

# Modélisation géométrique itérative sous contraintes

THÈSE N° 4575 (2009)

PRÉSENTÉE LE 14 JANVIER 2010

À LA FACULTÉ ENVIRONNEMENT NATUREL, ARCHITECTURAL ET CONSTRUIT  
LABORATOIRE DE CONSTRUCTION EN BOIS  
PROGRAMME DOCTORAL EN STRUCTURES

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Gilles GOUATY

acceptée sur proposition du jury:

Prof. I. Smith, président du jury  
Prof. Y. Weinand, Dr E. Tosan, directeurs de thèse  
Dr C. Gentil, rapporteur  
Prof. P. Lienhardt, rapporteur  
Prof. D. Thalmann, rapporteur



ÉCOLE POLYTECHNIQUE  
FÉDÉRALE DE LAUSANNE

Suisse  
2010



## Résumé

Les modèles itératifs sont très largement utilisés aujourd’hui en CAO. Ils permettent, avec un nombre restreint de paramètres, de représenter des formes relativement complexes par l’intermédiaire d’un algorithme qui applique un certain procédé de subdivision de manière itérative. Il existe une grande variété de tels modèles (Catmull-Clark, Doo-Sabin, L-Systèmes . . .).

La plupart des modèles itératifs utilisés en CAO permettent de représenter des formes lisses, de type polynomiales ou bien rationnelles. Le modèle IFS (Iterated Function System) est un modèle mathématique permettant de représenter des formes pouvant être lisses, dans des cas particuliers, mais qui sont plus généralement fractales. Un IFS est défini par un ensemble d’opérateurs géométriques dits “opérateurs de subdivision”. Ces opérateurs définissent une forme de manière itérative, en appliquant successivement cet ensemble d’opérateurs de subdivision sur un objet géométrique de base. Les propriétés topologiques ou géométriques des formes générées par IFS sont relativement difficiles à contrôler. Certains logiciels permettent de créer des formes par IFS. Beaucoup de ces logiciels ont des applications purement graphiques, et très peu permettent des applications à la réalisation physique des objets modélisés [SYC06]. Ceci est dû notamment à la difficulté à contrôler les propriétés des formes modélisées.

Nous définissons un modèle plus général, le modèle BCIFS (Boundary Controlled Iterative Function System) qui permet de représenter, dans un même formalisme, une gamme très large de modèles itératifs, dont le modèle IFS, ainsi que la plupart des modèles classiques utilisés en CAO actuellement. Ce formalisme utilise une représentation sous la forme d’un automate dont chaque état correspond à différentes parties ou sous-parties de l’objet modélisé. Elles sont reliées entre elles par des relations de subdivision et des relations d’incidence, qui constituent les arcs de l’automate. Enfin, on établit des relations d’adjacence en introduisant une relation d’équivalence entre certains chemins de l’automate.

Ce formalisme permet de spécifier certaines propriétés du modèle itératif que l’on choisit de définir. Cela concerne notamment la structure topologique de l’objet modélisé (courbe, surface, volume, ou autre. . .), mais également la structure de raccord des éléments adjacents (exemples : raccords sur deux points pour les B-Splines, trois points pour les schémas de Catmull-Clark, . . .).

Les schémas de subdivision classiques permettent de représenter des formes lisses. Ils prennent comme paramètres un ensemble de points de contrôle, pouvant être positionnés n’importe où dans l’espace. La donnée de ces points de contrôle sert d’entrée à l’algorithme de subdivision, qui utilise des matrices prédéfinies pour calculer les nouveaux points.

Contrairement à cela, dans le modèle BCIFS, les opérateurs de subdivision ne sont pas prédéfinis, mais sont paramétrables. Ces nouveaux paramètres peuvent être représentés graphiquement comme des points que l’on peut positionner dans l’espace, à la manière des points de contrôle. Chacun de ces points, que l’on désignera par le terme “point de subdivision”, est l’image d’un point de contrôle par l’intermédiaire d’un opérateur de subdivision. Contrairement aux points de contrôle dont le positionnement contrôle l’aspect global de l’objet modélisé, la modification des points de subdivision affecte l’objet à chaque niveau de subdivision, et donc à des échelles de plus en plus petites. Les formes obtenues ne sont pas forcément lisses mais peuvent être fractales, et ceci dans le cas le plus général.

Par ailleurs, les contraintes dues à la construction nécessitent un contrôle plus précis des propriétés géométriques des objets modélisés. Dans le cadre de la construction en bois, nous souhaitons notamment réaliser des structures surfaciques par assemblage de panneaux de bois. Cela nécessite de modéliser des maillages composés de faces planes.

Nous nous intéressons plus particulièrement à la modélisation de maillages quadrangulaires. La triangulation de maillages pour obtenir des faces planes est une solution que nous écartons. Ceci provient des contraintes liées à la construction, et plus particulièrement de

la complexité à réaliser des assemblages autour de sommets de valence élevée. Les maillages triangulés, par exemple, présentent une valence moyenne de six tandis que les maillages à base de carreaux ont des sommets de valence quatre.

L'élaboration et la réalisation des solutions constructives sont relativement coûteuses en fonction de la valence du nœud. La valence des nœuds d'un maillage a une influence directe sur la géométrie des faces : plus la valence d'un sommet est élevée, plus les angles des faces avoisinant ces sommets seront aigus. Les faces à angles aigus ne sont pas souhaitables pour une application constructive parce que les éléments constructifs ont des coins fragiles et leur manipulation lors du processus de mise en œuvre devient une opération délicate.

Nous proposons une méthode basée sur un modèle itératif qui génère directement des maillages quadrangulaire à faces planes. Nous partons d'une somme de Minkowski de deux courbes. Nous étendons les possibilités de création de formes en travaillant dans un espace  $4D$  de coordonnées homogènes, et en projetant les figures obtenues dans l'espace de modélisation  $3D$ . L'utilisation de la géométrie projective permet d'étendre la méthode par des paramètres supplémentaires comme le poids de certains points. Cela permet d'obtenir une gamme de maillages surfaciques relativement générale.

### Mots clés

Informatique graphique, modélisation géométrique, modèle itératif, IFS, schéma de subdivision, contraintes topologiques, saisie interactive, maillages surfaciques, contraintes géométriques, coplanarité, maillages quadrangulaires à faces planes, architecture, construction en bois.

---

Ce projet de recherche a été financé par le Fonds National Suisse de la recherche scientifique (FNS 200021-112103 et 200020-120037).

## Abstract

Iterative models are widely used today in CAD. They allow, with a limited number of parameters, to represent relatively complex forms through a subdivision algorithm. There is a wide variety of such models (Catmull-Clark, Doo-Sabin, L-Systems . . .). Most iterative models used in CAD can represent smooth shapes, such as polynomial or rational.

The IFS model (Iterated Function System) is a mathematical model allowing to represent objects that can be smooth, in particular cases, or fractal, in more general cases. An IFS is defined by a set of geometric operators called “subdivision operators”. These operators define an object iteratively, by successively applying this set of subdivision operators on a geometric base object.

Classical subdivision schemes take as parameters a set of control points, that can be moved anywhere in space. These control points are the entry parameters of the subdivision algorithm, which uses predefined subdivision matrices to calculate the new points.

In the IFS model, subdivision operators are not predefined, but customizable. These new parameters are graphically represented as movable points in space, like the control points. Each of these points, referred to as “subdivision point” is the image of a control point through a subdivision operator. The position of the control points allow to control the global aspect of the modelled object. Moving subdivision points affects the object at each level of subdivision, and therefore at smaller and smaller scales. The generated objects are not necessarily smooth, but are generally fractal.

The constraints due to construction require some precise geometric properties of the modelled objects. As part of the wooden building, we want to achieve particular surface structures by assembly of wood panels. This requires modelling meshes composed of planar faces.

We are particularly interested in modelling quadrangular mesh. We discard the solution of triangular meshes. This comes from constraints related to construction, and is more particularly due to the complexity of realizing assemblies around high valence vertices. The vertices in triangulated meshes have a valence of six, while in quadrangular meshes they have a valence of four.

The development and implementation of solutions are relatively expensive in terms of the valence of the node. The valence of the nodes of a mesh has a direct influence on the geometry of faces; the higher the valence of a vertex, the higher angles of faces around this vertex will be acute. Faces with acute angles are not desirable for a constructive application, because constructive elements have fragile parts and handling them during the implementation process is a delicate operation.

We propose a method based on an iterative model that generates directly planar-quadrilateral meshes. We start from a Minkowski sum of two curves. This process is rather limited, because it generates meshes only composed by parallelograms. We expand the possibilities for creating forms, working in a  $4D$  homogeneous coordinate system, and projecting these forms in the  $3D$  modelling space. Using projective geometry allows to extend the method by additional parameters such as the weight of points. This allows to reach a relatively general range of surface meshes.

## Keywords

Computer graphics, geometric modeling, iterative models, IFS, subdivision schemes, topological constraints, interactive capture, surface meshes, geometric constraints, coplanarity, quadrangular planar meshes, architecture, timber construction.

---

This research was funded by the Swiss National Science Foundation (FNS 200021-112103 and 200020-120037).



# Table des matières

Résumé . . . . .	3
Abstract . . . . .	5
Notations . . . . .	11
<b>1 Introduction</b>	<b>13</b>
1.1 Contexte . . . . .	13
1.2 Problématique . . . . .	13
1.3 État de l'art . . . . .	14
1.3.1 IFS et généralisations . . . . .	14
1.3.2 Schémas de subdivision . . . . .	14
1.3.3 Modélisation fractale . . . . .	15
1.4 Objectifs . . . . .	15
1.4.1 Contraintes topologiques . . . . .	15
1.4.2 Contrôle géométrique . . . . .	16
1.4.3 Contraintes de fabrication . . . . .	16
1.4.4 Modèle fractal . . . . .	16
<b>2 Modèles itératifs basée sur les IFS : rappels</b>	<b>17</b>
2.1 Le modèle IFS . . . . .	17
2.1.1 Définition ensembliste . . . . .	17
2.1.2 Exemples d'IFS . . . . .	17
2.1.3 Algorithme itératif pour le calcul d'un attracteur . . . . .	21
2.1.4 Fonction d'adressage . . . . .	22
2.2 Le modèle CIFS (Controlled Iterated Function System) . . . . .	22
2.2.1 Données du modèle . . . . .	22
2.2.2 Définition ensembliste des attracteurs . . . . .	23
2.2.3 Algorithme itératif pour le calcul d'un attracteur . . . . .	27
2.2.4 Langages et fonction d'adressage . . . . .	28
2.2.5 Exemples d'utilisations du modèle CIFS . . . . .	30
<b>3 Contrôle topologique</b>	<b>33</b>
3.1 Le modèle BCIFS (Boundary Controlled Iterated Function System) . . . . .	34

3.1.1	B-Rep (Boundary representation) . . . . .	34
3.1.2	Formulation itérative de la B-Rep . . . . .	35
3.1.3	Formalisation du modèle BCIFS . . . . .	40
3.1.4	Arbre de décomposition, et graphe quotient . . . . .	42
3.2	Topologies classiques . . . . .	43
3.2.1	Arc de courbe . . . . .	43
3.2.2	Carreau quadrangulaire . . . . .	46
3.2.3	Carreau triangulaire . . . . .	50
3.3	Problèmes d'orientation et de symétrie . . . . .	55
3.3.1	Opérateurs de symétrie . . . . .	55
3.3.2	Contraintes de symétrie . . . . .	56
3.3.3	Exemple du carreau triangulaire . . . . .	57
3.4	Topologies fractales . . . . .	59
3.4.1	Triangle de SIERPINSKI . . . . .	59
3.4.2	Arbre . . . . .	60
<b>4</b>	<b>Plongement géométrique</b> . . . . .	<b>65</b>
4.1	Introduction . . . . .	65
4.1.1	Conventions . . . . .	65
4.1.2	Lien avec les formes à pôles . . . . .	66
4.1.3	Exemple d'une courbe . . . . .	67
4.2	Résolution des contraintes topologiques . . . . .	71
4.2.1	Construction des espaces . . . . .	72
4.2.2	Matrices de subdivision . . . . .	80
4.2.3	Exemple d'une courbe . . . . .	81
4.3	Exemples de formes à pôles . . . . .	84
4.3.1	Courbe avec raccord d'ordre 2 . . . . .	84
4.3.2	Courbe avec raccord d'ordre 2 et avec chevauchements . . . . .	88
4.3.3	Courbe symétrique . . . . .	93
4.3.4	Carreau quadrangulaire . . . . .	94
4.3.5	Planaire de HILBERT . . . . .	101
4.4	Saisie interactive . . . . .	107
4.4.1	Matrice globale . . . . .	107
4.4.2	Saisie de points . . . . .	108
4.4.3	Masques . . . . .	108
4.4.4	Paramètres indépendants . . . . .	109
<b>5</b>	<b>Contraintes géométriques</b> . . . . .	<b>111</b>
5.1	Introduction . . . . .	111
5.1.1	Contexte . . . . .	111



