solution est très bien comprise, et ses implémentations très efficaces. Commençons par une interprétation très géométrique des matrices inversibles : une matrice inversible (n, n) peut être vue comme une base de \mathbb{R}^n puisque ses colonnes sont indépendantes. Résoudre $M\mathbf{x} = \mathbf{y}$ correspond alors à trouver les coordonnées \mathbf{x} du vecteur \mathbf{y} dans cette base. Comment faire cela géométriquement ? Si la matrice M est orthonormale (ses colonnes \mathbf{m}_i forment une base orthonormale), les coordonnées de \mathbf{y} dans cette base sont simplement données par les produits scalaires $\mathbf{m}_i \cdot \mathbf{y}$, de sorte que $M^{-1} = M^T$. Si M n'est qu'orthogonale, le produit scalaire doit être divisé par $\|\mathbf{m}_i\|^2$, et l'inverse est simplement :

$$M^{-1} = (\mathbf{m}_1 / \|\mathbf{m}_1\|^2, \dots, \mathbf{m}_n / \|\mathbf{m}_n\|^2)^T. \quad (\text{A.1})$$

Dans le cas général, M n'est pas orthogonale et la solution est donc plus subtile. Si l'on trouve des vecteurs \mathbf{m}_i^* orthogonaux aux \mathbf{m}_j pour $j \neq i$ alors le produit matriciel $(\mathbf{m}_1^*, \dots, \mathbf{m}_n^*)^t M$ est diagonal puisque les termes non diagonaux sont de la forme $\mathbf{m}_i^* \cdot \mathbf{m}_j$ et \mathbf{m}_i^* et \mathbf{m}_j sont orthogonaux. Il suffit alors de diviser chaque \mathbf{m}_i^* par $\mathbf{m}_i^* \cdot \mathbf{m}_i$ pour que les éléments de la diagonale soient tous 1, auquel cas on a bien construit l'inverse de M . Ainsi, inverser une matrice peut être vu géométriquement comme chercher les normales aux hyperplans générés par toutes les colonnes de la matrice sauf une. Le calcul extérieur permet une telle construction : $\mathbf{m}_i^* = \star \wedge_{j \neq i} \mathbf{m}_j$. Enfin, comme $\mathbf{m}_i^* \cdot \mathbf{m}_i = \star(\mathbf{m}_1 \wedge \dots \wedge \mathbf{m}_n) = \text{Det}(M)$ il suffit de diviser chacun de ces vecteurs par $\text{Det}(M)$, ce qui donne une formule géométrique explicite pour l'inverse de M :

$$\text{Det}(M)^{-1} (\mathbf{m}_1^*, \dots, \mathbf{m}_n^*)^t \quad \text{où} \quad \mathbf{m}_i^* = \star \wedge_{j \neq i} \mathbf{m}_j. \quad (\text{A.2})$$

En développant les calculs, on montre que l'équation (A.2) est équivalente à la formule de Kramer (mais plus élégante). Cela montre néanmoins que le calcul extérieur est particulièrement bien adapté au calcul géométrique. En deux dimensions, on a simplement $\mathbf{m}_i^* = \star(\mathbf{m}_{3-i})$ et l'étoile de Hodge est la rotation de $\pi/2$ dans le plan. En trois dimensions, $\mathbf{m}_i^* = \star(\mathbf{m}_{i-1} \wedge \mathbf{m}_{i+1})$, et donc les lignes de l'inverse sont les produit vectoriels de deux colonnes de la matrice.

A.1.2 Application aux calculs d'intersection

En basse dimension, cette interprétation géométrique donne des solutions géométriques simples aux problèmes d'intersection.

Intersection de droites

L'intersection I de deux droites données par un point P_i et un vecteur directeur \mathbf{e}_i ($i = 0, 1$), est la solution de :

$$P_0 + \alpha_0 \mathbf{e}_0 = P_1 + \alpha_1 \mathbf{e}_1 \quad P_1 - P_0 = (\mathbf{e}_0, -\mathbf{e}_1) \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix}$$

et l'on doit donc inverser la matrice $(\mathbf{e}_0, -\mathbf{e}_1)$. Notons \mathbf{v}_{01} le vecteur $P_0 P_1 = P_1 - P_0$. En utilisant (A.2), on a obtenu des expressions géométriques très simples :

$$\begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} = (\mathbf{e}_0, -\mathbf{e}_1)^{-1} \mathbf{v}_{01} = -\frac{1}{\text{Det}(\mathbf{e}_0, \mathbf{e}_1)} (\mathbf{e}_1^\perp, \mathbf{e}_0^\perp)^T \mathbf{v}_{01} = -\frac{1}{\text{Det}(\mathbf{e}_0, \mathbf{e}_1)} \begin{pmatrix} \mathbf{e}_1 \wedge \mathbf{v}_{01} \\ \mathbf{e}_0 \wedge \mathbf{v}_{01} \end{pmatrix},$$

$$I = P_0 - \frac{\mathbf{e}_1 \wedge \mathbf{v}_{01}}{\text{Det}(\mathbf{e}_0, \mathbf{e}_1)} \mathbf{e}_0 = v_1 + \frac{\mathbf{e}_0 \wedge \mathbf{v}_{01}}{\text{Det}(\mathbf{e}_0, \mathbf{e}_1)} \mathbf{e}_1.$$

Intersection de plans

La droite d'intersection de deux plans donnés par un point P_i et une normale \mathbf{n}_i ($i = 0, 1$) est orientée par $\mathbf{n} = \mathbf{n}_0 \times \mathbf{n}_1$. On obtient un point I de cette droite en l'intersectant avec un troisième plan. On peut choisir un plan de normale \mathbf{n} (pour s'assurer de l'intersection), et passant par $P_{01} = (P_0 + P_1)/2$ (pour la symétrie), de sorte que l'intersection est :

$$\begin{pmatrix} \mathbf{n}_0^t \\ \mathbf{n}_1^t \\ \mathbf{n}^t \end{pmatrix} P_{01} I = \frac{1}{2} \begin{pmatrix} -\mathbf{v} \cdot \mathbf{n}_0 \\ \mathbf{v} \cdot \mathbf{n}_1 \\ 0 \end{pmatrix}$$

où $\mathbf{v}_{01} = P_0 P_1 = P_1 - P_0$. Encore une fois, l'utilisation de (A.2) permet une solution géométrique simple :

$$\frac{(\mathbf{v} \cdot \mathbf{n}_1 \mathbf{n}_{010} - \mathbf{v} \cdot \mathbf{n}_0 \mathbf{n}_{101})}{2\mathbf{n}^2}$$

avec $\mathbf{n}_{ijk} = \mathbf{n}_i \times \mathbf{n}_j \times \mathbf{n}_k$, ($i, j, k = 0$ or 1).

A.1.3 Pseudo-inverse

Considérons maintenant une matrice M qui a n lignes mais moins de n colonnes, de sorte que ces colonnes ne génèrent pas \mathbb{R}^n tout entier, mais seulement un sous-espace de \mathbb{R}^n . Dans ce cas, le système $M\mathbf{x} = \mathbf{y}$ n'aura pas de solution exacte en général puisque seuls les vecteurs de $\text{span}(M) \subset \mathbb{R}^n$ peuvent s'écrire comme une combinaison de vecteurs colonnes de M . Cependant, on peut chercher un vecteur \mathbf{x} qui minimise $(M\mathbf{x} - \mathbf{y})^2$, auquel cas $M\mathbf{x}$ sera la projection de \mathbf{y} sur $\text{span}(M)$. Le minimiseur vérifie :

$$\frac{\partial}{\partial \mathbf{x}} (M\mathbf{x} - \mathbf{y})^2 = M^t (M\mathbf{x} - \mathbf{y}) = 0 \quad M^t M\mathbf{x} = M^t \mathbf{y}$$

qui est la projection de $M\mathbf{x} = \mathbf{y}$ sur $\text{span}(M)$. Si les colonnes de M sont indépendantes, $M^t M$ est de rang plein, c'est à dire qu'elle est inversible, et les solutions sont données par :

$$\mathbf{x} = (M^t M)^{-1} M^t \mathbf{y}. \tag{A.3}$$

La matrice $(M^t M)^{-1} M^t = \tilde{M}$ est appelée pseudo-inverse de M .

A.1.4 Factorisations de matrices

Des informations géométriques pertinentes sur les matrices peuvent être obtenues par factorisation. La factorisation exploite la structure et les propriétés des matrices et les exhibent. Les factorisations suivantes nous intéresseront dans le cadre de cette thèse :

- Si A est une matrice (n, m) , on peut trouver des matrices orthonormales O_n et O_m de tailles (n, n) et (m, m) , et une matrice D diagonale de taille (n, m) à coefficients positifs de sorte que $A = O_n D O_m^t$. Ceci est appelé la *décomposition en valeurs singulières* (DVS) et c'est le cas le plus général de factorisation. En effet, la matrice A est complètement arbitraire. En voyant A comme une application de \mathbb{R}^n vers \mathbb{R}^m , la DVS cherche les bases orthonormales O_n et O_m de \mathbb{R}^n et \mathbb{R}^m dans lesquelles A peut être vue

comme une simple mise à l'échelle le long des directions de base. Dans le cas particulier où A est carrée et symétrique, la base O_n est unique pour les deux espaces, et $A = O_n D O_n^T$. Ceci est appelé la *décomposition en vecteur propres* de A , les valeurs propres étant les éléments diagonaux de D et les vecteurs propres sont les vecteurs de la base (colonnes de O_n). Ces factorisations ne sont pas uniques, puisqu'elles dépendent du choix arbitraire de l'ordre des colonnes de O_n . Pour résoudre cette ambiguïté, on demandera généralement que les éléments diagonaux de D soient classés dans l'ordre croissant.

- Si A est une matrice carrée, on peut trouver une matrice triangulaire inférieure L avec des 1 sur sa diagonale, et une matrice triangulaire supérieure U de sorte que $A = LU$. En pratique, on obtient U en ajoutant des multiples des $k - 1$ premières lignes à la k ème ligne pour annuler ses $k - 1$ premiers coefficients. Cette décomposition est unique du fait qu'on impose la diagonale de L à n'avoir que des 1. Si A est symétrique, alors on peut "diviser" U par sa diagonale $D : U = DU_1$. Par symétrie ($A = A^T$) on a alors $LDU_1 = U_1^T DL$ et donc $U_1 = L^T$. La factorisation LU d'une matrice symétrique A peut donc prendre la forme $A = LDL^T = LD^{1/2}D^{1/2}L^T = BB^T$ avec $B = LD^{1/2}$. On l'appelle alors factorisation de *Cholesky*.

A.1.5 Orthonormalisation de Gram-Schmidt

Dans cette thèse, on montre que l'orthonormalité des bases de fonctions est une propriété utile et désirable. Pour construire une base de fonction orthonormale, on peut partir d'une base de fonction usuelle (comme les bases de fonctions interpolantes) et s'en servir pour construire une base orthonormale du même espace. L'orthonormalisation de Gram-Schmidt propose un moyen simple de construire une base orthogonale $\{\mathbf{e}_i\}$ (ainsi que la base orthonormale $\{\bar{\mathbf{e}}_i\}$ correspondante) à partir d'une base arbitraire $\{\mathbf{v}_i\}$. La seule condition sur \mathbf{v}_i est que ce soit une base, c'est à dire que les \mathbf{v}_i soient linéairement indépendants. Pour rendre l'exposition plus claire, on appelle $V_k = \text{span}(\mathbf{v}_1, \dots, \mathbf{v}_k)$ le sous espace généré par les k premiers vecteurs de base. Comme \mathbf{v}_i est une base, V_k est de dimension k . Connaissant une base orthonormale $\{\bar{\mathbf{e}}_i\}_{i=1..k}$ de V_k , l'orthonormalisation consiste à calculer itérativement un nouveau vecteur $\bar{\mathbf{e}}_{k+1}$ de sorte que $\{\bar{\mathbf{e}}_i\}_{i=1..k+1}$ soit encore une base orthonormale de V_{k+1} . Pour ce faire, rappelons que puisque les $\bar{\mathbf{e}}_i$ sont unitaires, tout vecteur \mathbf{v} peut s'écrire :

$$\mathbf{v} = (\mathbf{v} - (\mathbf{v} \cdot \bar{\mathbf{e}}_i)\bar{\mathbf{e}}_i) + (\mathbf{v} \cdot \bar{\mathbf{e}}_i)\bar{\mathbf{e}}_i = o_i(\mathbf{v}) + p_i(\mathbf{v})$$