5.1.2	Contraintes liées à la construction . . . . .	111
5.1.3	Travaux précédents . . . . .	112
5.1.4	Méthode . . . . .	112
5.2	Définition de surfaces par somme de courbes . . . . .	113
5.2.1	Homogénéisation d'un espace affine . . . . .	113
5.2.2	Somme centrée . . . . .	113
5.2.3	Interprétation en géométrie affine . . . . .	114
5.2.4	Interprétation en géométrie projective . . . . .	116
5.3	Modélisation itérative . . . . .	119
5.3.1	Modèle itératif des courbes et des surfaces . . . . .	119
5.3.2	Calcul itératif des maillages . . . . .	119
5.3.3	Interprétation en géométrie affine . . . . .	120
5.3.4	Interprétation en géométrie projective . . . . .	122
5.4	Application à la construction . . . . .	124
<b>6</b>	<b>Conclusion</b>	<b>127</b>
6.1	Topologie . . . . .	127
6.2	Géométrie . . . . .	128
6.3	Contraintes de fabrication . . . . .	129
	<b>Annexe</b>	<b>131</b>
	Un langage de description de modèles itératifs . . . . .	131



## Notations

Notation	Signification	Propriété générale
$E$	Espace affine	
$(E, d)$	Espace métrique complet	
$\mathcal{H}(E)$	Ensemble des compacts non vides de $E$	
$d_H$	Distance de HAUSDORFF	
$\mathcal{S}$	Semigroupe des opérateurs contractants sur $E$	
$\Sigma$	Alphabet	
$T_i$	Opérateur dans $\mathcal{S}$	$i \in \Sigma$ $T_i : E \leftarrow E$
$\mathcal{T}$	Opérateur de HUTCHINSON	$\mathcal{T} : \mathcal{H}(E) \leftarrow \mathcal{H}(E)$ $\mathcal{T}(\kappa) = \bigcup_{i \in \Sigma} T_i(\kappa)$
$\mathcal{A}$	Attracteur d'un IFS	$\mathcal{A} \in \mathcal{H}(E)$ $\mathcal{A} = \mathcal{T}(\mathcal{A})$
$h_{(p,r)}$	Homothétie centrée en $p$ de rapport $r$	
$K$	Primitive d'affichage	$K \in \mathcal{H}(E)$
$(K_n)_{n \in \mathbb{N}}$	Suite approximant l'attracteur	$K_0 = K$ $K_{n+1} = \mathcal{T}(K_n)$ $\lim_{n \rightarrow \infty} K_n = \mathcal{A}$
$\Sigma^*$	Ensemble des mots finis sur $\Sigma$	
$\Sigma^\omega$	Ensemble des mots infinis sur $\Sigma$	
$T_\sigma$	Opérateur composé	$\sigma \in \Sigma^*$ $T_\sigma = T_{\sigma_1} T_{\sigma_2} \dots T_{\sigma_n}$
$\varphi$	Fonction d'adressage	$\varphi : E \leftarrow \Sigma^\omega$
$(\mathcal{Q}, \Sigma, \delta)$	Automate	
$\mathcal{Q}$	Ensemble d'états	
$\delta$	Fonction de transition	$\delta : \mathcal{Q} \leftarrow \mathcal{Q} \times \Sigma$
$\mathfrak{q}$	État initial	$\mathfrak{q} \in \mathcal{Q}$
$E^x$	Espace affine associé à l'état $x$	
$\Sigma^x$	Alphabet associé à l'état $x$	
$\Sigma_{\div}$	Symboles de subdivision	$\Sigma_{\div} = \{0, \dots, n\}$
$\Sigma_{\partial}$	Symboles d'incidence	$\Sigma_{\partial} = \{\partial_0, \dots, \partial_n\}$
$T_i^x$	Opérateur de subdivision associé à l'état $x$	$i \in \Sigma_{\div}^x$ $T_i : E^x \leftarrow E^{\delta(x,i)}$
$\Pi_i^x$	Opérateur d'incidence associé à l'état $x$	$\partial_i \in \Sigma_{\partial}^x$ $\Pi_i : E^x \leftarrow E^{\delta(x,\partial_i)}$
$S_u^x$	Opérateur de permutation associé à l'état $x$	$S_u^x \in \mathfrak{S}_{J^x}$
$\hat{T}_u^x$	Opérateur général associé à l'état $x$	$u \in \Sigma^x$ $T_i : E^x \leftarrow E^{\delta(x,u)}$

Notation	Signification	Propriété générale
$\mathcal{A}^x$	Attracteur associé à l'état $x$	$\mathcal{A}^x \in \mathcal{H}(E^x)$ $\mathcal{A}^x = \bigcup_{i \in \Sigma_+^x} T_i^x(\mathcal{A}^{\delta(x,i)})$
$K^x$	Primitive d'affichage associée à l'état $x$	$K \in \mathcal{H}(E^x)$
$(K_n^x)_{n \in \mathbb{N}}$	Suite approximant l'attracteur	$K_0^x = K^x$ $K_{n+1}^x = \bigcup_{i \in \Sigma_+^x} T_i^x(K_n^{\delta(x,i)})$ $\lim_{n \rightarrow \infty} K_n^x = \mathcal{A}^x$
$\mathcal{L}_*^x$	Langage des mots finis associé à l'état $x$	
$\mathcal{L}_\omega^x$	Langage des mots infinis associé à l'état $x$	
$T_\sigma^x$	Opérateur composé associé à l'état $x$	$\sigma \in \mathcal{L}_*^x$ $T_\sigma^x : E^x \leftarrow E^{\delta(x,\sigma)}$ $T_\sigma^x = T_{\sigma_1}^x \dots T_{\sigma_n}^x(\delta(x,\sigma_1 \dots \sigma_{n-1}))$
$\varphi^x$	Fonction d'adressage associée à l'état $x$	$\varphi^x : E^x \leftarrow \mathcal{L}_\omega^x T_\sigma^x : E^x \leftarrow E^{\delta(x,\sigma)}$ $\varphi^x(\sigma) = \lim_{n \rightarrow \infty} T^x(\sigma_1 \dots \sigma_n)p$
$\gamma_\sigma^x$	Chemin $\sigma$ partant de l'état $x$	$\sigma \in \mathcal{L}_*^x$
$\Upsilon^x$	Règles d'équivalences associées à l'état $x$	$\forall(\theta, \theta') \in \Upsilon^x \begin{cases} \delta(x, \theta) = \delta(x, \theta') \\ \hat{T}_\theta^x = \hat{T}_{\theta'}^x \end{cases}$
$\tilde{\Upsilon}^x$	Équations sur les opérateurs d'incidence	
$\hat{\Upsilon}^x$	Équations sur les opérateurs de subdivision	
$B(\mathbb{R}^n)$	Espace barycentrique à $n$ coordonnées	
$J^x$	Ensemble d'indices de l'espace associé à $x$	$E^x = B(\mathbb{R}^{J^x})$
$m^x$	Nombre d'indices associés à $x$	$m^x =  J^x $
$j^x$	Ensemble d'indices internes à $x$	
$\dot{m}^x$	Dimension interne associée à $x$	$\dot{m}^x =  j^x $
$(E, \vec{E})$	Espace affine	
$\widehat{E}$	Espace homogénéisé	
$\star$	Opérateur de somme centrée en $c$	$\star : \widehat{E} \times \widehat{E} \rightarrow \widehat{E}$ $G_1 \star G_2 = G_1 + G_2 - c$
$\Pi$	Projection centrée en 0 sur l'hyperplan $w = 1$	$\Pi(w, x, y, z)^T = \left(1, \frac{x}{w}, \frac{y}{w}, \frac{z}{w}\right)^T$

# Chapitre 1

## Introduction

### 1.1 Contexte

Notre recherche s’inscrit dans le cadre de la modélisation géométrique d’objets appliquée à la construction en bois et la création de formes architecturales nouvelles [Wei04] [Wei06] [Wei08] [Wei09]. L’outil informatique et plus particulièrement les développements de la CAO (Conception Assistée par Ordinateur) ont apporté de nouveaux procédés de conception de formes complexes qui ont contribué à l’apparition de nouvelles formes architecturales.

Les modèles de CAO qui permettent de créer des formes complexes utilisent une approche classique de la géométrie avec des formes toujours localement planes ou lisses. Le projet de recherche “géométrie fractale et ses applications dans la construction en bois” à l’IBOIS a pour but de développer des applications concrètes de la géométrie fractale dans le domaine de la construction. C’est un projet interdisciplinaire qui regroupe les domaines de l’informatique et de la construction. Ivo STOTZ, doctorant en architecture, a été conjointement en charge de ce projet de recherche, en étudiant et développant des procédés liés à la construction. Le travail présenté ici porte sur le développement d’un modèleur basé sur un modèle fractal, permettant la conception de formes complexes appliquées à la construction.

### 1.2 Problématique

La plupart des modèleurs actuels utilisés en CAO (Conception Assistée par Ordinateur) sont basés sur une géométrie classique. Ils permettent de modéliser des courbes ou des surfaces analytiques, qui sont généralement polynomiales ou rationnelles, et d’aspect lisse. Nous proposons d’élaborer un nouveau type de modèleur, basé sur la *géométrie fractale*. Rappelons que Benoît MANDELBROT l’a introduit pour prendre toute une famille de formes considérées comme irrégulières du point de vue classique, en particulier des formes naturelles (arbres, côtes du littoral, montagnes, nuages, ...) [Man75] [Man83] [PR86] [LM90] [PLH88] [Smi84] [PH90].

Nous utilisons un formalisme basé sur le modèle IFS (Iterated Function System) [Hut81] [BD85] [DHN85] [Gen92] [Tos96a], qui permet de modéliser une certaine gamme de formes autosimilaires. Un IFS est défini par un ensemble d’opérateurs géométriques qui décrivent l’autosimilarité de la forme modélisée : cette forme est égale à l’union des copies d’elle-même par l’ensemble de ces opérateurs géométriques.

Le modèle IFS est un modèle itératif. La description d’un IFS est assimilable à la description d’un processus de calcul itératif permettant de construire la figure modélisée. Dans ce processus de calcul, on part d’une figure de base donnée, que l’on subdivise en un ensemble de sous-figures en appliquant l’ensemble des opérateurs géométriques de l’IFS.

En réitérant ce processus, on obtient une suite de figures de plus en plus complexes et détaillées, et qui converge vers une certaine limite appelée “attracteur”. Cet attracteur est indépendant de la figure de base choisie, mais ne dépend que des opérateurs géométriques.

Les formes modélisées étant destinées à la construction, il est avantageux d’intégrer les contraintes de fabrication le plus en amont possible, c’est-à-dire au niveau de la création de la forme de l’objet. Cela aurait plusieurs avantages :

- Raccourcir la chaîne de traitement et les délais de fabrication d’un produit.
- Produire plus facilement et plus rapidement des objets uniques ou des gammes d’objets.
- Favoriser la créativité du concepteur en lui offrant un outil de travail très ouvert, tout en le libérant des contraintes de fabrication.
- Mieux exploiter cette créativité : les formes produites avec le logiciel seront proches des formes fabriquées. Il y aura moins de pertes et de dégradation d’information.