$p_i(\mathbf{v})$ est la projection de \mathbf{v} sur $\bar{\mathbf{e}}_i$ et $o_i(\mathbf{v})$ est un vecteur orthogonal à $\bar{\mathbf{e}}_i$. Cela nous donne l'itération de l'orthonormalisation Gram-Schmidt :

$$\mathbf{e}_1 = \mathbf{v}_1 \quad \bar{\mathbf{e}}_k = \mathbf{e}_k / \|\mathbf{e}_k\| \quad \mathbf{e}_{k+1} = \mathbf{v}_{k+1} - \sum_{i=1..k} o_i(\mathbf{v}_{k+1}) = \mathbf{v}_{k+1} - \sum_{i=1..k} (\mathbf{v}_{k+1} \cdot \mathbf{e}_i)\mathbf{e}_i$$

Si l'on regroupe les $\{\bar{\mathbf{e}}_i\}$ dans une matrice (orthonormale) O et les $\{\mathbf{v}_i\}$ dans une matrice (de rang plein) V , alors la matrice $V^T O = L$ a pour terme $L_{i,j} = \mathbf{v}_i \cdot \bar{\mathbf{e}}_j = 0$ si $j > i$, et elle est donc triangulaire inférieure. On a donc $V^T O O^T V = V^T V = L^T L$. L'orthonormalisation de Gram-Schmidt peut donc être vue comme la factorisation de Cholesky de la matrice symétrique (définie positive) $V^T V$.

A.2 Systèmes linéaires contraints

Cette section présente maintenant quelques outils simples pour résoudre des systèmes linéaires (au sens des moindres carrés) sous diverses contraintes.

A.2.1 Minimisation de formes quadratiques en variables réelles

Le problème général de la minimisation de formes quadratiques est de trouver le minimiseur \mathbf{x}_{min} et le minimum $E(\mathbf{x}_{min})$ de la forme quadratique :

$$E(\mathbf{x}) = \mathbf{x}^t U \mathbf{x} - \mathbf{x}^t \mathbf{b}$$

avec $\mathbf{x} \in \mathbb{R}^n$. E est minimum si :

$$\frac{\partial E}{\partial \mathbf{x}} = 2U\mathbf{x} - \mathbf{b} = 0 \Leftrightarrow 2U\mathbf{x} = \mathbf{b}$$

Le minimum n'est unique que si U est inversible (sinon il faut rajouter des contraintes). Dans ce cas, le minimiseur est $\mathbf{x}_{min} = U^{-1}\mathbf{b}/2$ et le *min* est $-b^t U^{-1} b/4$. Pour résoudre un système linéaire $A\mathbf{x} = \mathbf{b}$ au moindre carrés, E s'écrit :

$$E(\mathbf{x}) = (A\mathbf{x} - \mathbf{b})^2 = \mathbf{x}^t A^t A \mathbf{x} - 2\mathbf{x}^t A^t \mathbf{b} + \mathbf{b}^2.$$

E est minimum si :

$$\frac{\partial E}{\partial \mathbf{x}} = 2A^t(A\mathbf{x} - \mathbf{b}) = 0 \Leftrightarrow A^t A \mathbf{x} = A^t \mathbf{b}.$$

Encore une fois, la solution n'est unique que si $A^T A$ est inversible. Dans ce cas, on pose :

1. $\tilde{A} = (A^t A)^{-1} A^t$
2. $A^* = A \tilde{A} = A(A^t A)^{-1} A^t$
3. $A^\delta = A^* - Id$

ce qui donne le minimiseur et le minimum suivants :

$$\mathbf{x}_{min} = \tilde{A}\mathbf{b}, \quad E(\mathbf{x}_{min}) = (A^\delta \mathbf{b})^2.$$

Ces matrices ont les propriétés suivantes :

1. $\tilde{A}A = Id$: \tilde{A} est l'inverse à gauche de A ,
2. $A^*A = A$, $A^T A^* = A^T$, $\tilde{A}A^* = \tilde{A}$,
3. $A^\delta A = 0$, $A^T A^\delta = 0$, $\tilde{A}A^\delta = 0$,
4. $A^{*n} = A^*$, $A^{\delta n} = (-1)^n A^\delta$,
5. A^* et A^δ sont symétriques.

A.2.2 Minimisation de formes quadratiques en variables entières

Soit $\mathbf{z} \in \mathbb{Z}^m$ et $F(\mathbf{z}) = (A\mathbf{z} - \mathbf{b})^2$ une forme quadratique en variables entières. On pose $\tilde{\mathbf{x}} = \tilde{A}\mathbf{b}$ et $\mathbf{z}_0 = \mathbf{z} - \tilde{\mathbf{x}}$:

$$F(\mathbf{z}_0) = (A(\mathbf{z}_0 + \tilde{\mathbf{x}}) - \mathbf{b})^2 = (A\mathbf{z}_0 + A^\delta \mathbf{b})^2 = (A\mathbf{z}_0)^2 + 2\mathbf{z}_0^t A^T A^\delta \mathbf{b} + (A^\delta \mathbf{b})^2.$$

Ce problème peut donc se voir de deux façons :

- Trouver le point de la grille $A\mathbb{Z}^m - \mathbf{b}$ le plus proche de l'origine,
- Trouver le plus petite ellipsoïde de centre $\tilde{\mathbf{x}}$ et d'anisotropie \tilde{A} qui contient au moins un point de la grille \mathbb{Z}^m .

Ce problème (que l'on rencontre au Chapitre 9) est connu pour être NP-difficile. Il constitue même un sujet de recherche active en cryptologie. Ainsi, quand on rencontrera ce problème, on utilisera l'heuristique simple qui consiste à arrondir les termes de $\tilde{\mathbf{x}}$ aux valeurs entières les plus proches, ce qui donne la solution exacte dans le cas où A est orthogonale. De plus, elle permet d'obtenir une bonne approximation si A n'est pas trop anisotrope.