Ces avantages ne peuvent qu’améliorer :

- la productivité : la chaîne conception-simulation-fabrication est plus directe ;
- la créativité : cette modélisation offre des possibilités, non offertes par les modeleurs actuels.

Le modeleur doit également permettre une facilité d’utilisation, en permettant de contrôler de manière rapide et intuitive les objets modélisés.

## 1.3 État de l’art

Les procédés itératifs pour produire des images ou des figures sont nombreux et divers. Parmi ceux qui possèdent des propriétés utiles à la modélisation géométrique, on peut distinguer deux principales familles : la formulation IFS basée sur les matrices de subdivision, et la formulation par schéma de subdivision basée sur les points de subdivision.

### 1.3.1 IFS et généralisations

Pour obtenir des figures non autosimilaires, le modèle IFS a été généralisé par plusieurs auteurs avec des formulations différentes (règles de production, systèmes d’équations, graphes, langages, . . .). mais qui génèrent la même famille de figures. La formulation la plus intéressante est celle de PRUSINKIEWICZ, qui a introduit un système de règles exprimé par un automate, sous l’appellation LRIFS (Langage Restricted Iterated Function System) ou CIFS (Controlled Iterated Function System). [PH91]. Ce type de procédé a également été développé par CULIK (MRIFS) et BARNSLEY (RIFS) [CD92] [CD93].

Sous certaines contraintes, ces modèles décrivent des courbes — FIF (Fractal Interpolation Function) de BARNSLEY [Bar88] — ou des figures filaires — les L-systèmes de PRUSINKIEWICZ décrivent un CIFS arborescent [PH94].

Les modèles IFS et CIFS seront présentés plus en détails dans les deux chapitres suivants.

### 1.3.2 Schémas de subdivision

On peut ranger dans l’appellation schémas de subdivision, une famille très utilisée, basée sur les grilles de contrôle et des masques de moyennage. Le procédé itératif consiste à raffiner des maillages. A chaque étape sont ajoutés de nouveaux points calculés par moyennage des points précédents.

Les schémas de subdivision déterministes sont utilisés pour générer des courbes, des surfaces ou encore des bases d’ondelette. Ce sont en général des schémas stationnaires et uniformes. Les coefficients des masques utilisés en CAO (CHAINKIN, LOOP, . . .) sont choisis

de manière à générer des arcs et carreaux polynomiaux, afin de décrire des formes lisses. Les coefficients des masques utilisés dans les ondelettes peuvent conduire à des fonctions fractales [DL92].

On peut également citer les schémas de subdivision non déterministes utilisés en synthèse d'images pour simuler des formes naturelles (montagnes, rochers, paysages de Kenton MUSGRAVE). Les coefficients des masques sont des variables aléatoires. Mais ces procédés ne génèrent pas une forme paramétrée bien définie. C'est suffisant si on ne s'intéresse qu'à un simulacre visuel, mais insuffisant pour représenter un objet bien précis, et constituer un vrai modèle géométrique.

### 1.3.3 Modélisation fractale

L'utilisation de ces procédés itératifs est relativement marginale [BJM\*88] [LV88] [GV90] [GTV91]. Ils sont utilisés pour générer des images sans entrer dans une problématique de modélisation géométrique avec une exploitation complète. D'autres sont exploités en synthèses d'images pour générer des montagnes, structures arborescentes ou encore des formes nuageuses, et sont associés à des générateurs aléatoires pour donner un aspect "plus naturel" aux résultats. Ce qui prime dans ce cas est le "rendu" naturel au détriment du contrôle des formes. De plus, la plupart de ces méthodes sont très spécifiques au type d'objet que l'on souhaite générer. Il s'agit le plus souvent de systèmes de génération de structures qui sont par la suite stockées et exploitées sous une représentation classique (B-Rep, CSG, ...). Ainsi leur manipulation et modification ne sont plus liées au procédé de génération, ce qui rend difficile leur exploitation.

Dans ses travaux précédents, le LIRIS a étudié et développé la modélisation géométrique fractale comme une extension de la modélisation géométrique classique aux formes fractales. En effet, il est possible de manipuler les formes fractales avec les mêmes facilités que les formes classiques (formes à pôles, déformations, produits tensoriels) [ZT96, TZTV97]. D'autre part, en introduisant certaines contraintes sur les matrices, il est possible de choisir la structure topologique des formes produites : figures filaires [Tos99b] ou surfaciques [TRM99, TGB02]. L'algorithme de visualisation produisant des images devient ainsi un algorithme générant des suites de figures bien structurées (des maillages faces-arêtes-sommets) [Gen92] [Tho96] [Tos96b] [Zai98] [Gué02].

Parallèlement à ceci, le LE2I a mené plusieurs travaux de recherche sur la modélisation géométrique de formes fractales. Ils ont notamment étudié les courbes et les surfaces avec une texture fractale variable [GTN06]. Par ailleurs, il est possible d'étendre la notion de dérivée pour l'appliquer aux formes fractales [BGN07] [BGN08].

## 1.4 Objectifs

Nous étudions et développons la modélisation géométrique fractale comme une extension de la modélisation géométrique classique aux formes fractales.

Nous partons du modèle IFS, qui permet d'accéder à une gamme extrêmement variée de formes tout en étant simple à utiliser.

Nous enrichissons ce modèle, pour l'intégrer dans un modéleur utilisé pour la conception de formes appliquées à la construction. Les formes produites doivent donc vérifier des critères de constructibilité.

### 1.4.1 Contraintes topologiques

Les formes des objets virtuels ont peu de contraintes, à part celle d'être localement approximées par des primitives graphiques (elles peuvent être non connexes, constituer des

nuages, flotter dans l'espace, ...). Ce n'est pas le cas des formes des objets physiques qui doivent être matérialisables, c'est-à-dire réalisables dans un dispositif de fabrication donné. Les formes générées doivent être munies d'une structure topologique bien définie, filaire, surfacique ou volumique. Elles doivent avoir une bonne traduction dans un espace physique : elles doivent pouvoir être épaissies pour passer de figures filaires ou surfaciques à des objets en volume.

### 1.4.2 Contrôle géométrique

Les modeleurs classiques permettent la déformation globale, par l'intermédiaire des points de contrôle, mais ne permettent pas de modifier les paramètres de subdivision. Ils utilisent des masques de subdivision prédéfinis, de manière à générer des formes lisses.

Nous devons établir des procédés permettant à l'utilisateur de contrôler les paramètres de subdivision, spécifiques aux modèles itératifs. Le contrôle de ces paramètres doit se faire de la manière la plus simple et la plus intuitive possible.

### 1.4.3 Contraintes de fabrication

Certaines contraintes sur les formes géométriques, spécifiques aux applications et aux procédés de fabrication doivent être prises en compte.

Dans le domaine de l'architecture nous avons identifié plusieurs contraintes. Les maillages produits par un modeleur surfacique ont généralement des faces triangulaires, ou des faces quadrangulaires gauches. La modélisation géométrique s'est intéressée récemment à la production de maillages à faces quadrangulaires planes, pour les appliquer à la construction architecturale [HPW07] [GSC\*04].

### 1.4.4 Modèle fractal

Plusieurs auteurs ont souligné l'analogie entre IFS et schémas de subdivision [SLG04]. Mais celle-ci a été effectuée cas par cas, comme DAUBECHIES pour les courbes [DL92]. Un attracteur est généralement une forme fractale, mais il peut également être lisse dans certains cas particuliers. Ces cas particuliers correspondent aux formes que l'on obtient généralement avec les modeleurs utilisant les schémas de subdivision classiques. Le modeleur développé doit permettre à un concepteur de modéliser un ensemble de formes beaucoup plus vaste que les modeleurs actuels, tout en intégrant les possibilités de la modélisation classique. Il est nécessaire pour cela de définir un formalisme qui unifie le modèle IFS avec celui des schémas de subdivision.

Nous introduisons un nouveau modèle, BCIFS (Boundary Controlled Iteratde Function System), qui est une généralisation du modèle IFS et du modèle CIFS de PRUSINKIEWICZ. Dans cette thèse, nous étudions les propriétés de ce modèle et nous montrons comment il peut répondre à notre problématique :

- sa formulation permet de contrôler explicitement la topologie de l'objet modélisé,
- il permet d'englober les schémas de subdivision classiques et une grande part des modèles itératifs existants,
- nous utilisons ce modèle comme base de notre modeleur et montrons comment il peut être contrôlé par l'utilisateur.



## Chapitre 2

# Modèles itératifs basée sur les IFS : rappels

### 2.1 Le modèle IFS

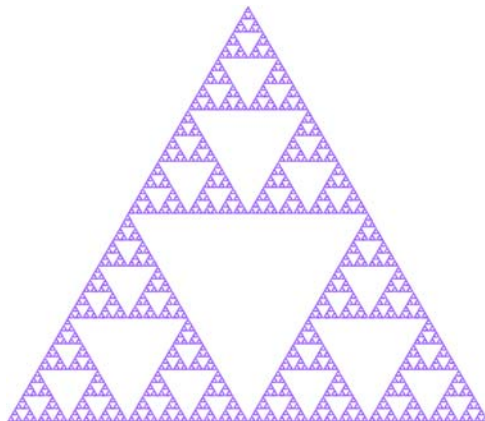
#### 2.1.1 Définition ensembliste

Un IFS est défini par un ensemble  $\{T_i\}_{i \in \Sigma}$  d'opérateurs appartenant à un semigroupe  $\mathcal{S}$  opérant sur  $E$ . En pratique,  $E$  est un espace affine, et  $\mathcal{S}$  est le semigroupe des opérateurs affines contractants. Ces opérateurs définissent un attracteur  $\mathcal{A}$  de manière récursive, et qui est un compact non vide de  $E$ .

$$\mathcal{A} = \bigcup_{i \in \Sigma} T_i(\mathcal{A})$$

#### 2.1.2 Exemples d'IFS

Dans les exemples qui suivent, on se place dans  $(E, d)$  le plan euclidien muni de la distance euclidienne.



**Fig. 2.1** – *Triangle de SIERPINSKI*

Le premier exemple que nous donnons est celui du triangle de SIERPINSKI, illustré sur la figure 2.1. On définit l'IFS par trois opérateurs  $T_0$ ,  $T_1$  et  $T_2$ . Chacun de ces opérateurs

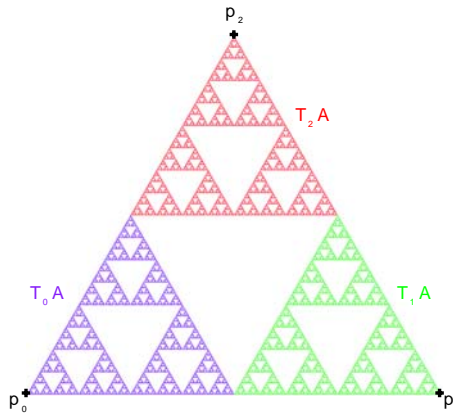
$T_i$  est défini comme une homothétie de rapport  $\frac{1}{2}$  centrée en un point  $p_i$  du plan.

$$\begin{aligned} T_0 &= h_{(p_0; \frac{1}{2})} \\ T_1 &= h_{(p_1; \frac{1}{2})} \\ T_2 &= h_{(p_2; \frac{1}{2})} \end{aligned}$$

Le triangle de SIERPINSKI est l'attracteur  $\mathcal{A}$  de cet IFS, qui est un ensemble compact non vide vérifiant :

$$\mathcal{A} = \bigcup_{i \in \Sigma} T_i(\mathcal{A}) = h_{(p_0; \frac{1}{2})}\mathcal{A} \cup h_{(p_1; \frac{1}{2})}\mathcal{A} \cup h_{(p_2; \frac{1}{2})}\mathcal{A}$$

Nous montrerons par la suite que l'attracteur  $\mathcal{A}$  est entièrement déterminé par la donnée de l'ensemble  $\{T_i\}_{i \in \Sigma}$  des opérateurs définissant l'IFS, car il existe un unique compact non vide vérifiant l'équation ci-dessus.