A.2.3 Minimisation de formes quadratiques en variables réelles et entières

Considérons maintenant la forme quadratique $\mathbf{g}(\mathbf{x}, \mathbf{z}) = (\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{b})^2$ en variables réelles $\mathbf{x} \in \mathbb{R}^n$ et entières $\mathbf{z} \in \mathbb{Z}^m$. Le minimum en \mathbf{x} est :

$$G(\tilde{\mathbf{A}}(\mathbf{b} - \mathbf{B}\mathbf{z}), \mathbf{z}) = (\mathbf{A}^\delta \mathbf{B}\mathbf{z} - \mathbf{A}^\delta \mathbf{b})^2.$$

Cette formulation permet de réduire ce problème à une minimisation de forme quadratique en variables discrètes seulement.

A.3 Minimisation contrainte

On cherche maintenant le minimiseur de $E(\mathbf{x}) = \mathbf{x}^t \mathbf{U} \mathbf{x} - \mathbf{x}^t \mathbf{b}$ sous un ensemble de m contraintes $C_i(\mathbf{x}) = 0$. Le Lagrangien de ce problème s'écrit :

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = E(\mathbf{x}) + \sum_{i=1}^m \lambda_i C_i(\mathbf{x}) \quad (\text{A.4})$$

Appliquons maintenant cette formule générale à des contraintes simples.

A.3.1 Blocage de variable

On sépare le vecteur de variables \mathbf{x} en un vecteur *libre* \mathbf{x}_f de variables sur lesquelles on n'a pas de contraintes, et un vecteur *bloqué* \mathbf{x}_l sur lesquelles on a imposé $x_i = x_i^p$, avec un ordonnancement des variables de sorte que \mathbf{x} s'écrive :

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_f \\ \mathbf{x}_l \end{pmatrix}.$$

Dans ce cas, le minimiseur contraint est donné par :

$$\begin{aligned} E(\mathbf{x}) &= [\mathbf{x}_f^t, \mathbf{x}_l^t] \begin{pmatrix} U_f & U_{fl} \\ U_{fl}^t & U_l \end{pmatrix} \begin{pmatrix} \mathbf{x}_f \\ \mathbf{x}_l \end{pmatrix} - [\mathbf{x}_f^t, \mathbf{x}_l^t] \begin{pmatrix} \mathbf{b}_f \\ \mathbf{b}_l \end{pmatrix} = \\ &= \mathbf{x}_f^t U_f \mathbf{x}_f + 2\mathbf{x}_f^t U_{fl} \mathbf{x}_l + \mathbf{x}_l^t U_l \mathbf{x}_l - \mathbf{x}_f^t \mathbf{b}_f - \mathbf{x}_l^t \mathbf{b}_l \\ &= \mathbf{x}_f^t U_f^{-1} (\mathbf{b}_f - 2U_{fl} \mathbf{x}_l) \end{aligned}$$

A.3.2 Contraintes linéaires

Les contraintes sont linéaires, i.e. de la forme $\mathbf{l}_i^t \mathbf{x} = c_i$. Dans ce cas, le Lagrangien (A.4) est :

$$\mathcal{L}(x, \boldsymbol{\lambda}) = \mathbf{x}^t \mathbf{U} \mathbf{x} - \mathbf{x}^t \mathbf{b} + \sum_i \lambda_i (\mathbf{l}_i^t \mathbf{x} - c_i) = [\mathbf{x}^t, \boldsymbol{\lambda}^t] \begin{pmatrix} U & \mathbf{l}_i^t/2 \\ \mathbf{l}_i/2 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{pmatrix} - [\mathbf{x}^t, \boldsymbol{\lambda}^t] \begin{pmatrix} \mathbf{b} \\ \mathbf{c} \end{pmatrix}$$

et le minimiseur est donné par :

$$\begin{pmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} U & \mathbf{l}_i^t/2 \\ \mathbf{l}_i/2 & 0 \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{b} \\ \mathbf{c} \end{pmatrix}$$

A.3.3 Contraintes quadratiques

Les contraintes sont quadratiques, i.e. de la forme $\mathbf{x}^t L \mathbf{x} = c_i$. Dans ce cas, le Lagrangien (A.4) est :

$$\mathcal{L}(x, \lambda) = \mathbf{x}^t U \mathbf{x} - \mathbf{x}^t \mathbf{b} + \sum_i \lambda_i (\mathbf{x}^t L_i \mathbf{x} - c_i) = \mathbf{x}^t (U + \sum_i \lambda_i L_i) \mathbf{x} - [\mathbf{x}^t, \lambda^t] \begin{pmatrix} \mathbf{b} \\ \mathbf{c} \end{pmatrix}$$

et le minimiseur vérifie :

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = (U + \sum_i \lambda_i L_i) \mathbf{x} - \mathbf{b} = 0 \quad \frac{\partial \mathcal{L}}{\partial \lambda_i} = \mathbf{x}^t L_i \mathbf{x} - c_i = 0.$$

Dans le cas où l'on a une seule contrainte et $\mathbf{b} = 0$, c'est un problème aux valeurs propres généralisé. Dans le cas contraire, on peut utiliser la méthode de Newton pour trouver le minimum, auquel cas on aura besoin des dérivées secondes :

$$\frac{\partial^2 \mathcal{L}}{\partial \mathbf{x} \partial \mathbf{x}^t} = (U + \sum_i \lambda_i L_i) \quad \frac{\partial^2 \mathcal{L}}{\partial \mathbf{x} \partial \lambda_i} = L_i \mathbf{x} \quad \frac{\partial^2 \mathcal{L}}{\partial \lambda_i \partial \lambda_j} = 0$$

Publications

Cette thèse se base en partie sur trois publications, et une quatrième est acceptée sous réserve de modifications majeures :