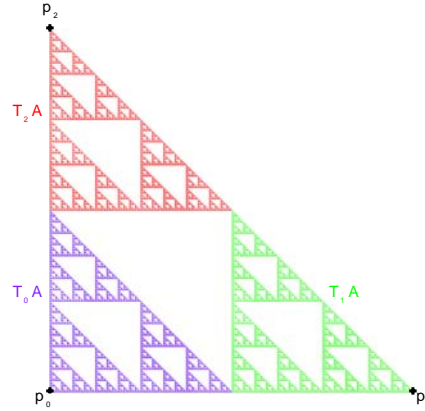
**Fig. 2.2** – Représentation du triangle de SIERPINSKI comme l'union de trois copies de lui-même par trois homothéties de rapport 0.5

La figure 2.2 représente dans trois couleurs différentes l'image de l'attracteur par chacune des homothéties définissant l'IFS, où les centres d'homothétie  $p_0$ ,  $p_1$  et  $p_2$  correspondent aux sommets du triangle principal. On constate visuellement la propriété fondamentale des attracteurs d'IFS  $\mathcal{A} = \bigcup_{i \in \Sigma} T_i(\mathcal{A})$  où chaque opérateur  $T_i$  est une homothétie de rapport  $\frac{1}{2}$  centrées en  $p_i$ , un des trois sommets du triangle principal.

Le triangle de SIERPINSKI possède un certain nombre de particularités, que l'on retrouve généralement pour les attracteurs d'autres IFS. L'une des caractéristiques principales, commune aux attracteurs d'IFS, est celle de l'autosimilarité par changement d'échelle. Cela signifie que l'on trouve à l'intérieur de l'attracteur certains sous-ensembles similaires à l'ensemble entier, et ceci se produit à des échelles aussi petites que l'on veut.

Un IFS est la donnée d'un ensemble d'opérateurs  $\{T_i\}_{i \in \Sigma}$ , et définit un certain attracteur  $\mathcal{A}$ . En choisissant un IFS différent, on obtient un attracteur différent. Nous montrons, sur des exemples, quelques attracteurs que l'on peut obtenir en faisant varier les IFS.

La figure 2.3 est une légère variante géométrique du triangle de SIERPINSKI "standard". Ici, l'IFS est toujours défini par trois homothéties de rapport  $\frac{1}{2}$ , mais on a simplement modifié le centre d'homothétie d'un des opérateurs.

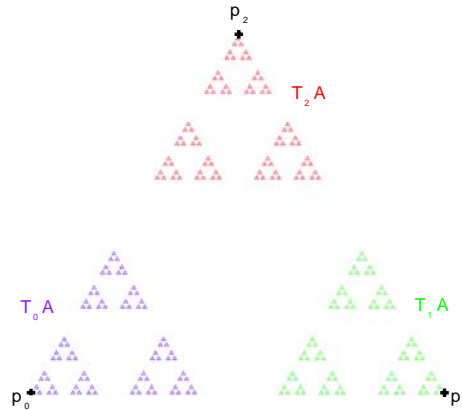


**Fig. 2.3** – Attracteur d'un IFS composé de trois homothéties de rapport 0.5 et dont les centres d'homothétie forment un triangle isocèle rectangle

Sur les prochains exemples, on définit des IFS toujours composés de trois homothéties, dont les centres forment un triangle équilatéral, mais en faisant varier uniquement les rapports d'homothéties.

Dans l'exemple suivant, on définit un IFS par trois homothéties de rapport 0.4 (au lieu de 0.5 pour le triangle de SIERPINSKI). L'attracteur de cet IFS est montré sur la figure 2.4.

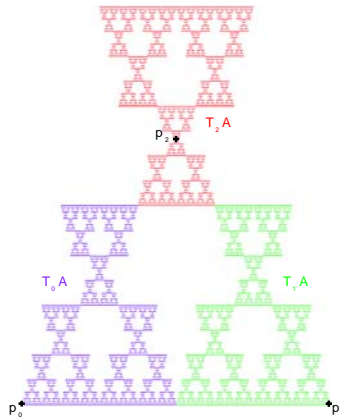
$$\begin{aligned} T_0 &= h_{(p_0;0.4)} \\ T_1 &= h_{(p_1;0.4)} \\ T_2 &= h_{(p_2;0.4)} \end{aligned}$$



**Fig. 2.4** – Attracteur d'un IFS composé de trois homothéties de rapport 0.4

On peut définir des homothéties de rapports différents dans le même IFS. Dans l'exemple suivant, on définit un IFS par deux homothéties de rapport 0.5 et une homothétie de rapport  $-0.5$ . L'attracteur de cet IFS est montré sur la figure 2.5.

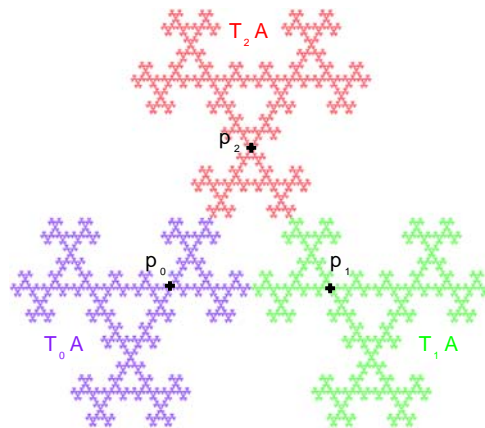
$$\begin{aligned} T_0 &= h_{(p_0;0.5)} \\ T_1 &= h_{(p_1;0.5)} \\ T_2 &= h_{(p_2;-0.5)} \end{aligned}$$



**Fig. 2.5** – Attracteur d'un IFS composé de deux homothéties de rapport 0.5 et d'une homothétie de rapport  $-0.5$

Enfin, la figure 2.6 montre l'attracteur d'un IFS composé de trois homothéties de rapport  $-0.5$ .

$$\begin{aligned} T_0 &= h_{(p_0; -0.5)} \\ T_1 &= h_{(p_1; -0.5)} \\ T_2 &= h_{(p_2; -0.5)} \end{aligned}$$



**Fig. 2.6** – Attracteur d'un IFS composé de trois homothéties de rapport  $-0.5$

Sur les exemples que nous venons de donner, on a pu observer graphiquement les attracteurs correspondant à certains types d'IFS, dont on a pu constater la propriété fondamentale selon laquelle l'attracteur d'un IFS est une union de copies de lui-même par un ensemble de transformations.

### Application aux IFS du théorème du point fixe

Étant donné un IFS  $\{T_i\}_{i \in \Sigma}$  défini sur  $(E, d)$ , on définit l'opérateur de HUTCHINSON  $\mathcal{T}$  sur  $\mathcal{H}(E)$  l'ensemble des compacts non vides de  $E$ .

$$\begin{aligned} \mathcal{T} : \mathcal{H}(E) &\rightarrow \mathcal{H}(E) \\ \mathcal{T}(K) &= \bigcup_{i \in \Sigma} T_i(K) \end{aligned}$$

On munit  $\mathcal{H}(E)$  de la distance de HAUSDORFF  $d_H$ .

$$d_H(K, K') = \max\left\{\sup_{k \in K} \inf_{\ell \in K'} d(k, \ell), \sup_{\ell \in K'} \inf_{k \in K} d(k, \ell)\right\}$$

L'espace métrique  $(E, d)$  étant complet, alors  $(\mathcal{H}(E), d_H)$  l'ensemble des compacts non vides de  $E$  muni de  $d_H$  est lui-même complet. De plus, les opérateurs  $\{T_i\}_{i \in \Sigma}$  étant contractants sur  $(E, d)$ , alors  $\mathcal{T}$  l'opérateur de HUTCHINSON est lui-même contractant sur  $(\mathcal{H}(E), d_H)$ . Les conditions d'application du théorème du point fixe sont réunies. L'attracteur  $\mathcal{A}$  d'un IFS correspond au point fixe de  $\mathcal{T}$  dans  $(\mathcal{H}(E), d_H)$ . Il est l'unique compact non vide tel que

$$\mathcal{A} = \mathcal{T}(\mathcal{A}) = \bigcup_{i \in \Sigma} T_i(\mathcal{A})$$

### 2.1.3 Algorithme itératif pour le calcul d'un attracteur

Le théorème du point fixe nous indique que le point fixe de  $\mathcal{T}$  est égal à la limite de la suite  $K_{n+1} = \mathcal{T}(K_n)$  avec  $K_0$  quelconque.

Ceci nous fournit un procédé itératif permettant d'approximer l'attracteur  $\mathcal{A}$  d'un IFS donné. On définit pour cela  $K$  un compact non vide quelconque, et qui est le terme initial de la suite  $(K_n)_{n \in \mathbb{N}}$  définie par :

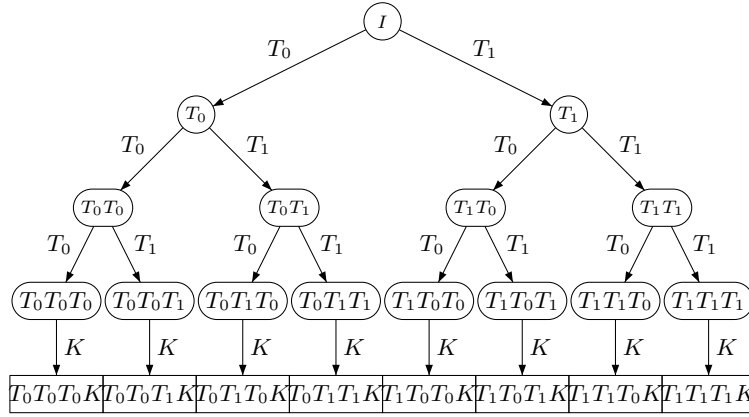
$$\begin{aligned} K_0 &= K \in \mathcal{H}(E) \\ K_{n+1} &= \mathcal{T}(K_n) = \bigcup_{\sigma \in \Sigma} T_\sigma K_n \end{aligned}$$

L'opérateur  $\mathcal{T}$  étant l'union des opérateurs  $T_i$  de l'IFS avec  $\mathcal{T}(K) = \bigcup_{i \in \Sigma} T_i(K)$ , chaque terme  $K_n$  est composé d'un ensemble de  $|\Sigma|^n$  images de  $K$  par une fonction composée  $n$  fois de fonctions  $T_i$ . Par exemple, voici l'expression développée des termes de  $K$  pour un IFS à deux transformations, avec  $\Sigma = \{0; 1\}$ .

$$\begin{aligned} K_1 &= \mathcal{T}(K_0) = T_0 K \cup T_1 K \\ K_2 &= \mathcal{T}(K_1) = T_0 T_0 K \cup T_0 T_1 K \cup T_1 T_0 K \cup T_1 T_1 K \\ K_3 &= \mathcal{T}(K_2) = T_0 T_0 T_0 K \cup T_0 T_0 T_1 K \cup T_0 T_1 T_0 K \cup T_0 T_1 T_1 K \cup \\ &\quad T_1 T_0 T_0 K \cup T_1 T_0 T_1 K \cup T_1 T_1 T_0 K \cup T_1 T_1 T_1 K \\ &\quad \vdots \quad \quad \quad \vdots \\ K_n &= \mathcal{T}(K_{n-1}) = \bigcup_{\sigma \in \Sigma^n} T_{\sigma_1} T_{\sigma_2} \dots T_{\sigma_n} K \end{aligned}$$

Un algorithme d'affichage basé sur ce principe a été défini pour la première fois par BARNSELY. Dans cet algorithme, l'ensemble des duplicata de  $K$  est construit de manière arborescente. Chaque nœud de l'arbre de profondeur  $d$  correspond au calcul d'une composée de  $d$  fonctions  $T_i$ . Cet arbre est parcouru jusqu'à une profondeur  $n$  donnée, où on affiche l'image de  $K$  par la fonction composée associée au nœud courant, comme le montre la figure 2.7. Dans la suite de ce document, on appellera  $K$  la "primitive d'affichage" associée à l'IFS.

Un point important à noter est que les composées de fonctions sont calculées de gauche à droite. La primitive d'affichage  $K$  est, dans un dernier temps, passée en paramètre à chacune de ces composées. Dans la pratique, les opérateurs  $T_i$  utilisés sont des opérateurs affines. Ces opérateurs sont représentés par des matrices. Cela rend possible le calcul de gauche à droite des fonctions composées, car la composition de fonctions représentées par des matrices revient à faire un produit matriciel.



**Fig. 2.7** – Arbre de calcul de  $K_3$ . Les nœuds internes correspondent au calcul d'une fonction composée. Les feuilles correspondent à une sous-figure de  $K_3$  à afficher

La figure 2.8 montre les étapes successives de la suite de figures  $(K_n)_{n \in \mathbb{N}}$  pour le triangle de SIERPINSKI. La primitive d'affichage choisie est un triangle dont les sommets sont les centres de chaque homothétie.

La suite de figures  $(K_n)_n$  dépend de la primitive d'affichage  $K$  choisie. Cependant,  $(K_n)_n$  tend vers l'unique attracteur  $\mathcal{A}$  pour tout compact non vide  $K \in \mathcal{H}(E)$ .

### 2.1.4 Fonction d'adressage

On utilise l'indexation sur  $\Sigma$  des opérateurs  $(T_i)_{i \in \Sigma}$ . À tout mot fini  $\sigma \in \Sigma^*$ , on associe le produit :

$$\begin{aligned} \Sigma^* &= \mathcal{S} \\ \mathcal{T}_\epsilon &= Id \\ \mathcal{T}_\sigma &= T_{\sigma_1} \circ T_{\sigma_2} \circ \dots \circ T_{\sigma_n} \end{aligned}$$

Chaque composante de  $K_n$  correspond à  $T_\sigma K$  avec  $\sigma$  un mot de longueur  $n$ .

$$K_n = \bigcup_{\sigma \in \Sigma^n} T_\sigma K$$

On définit une fonction d'adressage  $\varphi$  sur les mots infinis  $\Sigma^\omega$ .

$$\begin{aligned} \varphi : \Sigma^\omega &\rightarrow E \\ \varphi(\sigma) &= \lim_{n \rightarrow \infty} T_{\sigma_1} T_{\sigma_2} \dots T_{\sigma_n} p \end{aligned}$$

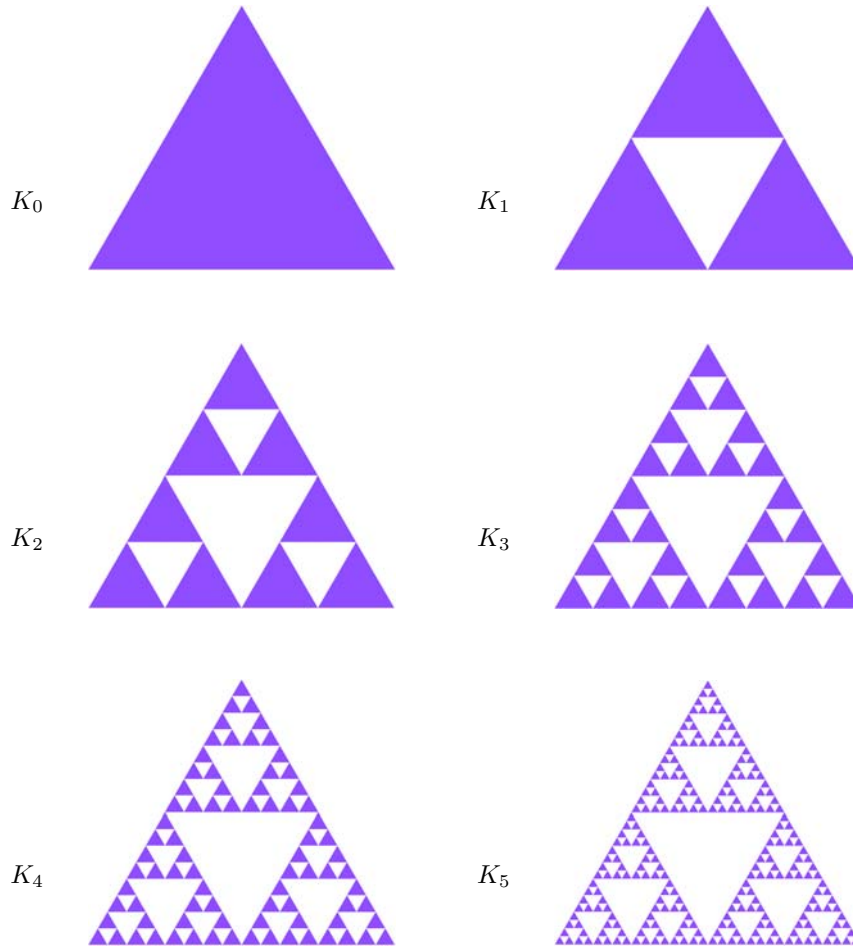
Le point obtenu, indépendant du paramètre  $p \in \mathcal{E}$ , appartient à l'attracteur  $\mathcal{A}$ . Réciproquement, tout point de l'attracteur est l'image de  $\varphi(\sigma)$  avec  $\sigma \in \Sigma^\omega$ . La fonction d'adressage permet de représenter l'attracteur sous forme paramétrée sur l'ensemble des mots infinis.

$$\mathcal{A} = \varphi(\Sigma^\omega)$$

## 2.2 Le modèle CIFS (Controlled Iterated Function System)

### 2.2.1 Données du modèle

Un CIFS est donné par un automate, dont chaque état  $x$  est associé à un attracteur  $\mathcal{A}^x$  dans un espace  $E^x$ , et les transitions de  $x$  vers  $y$  sont associées à un opérateur de  $E^y \rightarrow E^x$ .



**Fig. 2.8** – Construction itérative d'un triangle de SIERPINSKI

On donne ici la liste des paramètres permettant de décrire un CIFS :

- un automate  $(\mathcal{Q}, \Sigma, \delta)$  où  $\Sigma$  est un alphabet,  $\mathcal{Q}$  est un ensemble d'états et  $\delta$  est une fonction de transition  $\delta : \mathcal{Q} \times \Sigma \rightarrow \mathcal{Q}$ ,
- un ensemble d'espaces associés aux états :  $(E^x)_{x \in \mathcal{Q}}$ ,
- un ensemble d'opérateurs associés aux transitions :  $T_i^x : E^{\delta(x,i)} \rightarrow E^x$  que l'on notera plutôt  $T_i^x : E^x \leftarrow E^{\delta(x,i)}$ ,
- un ensemble de primitives d'affichage associées aux états :  $(K^x)_{x \in \mathcal{Q}}$  où  $K^x \in \mathcal{H}(E^x)$ , ne servant pas à définir les attracteurs, mais uniquement les suites approximantes,
- par la suite, l'automate sera éventuellement muni d'un état initial noté  $\natural$ .

On notera également  $\Sigma_x$  la restriction de  $\Sigma$  aux transitions sortantes associées à l'état  $x$ .

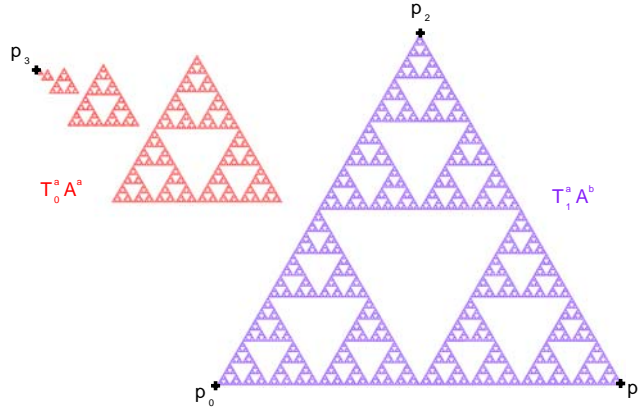
$$\Sigma_x = \{i \in \Sigma, \delta(x, i) \in \mathcal{Q}\}$$

## 2.2.2 Définition ensembliste des attracteurs

Un CIFS définit une famille d'attracteurs associés aux états :  $(\mathcal{A}^x)_{x \in \mathcal{Q}}$  où  $\mathcal{A}^x \in \mathcal{H}(E^x)$ . Nous avons vu qu'un IFS définit un attracteur de manière récursive. Les attracteurs d'un

CIFS sont définis de manière mutuellement récursive.

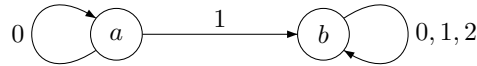
$$\mathcal{A}^x = \bigcup_{i \in \Sigma_x} T_i^x \mathcal{A}^{\delta(x,i)}$$



**Fig. 2.9** – Attracteur  $\mathcal{A}^a$ , représenté comme l'union d'une copie de lui-même par homothétie (en rouge), et d'un triangle de SIERPINSKI  $\mathcal{A}^b$  (en violet)

Un exemple d'attracteur de CIFS est illustré sur la figure 2.9. Le CIFS correspondant est décrit par un automate à deux états  $a$  et  $b$ . On attribue deux opérateurs de subdivision à  $a$ , l'un bouclant sur  $a$ , l'autre pointant sur  $b$ . On attribue à  $b$  trois opérateurs de subdivision bouclant sur  $b$ . La figure 2.10 représente l'automate de ce CIFS, dont la fonction de transition est décrite par la liste suivante.

$$\begin{aligned} \delta(a, 0) &= a & \delta(b, 0) &= b \\ \delta(a, 1) &= b & \delta(b, 1) &= b \\ & & \delta(b, 2) &= b \end{aligned}$$



**Fig. 2.10** – Automate décrivant un CIFS

À chaque transition  $\delta(x, i) = y$  de l'automate est associé un opérateur  $T_i^x : E^x \leftarrow E^y$ . On suppose que  $E^a$  et  $E^b$  sont tous les deux le même plan affine euclidien. Les opérateurs  $T_i^x$  associés aux transitions sont des applications de ce plan dans lui-même, que l'on définit comme des homothéties centrées en certains points du plan.

$$\begin{aligned} T_0^a &= h_{(p_3; 0.42)} & T_0^b &= h_{(p_0; 0.5)} \\ T_1^a &= Id & T_1^b &= h_{(p_1; 0.5)} \\ & & T_2^b &= h_{(p_2; 0.5)} \end{aligned}$$



Les attracteurs  $\mathcal{A}^a$  et  $\mathcal{A}^b$  vérifient alors le système suivant :

$$\begin{aligned}
\mathcal{A}^a &= \bigcup_{i \in \Sigma_a} T_i^a \mathcal{A}^{\delta(a,i)} \\
&= T_0^a \mathcal{A}^a \cup T_1^a \mathcal{A}^b \\
&= h_{(p_3;0.42)} \mathcal{A}^a \cup \mathcal{A}^b \\
\mathcal{A}^b &= \bigcup_{i \in \Sigma_b} T_i^b \mathcal{A}^{\delta(b,i)} \\
&= T_0^b \mathcal{A}^b \cup T_1^b \mathcal{A}^b \cup T_2^b \mathcal{A}^b \\
&= h_{(p_0;0.5)} \mathcal{A}^b \cup h_{(p_1;0.5)} \mathcal{A}^b \cup h_{(p_2;0.5)} \mathcal{A}^b
\end{aligned}$$

On constate d'après la définition de  $\mathcal{A}^b$  que cet attracteur correspond à un triangle de SIERPINSKI dont les sommets sont  $p_0, p_1$  et  $p_2$ . D'une manière générale, dans un CIFS, tout état dont les transitions sortantes sont des boucles correspond à un IFS.