1. L'article [[Vallet and Levy 2008](#)] "Spectral Geometry Processing with Manifold Harmonics" (Traitement spectral de la géométrie avec des Harmoniques Variétés) a été publié dans le journal *Computer Graphics Forum* 27(2) et présenté à la conférence Eurographics 2008 par l'auteur. Il correspond à une partie du Chapitre 4 et au Chapitre 5. L'article présente la discrétisation du Laplacien ainsi qu'une méthode efficace pour calculer ses vecteurs propres et leur application au filtrage spectral. C'est le résultat d'un travail commun avec Bruno Lévy.
2. L'article [[Li et al. 2006b](#)] "Representing Higher-Order Singularities in Vector Fields on Piecewise Linear Surfaces" (représenter des singularités d'ordre élevé de champs de vecteur sur des surfaces triangulées) a été publié dans *IEEE Transactions on Visualization and Computer Graphics* et présenté par l'auteur à la conférence Visualization 2006. Il introduit le concept de "saut de période" présenté au Chapitre 7 et la méthode de visualisation du Chapitre 8. C'est le résultat d'un travail commun avec Wan Chiu Li, Nicolas Ray et Bruno Lévy.
3. L'article [[Ray et al. 2008b](#)] "*N*-Symmetry Direction Fields on Surfaces of Arbitrary Genus" (Champs de direction symmétries sur des surfaces de genre arbitraire) va apparaître dans le journal *ACM Transactions On Graphics*. Il présente une grande partie de la théorie du Chapitre 7 et en particulier les "nombre de tours" et les théorèmes sur la topologie. Il introduit aussi l'algorithme du Chapitre 9 qui permet la création de champs de vecteurs avec un contrôle topologique prouvé. C'est le résultat d'un travail commun avec Nicolas Ray, Wan Chiu Li et Bruno Lévy.
4. L'article [[Ray et al. 2008a](#)] "Geometry aware direction field processing" (Traitement géométrique des champs de direction) a été accepté sous réserve de modifications majeures dans *ACM Transactions On Graphics*. Il correspond au Chapitre 10 et présente un algorithme de création de champs de directions avec un contrôle plus faible mais plus global de la topologie. C'est le résultat d'un travail commun avec Nicolas Ray, Laurent Alonso et Bruno Lévy.

Bibliographie

- [Allaire 2002] ALLAIRE, G. 2002. Analyse numérique et optimisation. *Ecole polytechnique*. 18
- [Alliez et al. 2003] ALLIEZ, P., COHEN-STEINER, D., DEVILLERS, O., LEVY, B., AND DESBRUN, M. 2003. Anisotropic polygonal remeshing. *ACM Transactions on Graphics. Special issue for SIGGRAPH conference*, 485–493. 103
- [Andronov et al. 1973] ANDRONOV, A., LEONTOVICH, E., GORDON, I., AND MAIER, A. 1973. Qualitative theory of second-order dynamic systems. *Israel Program For Scientific Translation*. 133
- [Arnold et al. 2006] ARNOLD, D. N., FALK, R. S., AND WINTHER, R. 2006. Finite element exterior calculus, homological techniques, and applications. *Acta Numerica* 15. 42, 59
- [Battke et al. 1997] BATTKE, H., STALLING, D., AND HEGE, H. 1997. Fast line integral convolution for arbitrary surfaces. In *Visualization and Mathematics*, Springer-Verlag, Heidelberg, 181–195. 136
- [Ben-Chen et al. 2008] BEN-CHEN, M., GOTSMAN, C., AND BUNIN, G. 2008. Conformal flattening by curvature prescription and metric scaling. In *IEEE Eurographics, Eurographics*. 155
- [Bloor and Wilson 1990] BLOOR, M., AND WILSON, M. 1990. Using partial differential equations to generate free-form surfaces. *Computer-Aided Design*, 22, 202–212. 76
- [Cabral and Leedom 1993] CABRAL, B., AND LEEDOM, L. 1993. Imaging vector fields using line integral convolution. In *Proceedings of SIGGRAPH*, 263–272. 141, 142, 143
- [Carr and Hart 2002] CARR, N. A., AND HART, J. C. 2002. Meshed atlases for real-time procedural solid texturing. *ACM Transactions on Graphics* 21, 2 (April), 106–131. 136
- [Davis and Hager 2005] DAVIS, T. A., AND HAGER, W. W. 2005. Row modifications of a sparse cholesky factorization. *Journal on Matrix Analysis and Applications* 26, 3, 621–639. 165
- [Desbrun et al. 2005a] DESBRUN, M., KANZO, E., AND TONG, Y. 2005. Discrete differential forms for computational modeling. *Siggraph '05 course notes on Discrete Differential Geometry, Chapter 7*. 42, 123
- [Desbrun et al. 2005b] DESBRUN, M., LEOK, M., AND MARSDEN, J. E. 2005. Discrete poincare lemma. *Applied Numerical Mathematics*, 53. 161

- [Dichler et al. 2002] DICHLER, J.-M., MARITAUD, K., LEVY, B., AND CHAZANFARPOUR, D. 2002. Texture particles. *Computer Graphics Forum* 21, 3. 99
- [Dong et al. 2006] DONG, S., BREMER, P.-T., GARLAND, M., PASCUCCI, V., AND HART, J. C. 2006. Spectral surface quadrangulation. In *SIGGRAPH '06 Proceedings*, ACM Press, New York, NY, USA, 1057–1066. 59, 81
- [Donnelly and Fefferman 1988] DONNELLY, H., AND FEFFERMAN, C. 1988. Nodal sets of eigenfunctions on riemannian manifolds. *Invent. Math.* 93, 1, 161–183. 80
- [Firby and Gardiner 1982] FIRBY, P., AND GARDINER, C. 1982. *Surface Topology*. Ellis Horwood Ltd., ch. 7, 115–135. 133
- [Fisher et al. 2007] FISHER, M., SCHRÖDER, P., DESBRUN, M., AND HOPPE, H. 2007. Design of tangent vector fields. In *SIGGRAPH '07 : ACM SIGGRAPH 2007 papers*, ACM, New York, NY, USA, 56. 105, 106
- [Floater and Hormann 2004] FLOATER, M. S., AND HORMANN, K. 2004. *Surface Parameterization : a Tutorial and Survey*. N. A. Dodgson, M. S. Floater, and M. A. Sabin (eds.), Springer-Verlag, Heidelberg, 157–186. 136
- [Forszell and Cohen 1995] FORSELL, L., AND COHEN, S. 1995. Using line integral convolution for flow visualization : Curvilinear grids, variable-speed animation, and unsteady flows. *IEEE Transactions on Visualization and Computer Graphics* 1, 2 (June), 133–141. 136
- [Forszell 1994] FORSELL, L. 1994. Visualizing flow over curvilinear grid surfaces using line integral convolution. In *Proceedings of IEEE Visualization*, 240–247. 136
- [Gortler et al. 1996] GORTLER, S., GRZESZCZUK, R., SZELISKI, R., AND COHEN, M.-F. 1996. The lumigraph. In *Proceedings of ACM SIGGRAPH*, 43–54. 102
- [Grabner and Laramée 2005] GRABNER, M., AND LARAMEE, R. S. 2005. Image space advection on graphics hardware. In *Proceedings of the 21st Spring Conference on Computer Graphics and its Applications*, 75–82. 143
- [Grinspun et al. 2002] GRINSPUN, E., KRYSL, P., AND SCHRÖDER, P. 2002. CHARMS : a simple framework for adaptive simulation. In *SIGGRAPH Proceedings*, vol. 21, 281–290. 89
- [Gu and Yau 2003] GU, X., AND YAU, S.-T. 2003. Global conformal surface parameterization. In *Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, 127–137. 103
- [Gu et al. 2002] GU, X., GORTLER, S., AND HOPPE, H. 2002. Geometry images. In *SIGGRAPH '02 : ACM SIGGRAPH 2002 papers*, 355–361. 165
- [Guskov et al. 1999] GUSKOV, I., SWELDENS, W., AND SCHRÖDER, P. 1999. Multiresolution signal processing for meshes. *Computer Graphics Proceedings (SIGGRAPH 99)*, 325–334. 76
- [Heidrich et al. 1999] HEIDRICH, W., WESTERMANN, R., SEIDEL, H., AND ERTL, T. 1999. Applications of pixel textures in visualization and realistic image synthesis. In *Proceedings of Symposium on Interactive 3D Graphics*, 127–134. 136
- [Hertzmann and Zorin 2000] HERTZMANN, A., AND ZORIN, D. 2000. Illustrating smooth surfaces. In *Proceedings of ACM SIGGRAPH*, 517–526. 98, 104, 110, 121, 153