L'attracteur  $\mathcal{A}^a$  est l'union d'une copie de lui-même par homothétie et d'un triangle de SIERPINSKI. Sur la figure 2.9, on constate que cet attracteur correspond à un ensemble infini de copies de triangles de SIERPINSKI par homothéties successives. Cette propriété peut se déduire des équations que l'on a définies.

$$\begin{aligned}
\mathcal{A}^a &= \mathcal{A}^b \cup T_0^a \mathcal{A}^a \\
&= \mathcal{A}^b \cup T_0^a \mathcal{A}^b \cup T_0^a T_0^a \mathcal{A}^a \\
&= \mathcal{A}^b \cup T_0^a \mathcal{A}^b \cup T_0^a T_0^a \mathcal{A}^b \cup T_0^a T_0^a T_0^a \mathcal{A}^a \\
&= \dots \\
&= \bigcup_{n \in \mathbb{N}} (T_0^a)^n \mathcal{A}^b \cup \lim_{m \rightarrow \infty} (T_0^a)^m \mathcal{A}^a \\
&= \bigcup_{n \in \mathbb{N}} h_{(p_3;(0.42)^n)} \mathcal{A}^b \cup \{p_3\}
\end{aligned}$$

On précise que l'ensemble  $\mathcal{A}^a \setminus \{p_3\}$  vérifierait aussi l'équation du CIFS  $\mathcal{A}^a \setminus \{p_3\} = \mathcal{A}^b \cup T_0^a (\mathcal{A}^a \setminus \{p_3\})$  mais ne serait pas compact, ce qui est contraire à la définition d'un attracteur.

Par ailleurs, comme dans le cas des IFS, l'existence et l'unicité des attracteurs dépend de certains critères de contraction des opérateurs. Les attracteurs d'un CIFS sont définis de manière unique si pour chaque circuit élémentaire de l'automate, l'opérateur associé est contractant. Dans notre exemple, les seuls circuits élémentaires sont des boucles.

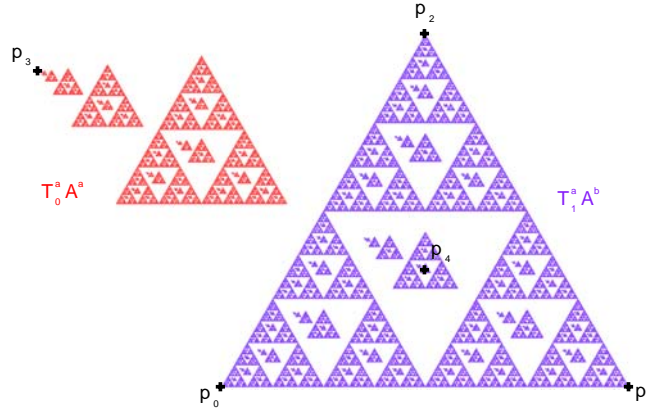
$$\begin{aligned}
\delta(a, 0) &= a & \delta(b, 0) &= b \\
&& \delta(b, 1) &= b \\
&& \delta(b, 2) &= b
\end{aligned}$$

À chaque circuit est associé un opérateur.

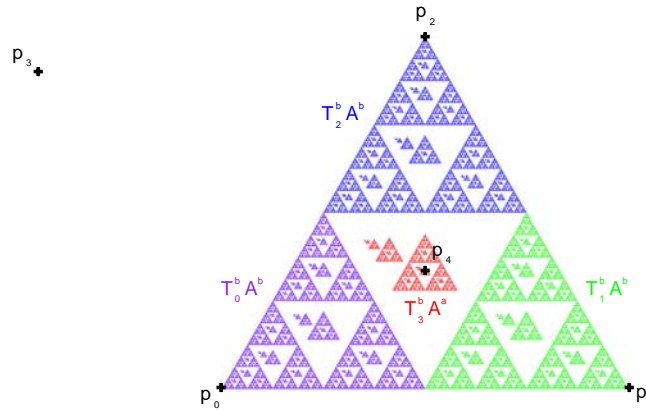
$$\begin{aligned}
T_0^a &= h_{(p_3;0.42)} & T_0^b &= h_{(p_0;0.5)} \\
&& T_1^b &= h_{(p_1;0.5)} \\
&& T_2^b &= h_{(p_2;0.5)}
\end{aligned}$$

Les opérateurs  $T_0^a, T_0^b, T_1^b$  et  $T_2^b$  associés à ces circuits sont contractants. Il existe donc un unique compact non vide  $\mathcal{A}^a$  et un unique compact non vide  $\mathcal{A}^b$  vérifiant le système  $\mathcal{A}^x = \bigcup_{i \in \Sigma_x} T_i^x \mathcal{A}^{\delta(x,i)}$ .

Un autre exemple CIFS est illustré sur les figures 2.11 et 2.12. Cet exemple est assez proche de l'exemple précédent. On y a apporté comme seule modification l'ajout d'une



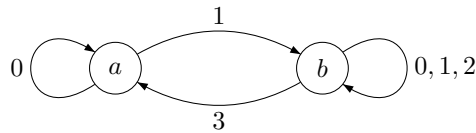
**Fig. 2.11** – Attracteur  $\mathcal{A}^a$ , représenté comme l'union d'une copie de lui-même par homothétie (en rouge), et d'une copie de  $\mathcal{A}^b$  (en violet)



**Fig. 2.12** – Attracteur  $\mathcal{A}^b$ , représenté comme l'union de trois copies de lui-même et d'une copie de  $\mathcal{A}^a$  (en rouge)

transition allant de  $b$  vers  $a$ . La figure 2.13 représente l'automate de ce CIFS, dont la fonction de transition est décrite par la liste suivante.

$$\begin{aligned}
 \delta(a, 0) &= a & \delta(b, 0) &= b \\
 \delta(a, 1) &= b & \delta(b, 1) &= b \\
 & & \delta(b, 2) &= b \\
 & & \delta(b, 3) &= a
 \end{aligned}$$



**Fig. 2.13** – Exemple d'automate décrivant un CIFS

La nouvelle transition  $\delta(b, 3) = a$  de l'automate est associé un opérateur  $T_3^b : E^b \leftarrow E^a$

définie comme une homothétie centrée en un point  $p_4$  du plan, barycentre des points  $p_0, p_1$  et  $p_2$ .

$$\begin{array}{ll} T_0^a & = h_{(p_3;0.42)} \\ T_1^a & = Id \\ T_0^b & = h_{(p_0;0.5)} \\ T_1^b & = h_{(p_1;0.5)} \\ T_2^b & = h_{(p_2;0.5)} \\ T_3^b & = h_{(p_4;0.16)} \end{array}$$

La donnée de l'automate et des opérateurs associés définissent les attracteurs  $\mathcal{A}^a$  et  $\mathcal{A}^b$ , vérifiant le système suivant :

$$\begin{aligned} \mathcal{A}^a &= \bigcup_{i \in \Sigma_a} T_i^a \mathcal{A}^{\delta(a,i)} \\ &= T_0^a \mathcal{A}^a \cup T_1^a \mathcal{A}^b \\ &= h_{(p_3;0.42)} \mathcal{A}^a \cup \mathcal{A}^b \\ \mathcal{A}^b &= \bigcup_{i \in \Sigma_b} T_i^b \mathcal{A}^{\delta(b,i)} \\ &= T_0^b \mathcal{A}^b \cup T_1^b \mathcal{A}^b \cup T_2^b \mathcal{A}^b \cup T_3^b \mathcal{A}^a \\ &= h_{(p_0;0.5)} \mathcal{A}^b \cup h_{(p_1;0.5)} \mathcal{A}^b \cup h_{(p_2;0.5)} \mathcal{A}^b \cup h_{(p_4;0.16)} \mathcal{A}^a \end{aligned}$$

L'attracteur  $\mathcal{A}^a$  s'exprime récursivement comme un ensemble d'une copie de lui-même, et d'une copie de  $\mathcal{A}^b$ . L'attracteur  $\mathcal{A}^b$  s'exprime récursivement comme un ensemble de trois copies de lui-même, et d'une copie de  $\mathcal{A}^a$ . Les deux attracteurs  $\mathcal{A}^a$  et  $\mathcal{A}^b$  sont donc définis de manière mutuellement récursive.

Par rapport à l'exemple précédent, l'expression de  $\mathcal{A}^a$  en fonction de  $\mathcal{A}^a$  et de  $\mathcal{A}^b$  reste inchangée. Cependant,  $\mathcal{A}^b$  est modifié car il contient désormais une copie de  $\mathcal{A}^a$ . Cela entraîne donc également la modification de  $\mathcal{A}^a$ .

Dans cet exemple, nous avons introduit un nouveau circuit élémentaire, de longueur 2, passant par  $a$  et  $b$ . On donne ici les circuits élémentaires partant de chaque état  $a$  et  $b$ .

$$\begin{array}{ll} \delta(a, 0) & = a & \delta(b, 0) & = b \\ \delta(a, 1\ 3) & = a & \delta(b, 1) & = b \\ & & \delta(b, 2) & = b \\ & & \delta(b, 3\ 1) & = b \end{array}$$

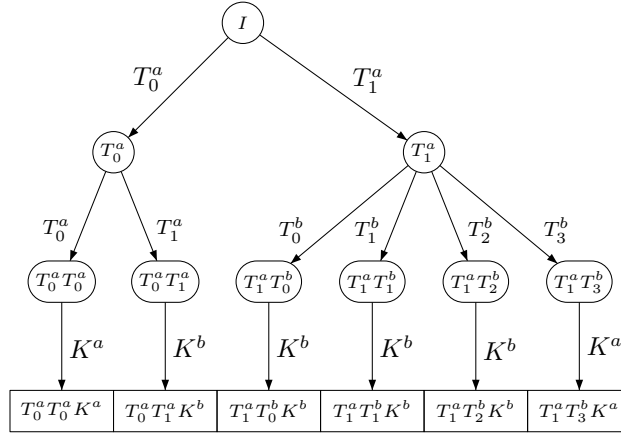
À chacun de ces circuits est associé un opérateur ou un opérateur composé.

$$\begin{array}{ll} T_0^a & = h_{(p_3;0.42)} & T_0^b & = h_{(p_0;0.5)} \\ T_1^a T_3^b & = Id \circ h_{(p_3;0.16)} & T_1^b & = h_{(p_1;0.5)} \\ & = h_{(p_3;0.16)} & T_2^b & = h_{(p_2;0.5)} \\ & & T_3^b T_1^a & = h_{(p_3;0.16)} \circ Id \\ & & & = h_{(p_3;0.16)} \end{array}$$

Les opérateurs associés à ces circuits sont contractants. Il existe donc un unique compact non vide  $\mathcal{A}^a$  et un unique compact non vide  $\mathcal{A}^b$  vérifiant le système  $\mathcal{A}^x = \bigcup_{i \in \Sigma_x} T_i^x \mathcal{A}^{\delta(x,i)}$ .

### 2.2.3 Algorithme itératif pour le calcul d'un attracteur

Dans un CIFS, chaque état  $x$  est associé à un attracteur  $\mathcal{A}^x$ . De la même manière que pour les IFS, chacun de ces attracteurs peut être approximé par une suite de figures  $(K_n^x)_{n \in \mathbb{N}}$  qui tend vers  $\mathcal{A}^x$ . On attribue à chaque état une primitive d'affichage  $K^x \in \mathcal{H}(E^x)$



**Fig. 2.14** – Arbre de calcul de  $K_2^a$ . Les nœuds internes correspondent au calcul d'une fonction composée. Les feuilles correspondent à une sous-figure de  $K_2^a$  à afficher

définissant le terme initial de chaque suite  $(K_n^x)_{n \in \mathbb{N}}$ . Les termes suivants sont définis de manière mutuellement récursive.

$$K_0^x = K^x$$

$$K_{n+1}^x = \bigcup_{i \in \Sigma_x} T_i^x K_n^{\delta(x,i)}$$

Chaque terme  $K_n^x$  est composé d'un ensemble d'images de  $K^y$ , avec  $y \in \mathcal{Q}$ , par une fonction composée.