-
- [Hildebrandt et al. 2006] HILDEBRANDT, K., POLTHIER, K., AND WARDETZKY, M. 2006. On the convergence of metric and geometric properties of polyhedral surfaces. *Geom Dedicata*. 59
- [Hirani 2003] HIRANI, A. N. 2003. *Discrete Exterior Calculus*. PhD thesis, California Institute of Technology, Pasadena, California. 42
- [Interrante 1997] INTERRANTE, V. 1997. Illustrating surface shape in volume data via principal direction-driven 3D line integral convolution. In *Proceedings of SIGGRAPH*, 109–116. 136
- [Kalberer et al. 2007] KALBERER, F., NIESER, M., AND POLTHIER, K. 2007. Quadcover - surface parameterization using branched coverings. *Computer Graphics Forum* 26, 3 (September), 375–384. 100, 169
- [Karni and Gotsman 2000] KARNI, Z., AND GOTSMAN, C. 2000. Spectral compression of mesh geometry. In *SIGGRAPH '00 : Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 279–286. 59, 81, 87
- [Kim and Rossignac 2005] KIM, B., AND ROSSIGNAC, J. 2005. GeoFilter : Geometric Selection of Mesh Filter Parameters. *Computer Graphics Forum* 24, 3, 295–302. 75, 83, 84
- [Kobbelt et al. 1998] KOBBELT, L., CAMPAGNA, S., VORSATZ, J., AND SEIDEL, H. 1998. Interactive multi-resolution modeling on arbitrary meshes. In *SIGGRAPH Conference Proceedings*, 105–114. 76
- [Kobbelt 1997] KOBBELT, L. 1997. Discrete fairing. In *Proceedings of the Seventh IMA Conference on the Mathematics of Surfaces*, 101–131. 76
- [Laramee et al. 2003] LARAMEE, R. S., SCHNEIDER, J., AND HAUSER, H. 2003. Image space based visualization of unsteady flow on surfaces. In *Proceedings of IEEE Visualization*, 131–138. 136, 141
- [Laramee et al. 2004] LARAMEE, R. S., VAN WIJK, J., JOBARD, B., AND H.HAUSER. 2004. ISA and IBFVS : Image space-based visualization of flow on surfaces. *IEEE Transactions on Visualization and Computer Graphics* 10, 6, 637–648. 136
- [Lee et al. 1998] LEE, A. W. F., SWELDENS, W., SCHRÖDER, P., COWSAR, L., AND DOBKIN, D. 1998. Maps : Multiresolution adaptive parameterization of surfaces. *Computer Graphics Proceedings (SIGGRAPH 98)*, 95–104. 76
- [Levy 2005] LEVY, B. 2005. Numerical methods for digital geometry processing. In *Israel Korea Bi-National Conference - Invited talk*. 105
- [Levy 2006] LEVY, B. 2006. Laplace-beltrami eigenfunctions : Towards an algorithm that understands geometry. In *IEEE International Conference on Shape Modeling and Applications*. 59
- [Li et al. 2006a] LI, W.-C., RAY, N., AND LEVY, B. 2006. Automatic and interactive mesh to t-spline conversion. In *EG/ACM Symposium on Geometry Processing*. 104, 105, 160, 165
- [Li et al. 2006b] LI, W.-C., VALLET, B., RAY, N., AND LEVY, B. 2006. Representing higher-order singularities in vector fields on piecewise linear surfaces. *IEEE Transactions on Visualization and Computer Graphics (proc. Vis 06)*. 3, 94, 132, 181

- [Mallet 1992] MALLET, J. 1992. Discrete Smooth Interpolation. *Computer Aided Design* 24, 4, 263–270. 76
- [Meshar et al. 2006] MESHAR, O., IRONY, D., AND TOLEDO, S. 2006. An out-of-core sparse symmetric indefinite factorization method. *ACM Transactions on Mathematical Software* 32, 445–471. 82
- [Meyer et al. 2003] MEYER, M., DESBRUN, M., SCHRÖDER, P., AND BARR, A. H. 2003. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III*, H.-C. Hege and K. Polthier, Eds. Springer-Verlag, Heidelberg, 35–57. 59
- [Mousa et al. 2006] MOUSA, M., CHAINE, R., AND AKKOUICHE, S. 2006. Direct spherical harmonic transform of a triangulated mesh. *GRAPHICS-TOOLS 11*, 2, 17–26. 76
- [Mullen et al. 2006] MULLEN, P., MCKENZIE, A., TONG, Y., AND DESBRUN, M. 2006. A variational approach to eulerian geometry processing. *ACM Transactions on Graphics (SIGGRAPH '06)*. 40
- [Nealen et al. 2006] NEALEN, A., IGARASHI, T., SORKINE, O., AND ALEXA, M. 2006. Laplacian mesh optimization. In *Proceedings of ACM GRAPHITE*, 381–389. 76
- [Ni et al. 2004] NI, X., GARLAND, M., AND HART, J. C. 2004. Fair morse functions for extracting the topological structure of a surface mesh. *ACM Transactions on Graphics (SIGGRAPH '04)* 23, 3, 613–622. 103
- [Nielson 1979] NIELSON, G. 1979. The side-vertex method for interpolation in triangles. *Journal of Approximation Theory* 25, 4, 318–336. 138
- [Ohtake et al. 2001] OHTAKE, Y., HORIKAWA, M., AND BELYAEV, A. 2001. Adaptive smoothing tangential direction fields on polygonal surfaces. In *Proceedings of Pacific Graphics*, 189. 103
- [Palacios and Zhang 2007] PALACIOS, J., AND ZHANG, E. 2007. Rotational symmetry field design on surfaces. *ACM Transactions on Graphics (SIGGRAPH '07)*, to appear. 106, 107, 165
- [Pauly and Gross 2001] PAULY, M., AND GROSS, M. 2001. Spectral processing of point sampled geometry. In *SIGGRAPH Proceedings*. 74, 76
- [Pinkall and Polthier 1993] PINKALL, U., AND POLTHIER, K. 1993. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics* 2, 1. 59
- [Polthier and Preuss 2002] POLTHIER, K., AND PREUSS, E. 2002. Identifying vector fields singularities using a discrete hodge decomposition. In *Visualization and Mathematics III*, H. Hege and K. Polthier, Eds. Springer Verlag, 113–134. 103
- [Prathap 1999] PRATHAP, G. 1999. Towards a science of fea : Patterns, predictability and proof through some case studies. *Current Science* 77, 1311–1318. 62
- [Praun et al. 2000] PRAUN, E., FINKELSTEIN, A., AND HOPPE, H. 2000. Lapped textures. In *Siggraph 2000, Computer Graphics Proceedings*, ACM Press / ACM SIGGRAPH / Addison Wesley Longman, K. Akeley, Ed., 465–470. 99, 102
- [Praun et al. 2001] PRAUN, E., HOPPE, H., WEBB, M., AND FINKELSTEIN, A. 2001. Real-time hatching. In *SIGGRAPH 2001, Computer Graphics Proceedings*, E. Fiume, Ed., 579–584. 103

-
- [Ray et al. 2006] RAY, N., LI, W. C., LEVY, B., SHEFFER, A., AND ALLIEZ, P. 2006. Periodic global parameterization. *ACM Transactions on Graphics*. 104, 107, 121, 153
- [Ray et al. 2008a] RAY, N., VALLET, B., ALONSO, L., AND LEVY, B. 2008. Geometry aware direction field processing. In *ALICE Tech. Report*. 3, 94, 159, 181
- [Ray et al. 2008b] RAY, N., VALLET, B., LI, W.-C., AND LEVY, B. 2008. N-symmetry direction fields on surfaces of arbitrary genus. In *ACM Transactions on graphics*. 3, 94, 126, 144, 148, 161, 165, 181
- [Ray et al. 2008c] RAY, N., WANG, H., VALLET, B., TURK, G., AND LEVY, B. 2008. Material space texturing. In *ALICE Tech. Report*. 93
- [Reuter et al. 2005] REUTER, M., WOLTER, F.-E., AND PEINECKE, N. 2005. Laplace-spectra as fingerprints for shape matching. In *SPM '05 : Proceedings of the 2005 ACM symposium on Solid and physical modeling*, ACM Press, New York, NY, USA, 101–106. 59
- [Rezk-Salama et al. 1999] REZK-SALAMA, C., HASTREITER, P., CHRISTIAN, T., AND ERTL, T. 1999. Interactive exploration of volume line integral convolution based on 3D-texture mapping. In *Proceedings of IEEE Visualization*, 233–240. 136
- [Rusinkiewicz et al. 2006] RUSINKIEWICZ, S., BURNS, M., AND DECARLO, D. 2006. Exaggerated shading for depicting shape and detail. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 25, 3 (July). 84
- [Scheuermann et al. 1998] SCHEUERMANN, G., KRUGER, H., MENZEL, M., AND ROCKWOOD, A. 1998. Visualizing non-linear vector field topology. *IEEE Transactions on Visualization and Computer Graphics* 4, 2, 109–116. 135
- [Scheuermann et al. 1999] SCHEUERMANN, G., TRICOCHÉ, X., AND HAGEN, H. 1999. C1-interpolation for vector field topology interpolation. In *Proceedings of IEEE Visualization*, 271–278. 135
- [Schroeder et al. 2005] SCHROEDER, W., BERTEL, F., MALATERRE, M., THOMPSON, D., AND PEBAY, P. 2005. Framework for visualizing higher-order basis functions. In *Proceedings of IEEE Visualization*, 43–50. 132
- [Sorkine et al. 2005] SORKINE, O., COHEN-OR, D., IRONY, D., AND TOLEDO, S. 2005. Geometry-aware bases for shape approximation. *IEEE Transactions On Visualization And Computer Graphics*. 76
- [Stalling and Hege 1995] STALLING, D., AND HEGE, H. 1995. Fast and resolution independent line integral convolution. In *Proceedings of SIGGRAPH*, 249–256. 136
- [Taubin 1995] TAUBIN, G. 1995. A signal processing approach to fair surface design. In *SIGGRAPH '95 : Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 351–358. 59, 73, 76
- [Theisel 2002] THEISEL, H. 2002. Designing 2D vector fields of arbitrary topology. *Computer Graphics Forum (Proceedings Eurographics 2002)* 21, 3, 595–604. 134
- [Tong et al. 2003] TONG, Y., LOMBAYDA, S., HIRANI, A., AND DESBRUN, M. 2003. Discrete multiscale vector field decomposition. *ACM Transactions on Graphics (SIGGRAPH '03)* 22, 3, 445–452. 103