Par exemple, voici l'expression développée des termes de  $(K_n^a)$  et  $(K_n^b)$  pour le CIFS défini dans l'exemple précédent.

$$K_1^a = T_0^a K^a \cup T_1^a K^b$$

$$K_2^a = T_0^a K_1^a \cup T_1^a K_1^b$$

$$= T_0^a T_0^a K^a \cup T_0^a T_1^a K^b \cup$$

$$T_1^a T_0^b K^b \cup T_1^a T_1^b K^b \cup T_1^a T_2^b K^b \cup T_1^a T_3^b K^a$$

$$\vdots \quad \quad \quad \vdots$$

$$K_1^b = T_0^b K^b \cup T_1^b K^b \cup T_2^b K^b \cup T_3^b K^a$$

$$K_2^b = T_0^b K_1^b \cup T_1^b K_1^b \cup T_2^b K_1^b \cup T_3^b K_1^a$$

$$= T_0^b T_0^b K^b \cup T_0^b T_1^b K^b \cup T_0^b T_2^b K^b \cup T_0^b T_3^b K^a \cup$$

$$T_1^b T_0^b K^b \cup T_1^b T_1^b K^b \cup T_1^b T_2^b K^b \cup T_1^b T_3^b K^a \cup$$

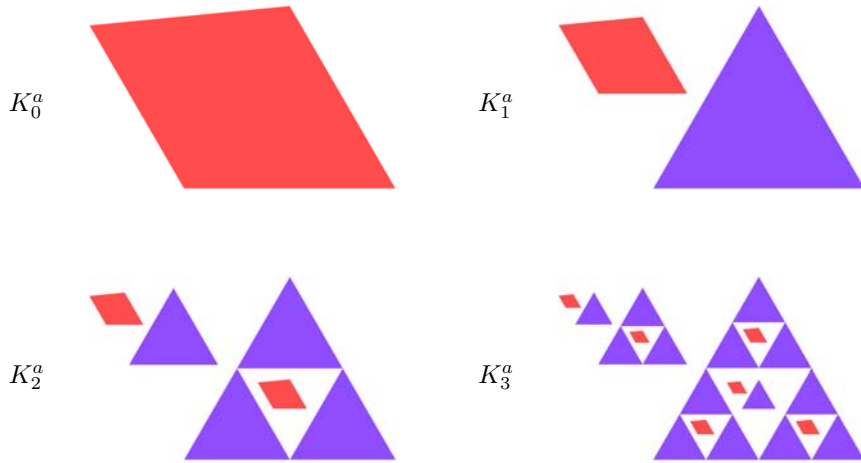
$$T_2^b T_0^b K^b \cup T_2^b T_1^b K^b \cup T_2^b T_2^b K^b \cup T_2^b T_3^b K^a \cup$$

$$T_3^b T_0^b K^a \cup T_3^b T_1^b K^b$$

$$\vdots \quad \quad \quad \vdots$$

## 2.2.4 Langages et fonction d'adressage

À chaque état  $x$  de l'automate, on définit une fonction d'adressage permettant d'adresser un point de l'attracteur  $\mathcal{A}^x$  ou bien une composante de  $K_n^x$  en fonction d'un mot infini ou fini sur un langage donné.



**Fig. 2.15** – Construction itérative de la suite de figures approximant l'attracteur  $\mathcal{A}^a$ . La primitive  $K^a$  et ses copies sont des quadrilatères (en rouge), et les copies de  $K^b$  sont des triangles (en violet).

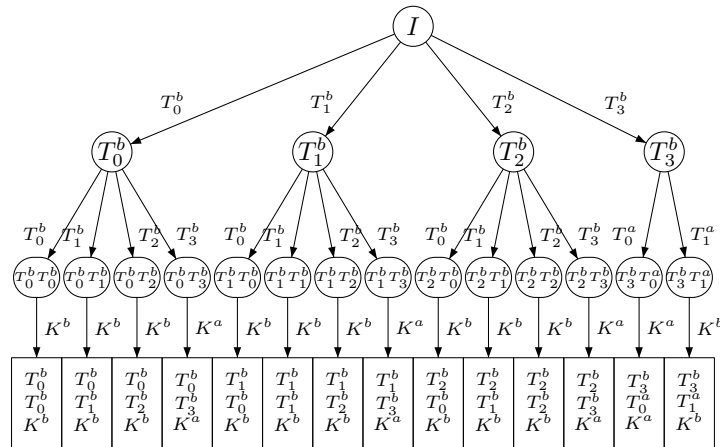
### Mots finis

Pour définir ce langage, on utilise l'indexation sur  $\Sigma$  des transitions  $\delta(x, i)$ . On définit  $\mathcal{L}_x^n$  l'ensemble des mots correspondant aux chemins dans l'automate partant de l'état  $x$  et de longueur  $n$ . L'ensemble  $\mathcal{L}_0^x$  est le singleton contenant le mot vide  $\{\varepsilon\}$ .

$$\begin{aligned} \mathcal{L}_0^x &= \{\varepsilon\} \\ \mathcal{L}_{n+1}^x &= \bigcup_{i \in \Sigma_x} i \mathcal{L}_n^{\delta(x, i)} \end{aligned}$$

À tout mot fini  $\sigma \in \mathcal{L}_n^x$  est associé l'opérateur  $T_\sigma^x$ , composé d'une suite d'opérateurs de CIFS indexés par les éléments de  $\sigma = \sigma_1 \dots \sigma_n$ .

$$T_\sigma^x = T_{\sigma_1}^x T_{\sigma_2}^{\delta(x, \sigma_1)} T_{\sigma_3}^{\delta(x, \sigma_1 \sigma_2)} \dots T_{\sigma_n}^{\delta(x, \sigma_1 \sigma_2 \dots \sigma_{n-1})}$$



**Fig. 2.16** – Arbre de calcul de  $K_2^b$ . Les nœuds internes correspondent au calcul d'une fonction composée. Les feuilles correspondent à une sous-figure de  $K_2^b$  à afficher

Chaque mot fini  $\sigma \in \mathcal{L}_n^x$  permet d'adresser la composante de  $K_n^x$  correspondant à  $T_\sigma^x K^{\delta(x,\sigma)}$ .

$$K_n^x = \bigcup_{\sigma \in \mathcal{L}_n^x} T_\sigma^x K^{\delta(x,\sigma)}$$

### Mots infinis

On note  $\mathcal{L}_\omega^x$  l'ensemble des mots de longueur infinie associés à l'état  $x$ .

$$\mathcal{L}_\omega^x = \{\sigma \in \Sigma^\omega, \forall n \in \mathbb{N}, \sigma_1 \dots \sigma_n \in \mathcal{L}_n^x\}$$

La famille de langages  $(\mathcal{L}_\omega^x)_{x \in \mathcal{Q}}$  vérifie un système d'équations récursives.

$$\mathcal{L}_\omega^x = \bigcup_{i \in \Sigma_x} i \mathcal{L}_\omega^{\delta(x,i)}$$

À chaque état  $x$  on associe une fonction d'adressage  $\varphi^x$  définie sur  $\mathcal{L}_\omega^x$ . À chaque mot infini, elle associe un point de  $E^x$  faisant partie de l'attracteur  $\mathcal{A}^x$ .

$$\begin{aligned} \varphi^x : \mathcal{L}_\omega^x &\rightarrow E^x \\ \forall \sigma \in \mathcal{L}_\omega^x, \varphi^x(\sigma) &= \lim_{n \rightarrow \infty} T^x(\sigma_1 \dots \sigma_n)p \end{aligned}$$

La fonction d'adressage  $\varphi^x$  permet de représenter l'attracteur  $\mathcal{A}^x$  sous forme paramétrée sur l'ensemble des mots infinis.

$$\mathcal{A}^x = \varphi^x(\mathcal{L}_\omega^x)$$

## 2.2.5 Exemples d'utilisations du modèle CIFS

### IFS simple

Le modèle CIFS est une généralisation du modèle IFS. Un IFS se représente dans le formalisme CIFS par un automate contenant un unique état  $x$ . Chaque opérateur  $T_i$  de l'IFS est représenté par une transition qui boucle sur  $x$ , comme le montre la figure 2.17.

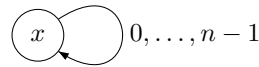


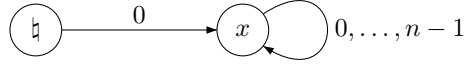
Fig. 2.17 – Automate décrivant un IFS

Par ailleurs, comme on l'a vu dans le premier exemple de CIFS, on peut définir un IFS comme faisant partie d'un CIFS, lorsque les transitions sortantes d'un état  $x$  bouclent sur ce même état, alors  $x$  correspond à un IFS, et  $\mathcal{A}^x$  est l'attracteur de l'IFS défini par les opérateurs  $\{T_i^x\}_{i \in \Sigma_x}$ .

### IFS projeté

Comme on l'a dit précédemment, à chaque état  $x$  de l'automate est associé un espace  $E^x$ . Nous avons vu jusqu'ici des exemples où les espaces associés à chaque état étaient les mêmes. Cependant, dans le cas général, on utilisera la possibilité d'associer des espaces différents pour chaque état. L'exemple simple de CIFS utilisant des espaces différents est celui des IFS projetés.

Un IFS projeté est donné par un IFS  $\{T_i\}$  défini dans un espace  $E^x$ , ainsi que par un opérateur de projection  $T_0^{\natural} : E^{\natural} \leftarrow E^x$ .



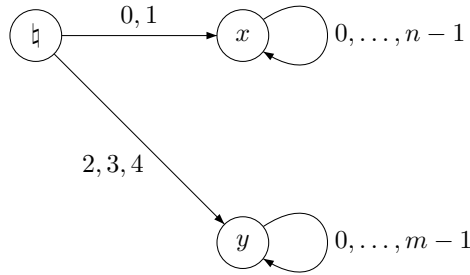
**Fig. 2.18** – Automate décrivant un IFS projeté

L'espace  $E^{\natural}$  est l'espace de modélisation, qui est généralement l'espace 3D classique, ou homogène.

$E^x$  est l'espace dans lequel les  $T_i$  opèrent et dans lequel l'attracteur  $\mathcal{A}^x$  est défini. Cet espace est généralement de dimension plus grande que l'espace de modélisation  $E^{\natural}$ . L'objet modélisé est  $\mathcal{A}^{\natural}$ , qui correspond à une projection dans  $E^{\natural}$  de l'attracteur  $\mathcal{A}^x$ .

$$\mathcal{A}^{\natural} = T_0^{\natural} \mathcal{A}^x$$

Ce principe peut être généralisé en définissant plusieurs projections  $T_i^{\natural}$  d'un attracteur d'IFS  $\mathcal{A}^x$  ou d'attracteurs différents. L'exemple d'automate sur la figure 2.19 illustre la description d'une scène, associée à l'état  $\natural$ , composée de cinq projections d'IFS, dont deux sont des IFS de type  $x$  et trois de type  $y$ .



**Fig. 2.19** – Automate décrivant une scène composée de projections de deux IFS de type  $x$  et de trois IFS de type  $y$

### Remarque

Les espaces  $E^{\natural}$  et  $E^{x \neq \natural}$  étant indépendants, il est possible de définir  $E^x$  comme un espace barycentrique de dimension  $n$  quelconque. Les opérateurs  $\{T_i^x\}$  sont alors des matrices barycentriques, et la projection  $\{T_i^{\natural}\}$  est une matrice dont l'ensemble des colonnes constitue un ensemble de points de contrôle. Un IFS projeté défini en coordonnées barycentriques permet de définir une forme à pôles. Ce formalisme sera développé dans le chapitre 4.



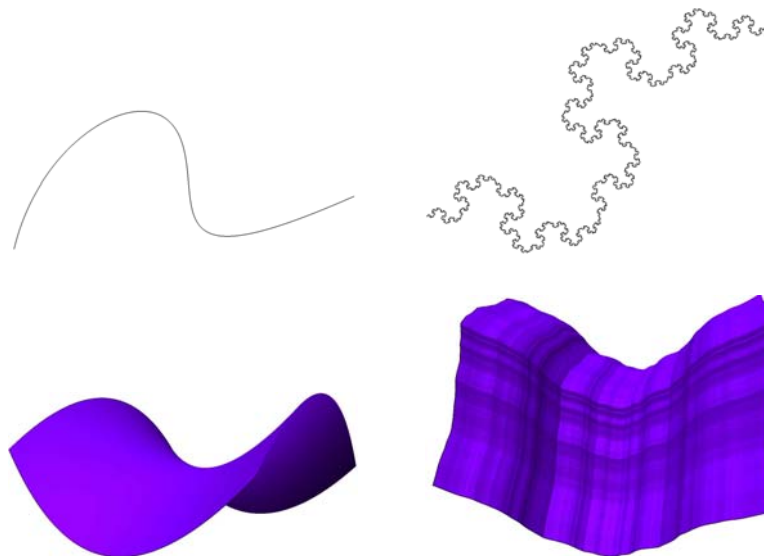


## Chapitre 3

# Contrôle topologique

Les modèles IFS et CIFS que nous avons vus précédemment permettent de modéliser un ensemble de formes varié. Cependant, il est difficile de contrôler les propriétés topologiques des objets modélisés. Les formes sont déterminées par la donnée d'un ensemble d'opérateurs géométriques. La modification de ces opérateurs entraîne des modifications de la figure aussi bien globales que locales, ce qui affecte non seulement la géométrie, mais également la topologie. Par exemple, en diminuant légèrement les rapports d'homothétie d'un triangle de SIERPINSKI, on obtient un nuage de points totalement discontinu (voir figures 2.2 et 2.4).