- [Tong et al. 2006] TONG, Y., ALLIEZ, P., COHEN-STEINER, D., AND DESBRUN, M. 2006. Designing quadrangulations with discrete harmonic forms. In *Symposium on Geometry Processing*, Eurographics, 201–210. 100
- [Tricoche et al. 2000a] TRICOCHÉ, X., SCHEUERMANN, G., AND HAGEN, H. 2000. Higher order singularities in piecewise linear vector fields. In *The Mathematics of Surfaces IX*, Springer, London, 99–113. 133, 134
- [Tricoche et al. 2000b] TRICOCHÉ, X., SCHEUERMANN, G., AND HAGEN, H. 2000. A topology simplification method for 2d vector fields. In *Proceedings of IEEE Visualization*, 359–366. 133, 134
- [Tricoche et al. 2003] TRICOCHÉ, X., SCHEUERMANN, G., AND HAGEN, H. 2003. Topology simplification of symmetric, second-order 2D tensor fields. In *Hierarchical and Geometrical Methods in Scientific Visualization*. 103
- [Tricoche 2002] TRICOCHÉ, X. 2002. *Vector and Tensor Topology Simplification, Tracking, and Visualization*. PhD thesis, Schriftenreihe Fachbereich Informatik (3), University of Kaiserslautern. 120, 132
- [Turk 2001] TURK, G. 2001. Texture synthesis on surfaces. In *SIGGRAPH '01 Proceedings*, ACM Press, New York, NY, USA, 347–354. 99, 102
- [Vallet and Levy 2008] VALLET, B., AND LEVY, B. 2008. Manifold harmonics. 3, 53, 75, 181
- [van Wijk 2003] VAN WIJK, J. 2003. Image based flow visualization for curved surfaces. In *Proceedings of IEEE Visualization*, 123–130. 136
- [Wang et al. 2006] WANG, K., WEIWEI, TONG, Y., DESBRUN, M., AND SCHRÖDER, P. 2006. Edge subdivision schemes and the construction of smooth vector fields. *ACM Transactions on Graphics (SIGGRAPH '06)*. 103
- [Wardetzky et al. 2007a] WARDETZKY, M., BERGOU, M., HARMON, D., ZORIN, D., AND GRINSPUN, E. 2007. Discrete quadratic curvature energies. *Computer Aided Geometric Design (CAGD)*. 59, 67
- [Wardetzky et al. 2007b] WARDETZKY, M., MATHUR, S., KALBERER, F., AND GRINSPUN, E. 2007. Discrete laplace operators : No free lunch. *Eurographics Symposium on Geometry Processing*. 59, 70, 170
- [Wei and Levoy 2001] WEI, L.-Y., AND LEVOY, M. 2001. Texture synthesis over arbitrary manifold surfaces. In *Proceedings of ACM SIGGRAPH*, 355–360. 99, 121
- [Weinkauff et al. 2004] WEINKAUFF, T., THEISEL, H., HEGE, H.-C., AND SEIDEL, H.-P. 2004. Topological construction and visualization of higher order 3d vector fields. *Computer Graphics Forum (Proceedings Eurographics 2004)* 23, 3, 469–478. 134
- [Weinkauff et al. 2005] WEINKAUFF, T., THEISEL, H., SHI, K., HEGE, H.-C., AND SEIDEL, H.-P. 2005. Extracting higher order critical points and topological simplification of 3d vector fields. In *Proceedings of IEEE Visualization*, 559–566. 134
- [Wu and Kobbelt 2005] WU, J., AND KOBBELT, L. 2005. Efficient spectral watermarking of large meshes with orthogonal basis functions. In *The Visual Computer*. 81
- [Zahariev 2007] ZAHARIEV, S. 2007. Discrete connection laplacians. Tech. rep. 48
- [Zelinka and Garland 2004] ZELINKA, S., AND GARLAND, M. 2004. Jump map-based interactive texture synthesis. *ACM Transactions on Graphics* 23, 4, 930–962. 103

-
- [Zhang et al. 2004] ZHANG, E., MISCHAIKOW, K., AND TURK, G. 2004. Vector field design on surfaces. Tech. rep., GVU 04-16, Georgia Institute of Technology. Accepted by *ACM Transactions on Graphics*, pending revision. 135
- [Zhang et al. 2005] ZHANG, E., HAYS, J., AND TURK, G. 2005. Interactive design and visualization of tensor fields on surfaces. Tech. rep., Oregon State University. 135
- [Zhang et al. 2006] ZHANG, E., MISCHAIKOW, K., AND TURK, G. 2006. Vector field design on surfaces. In *ACM Transactions on Graphics*, 1294–1326. 106
- [Zhang et al. 2007] ZHANG, E., HAYS, J., AND TURK, G. 2007. Interactive design and visualization of tensor fields on surfaces. In *IEEE Transactions on Visualization and Computer Graphics*, 94–107. 106
- [Zhang 2004] ZHANG, H. 2004. Discrete combinatorial laplacian operators for digital geometry processing. In *Proc. SIAM Conference on Geometric Design and Computing*. 59, 106
- [Zhou et al. 2004] ZHOU, K., BAO, H., AND SHI, J. 2004. 3d surface filtering using spherical harmonics. In *Computer Aided Design* 36, 363–375. 76

AUTORISATION DE SOUTENANCE DE THESE
DU DOCTORAT DE L'INSTITUT NATIONAL
POLYTECHNIQUE DE LORRAINE

o0o

VU LES RAPPORTS ETABLIS PAR :

Monsieur Claude PUECH, Directeur de Recherche, INRIA, Le Chesnay

Monsieur Leif KOBELT, Professeur, RWTH-Aachen, Allemagne

Le Président de l'Institut National Polytechnique de Lorraine, autorise :

Monsieur VALLET Bruno

à soutenir devant un jury de l'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE,
une thèse intitulée :

"Bases de fonctions sur les variétés"

en vue de l'obtention du titre de :

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

Spécialité : « **Informatique** »

Fait à Vandoeuvre, le 02 juillet 2008

Le Président de l'I.N.P.L.,

F. LAURENT



NANCY BRABOIS
2, AVENUE DE LA
FORET-DE-HAYE
BOITE POSTALE 3
F - 5 4 5 0 1
VANDŒUVRE CEDEX