Afin de contrôler la structure topologique des objets modélisés, nous avons défini un nouveau modèle, le modèle BCIFS (Boundary Controlled Iterated Function System). Il s'appuie sur le modèle CIFS, et permet d'établir un certain nombre de contraintes afin de garantir une structure topologique spécifiée.



**Fig. 3.1** – *Exemples de formes à topologie classique*

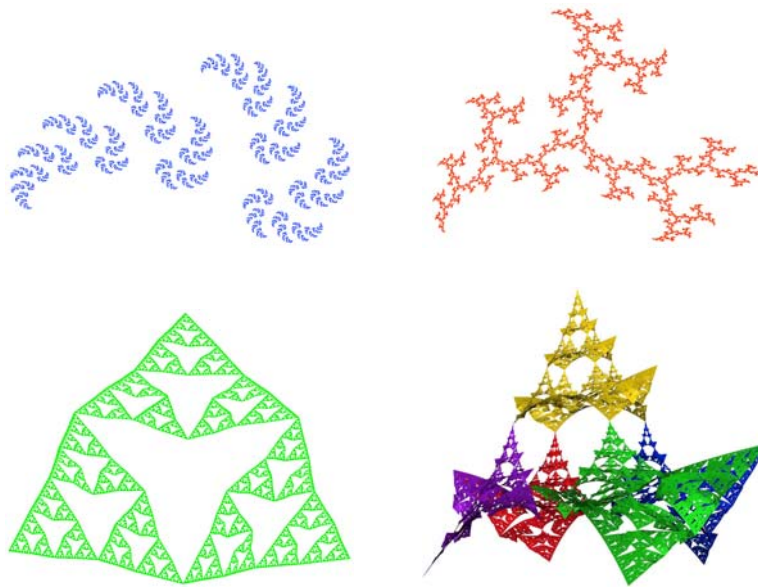


Fig. 3.2 – Exemples de formes à topologie fractale

### 3.1 Le modèle BCIFS (Boundary Controlled Iterated Function System)

Le modèle BCIFS s'exprime en complétant par une description B-Rep le modèle CIFS formalisé avec un automate.

#### 3.1.1 B-Rep (Boundary representation)

Le modèle B-Rep est couramment utilisé en CAO pour représenter les objets géométriques. Ce formalisme représente de manière distincte la structure topologique et le plongement géométrique.

La structure topologique est codée par un ensemble de cellules topologiques (faces, arêtes, sommets) liées entre elles par un ensemble de relations d'incidence et d'adjacence. La relation d'incidence est basée sur l'inclusion de cellules : chaque face est bordée par un ensemble d'arêtes, et chaque arête est bordée par deux sommets. La relation d'adjacence est basée sur la mise en commun de cellules : deux faces adjacentes sont bordées par une arête commune, et deux arêtes adjacentes sont bordées par un même sommet.

En complément de ceci, le plongement géométrique décrit le positionnement dans l'espace de chacune de ces cellules.

Le modèle BCIFS est une extension du modèle CIFS qui intègre les caractéristiques des B-Rep. Dans ce chapitre, on introduira le modèle BCIFS par sa manière de décrire des structures topologiques. Nous parlerons du plongement géométrique des BCIFS au chapitre 4.

#### B-Rep classique

Dans la formulation B-Rep (Boundary representation) utilisée en CAO, on distingue la structure combinatoire et le plongement géométrique. La structure combinatoire code les relations d'incidence et d'adjacence entre composantes [Kal83][Bri88][Lie91]. Le plongement

géométrique code avec des paramètres numériques (coordonnées de points et coefficients) la description de chaque composante (forme à pôles).

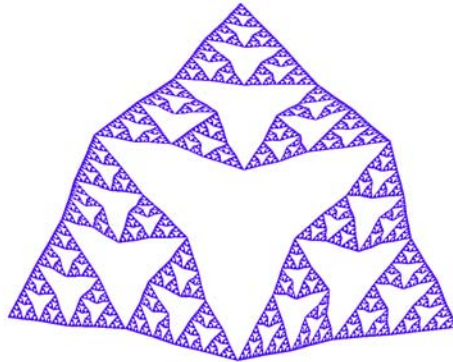
L'approche classique est basée sur la notion de complexe cellulaire. Les composantes sont des cellules topologiques, c.a.d. des figures homéomorphes à des boules (un arc est une déformation de l'intervalle  $[0, 1]$ , un carreau est une déformation du pavé  $[0, 1]^2$ ). Dans cette approche, il y a une identification entre cellules combinatoires et cellules topologiques.

La distinction entre cellules combinatoires et cellules topologiques permet de caractériser des structures topologiques plus générales.

## B-Rep fractale

Pour qualifier la topologie fractale, on distingue clairement structure combinatoire et "matière" topologique. Les composantes ne sont pas des cellules topologiques, mais elles possèdent des "côtés" et on peut leur appliquer des relations d'incidence et d'adjacence.

La figure figure 3.3 montre une variante géométrique du triangle de SIERPINSKI. Sa dimension topologique vaut 1 mais, comme le triangle surfacique classique, il possède une face, trois arêtes et trois sommets (voir figure 3.7).



**Fig. 3.3** – Exemple d'attracteur homéomorphe à un triangle de SIERPINSKI

La face n'est pas homéomorphe à un disque, mais les arêtes sont homéomorphes à des segments de droite. L'introduction de la notion de cellule itérative, permet d'avoir une définition formelle de la "topologie fractale". Une cellule, au sens itératif, est un attracteur pour un ensemble d'opérateurs de subdivision  $T_i^x$ . Elle n'est en général pas homéomorphe à une boule. Mais sous certaines conditions de raccord sur les opérateurs de subdivision, on peut définir des attracteurs homéomorphes à des boules (arcs, carreaux, ...).

La face correspond à l'attracteur d'un IFS composé de trois opérateurs :

$$\mathcal{A}^f = \mathcal{A}(\{T_0^f, T_1^f, T_2^f\})$$

Les arêtes et les sommets bordant cet attracteur sont des sous-ensembles de  $\mathcal{A}^f$ , et correspondent aux attracteurs d'IFS composés de sous-ensembles de  $\{T_0^f, T_1^f, T_2^f\}$ . Chaque arête correspond à un sous-attracteur composé de deux opérateurs  $\mathcal{A}_{i,j} = \mathcal{A}(\{T_i^f, T_j^f\})$ . Chaque sommet correspond à un sous-attracteur composé d'un seul opérateur  $\mathcal{A}_i = \mathcal{A}(\{T_i^f\})$ .

### 3.1.2 Formulation itérative de la B-Rep

Les figures à description classique comportent un nombre fini de composantes. Chaque figure composée est représentée par un *complexe cellulaire*, c.a.d. un ensemble fini de cellules

muni d'une relation d'incidence. Chaque cellule (carreau-arc-singleton) est représentée par une fonction polynomiale ou rationnelle (NURBS). Leur structure topologique est codée par une structure combinatoire finie (graphe, carte).

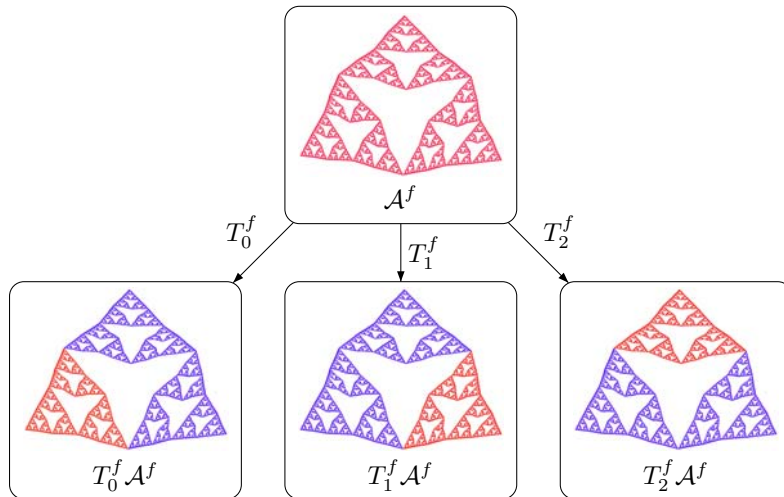
Les figures à description itérative sont constituées de composantes générées par subdivision. À chaque étape, l'algorithme de visualisation des cellules engendre des maillages dont le nombre de composantes est potentiellement infini (voir figure 2.8). Mais leur structure topologique va être codée par un ensemble fini de règles, selon lesquelles chaque niveau se déduit du niveau précédent.

### Définition des cellules

Chaque cellule correspond à un attracteur dans un certain espace.

La face correspond à l'attracteur  $\mathcal{A}^f$  d'un IFS composé de trois opérateurs dans  $E^f$  (voir figure 3.4).

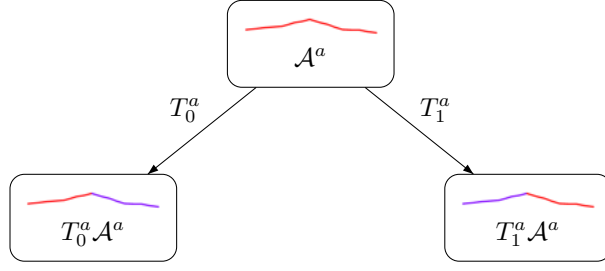
$$\begin{aligned}\mathcal{A}^f &= \mathcal{A}(\{T_0^f, T_1^f, T_2^f\}) \in \mathcal{H}(E^f) \\ T_i^f &: E^f \leftarrow E^f\end{aligned}$$



**Fig. 3.4** – Structure de subdivision de la face  $f$

Une arête correspond à l'attracteur  $\mathcal{A}^a$  d'un IFS composé de deux opérateurs dans  $E^a$  (voir figure 3.5).

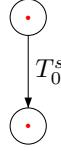
$$\begin{aligned}\mathcal{A}^a &= \mathcal{A}(\{T_0^a, T_1^a\}) \in \mathcal{H}(E^a) \\ T_i^a &: E^a \leftarrow E^a\end{aligned}$$



**Fig. 3.5** – Structure de subdivision de l'arête  $a$

Un sommet correspond à l'attracteur  $\mathcal{A}^s$  d'un IFS composé d'un opérateur dans  $E^s$  (voir figure 3.6).

$$\begin{aligned} \mathcal{A}^s &= \mathcal{A}(\{T_0^s\}) \in \mathcal{H}(E^s) \\ T_0^s &: E^s \leftarrow E^s \end{aligned}$$



**Fig. 3.6** – Structure de subdivision du sommet  $s$

### Définition des opérateurs d'incidence

On représente sur la figure 3.7 la structure en face, arêtes et sommets du triangle de SIERPINSKI. Pour cela, on utilise des opérateurs d'incidence, généralisant la notion de bord. Chaque cellule incidente à une autre est définie par un plongement dans celle-ci. La face est bordée par trois arêtes et chaque arête est bordée par deux sommets.

Les trois arêtes bordant la face correspondent au plongement de  $\mathcal{A}^a$  dans  $E^f$  par trois opérateurs d'incidence  $(\Pi_i^f)_{i \in \{0,1,2\}}$ .

$$\begin{aligned} \mathcal{A}^f &\supset \Pi_i^f \mathcal{A}^a \\ \Pi_i^f &: E^f \leftarrow E^a \end{aligned}$$

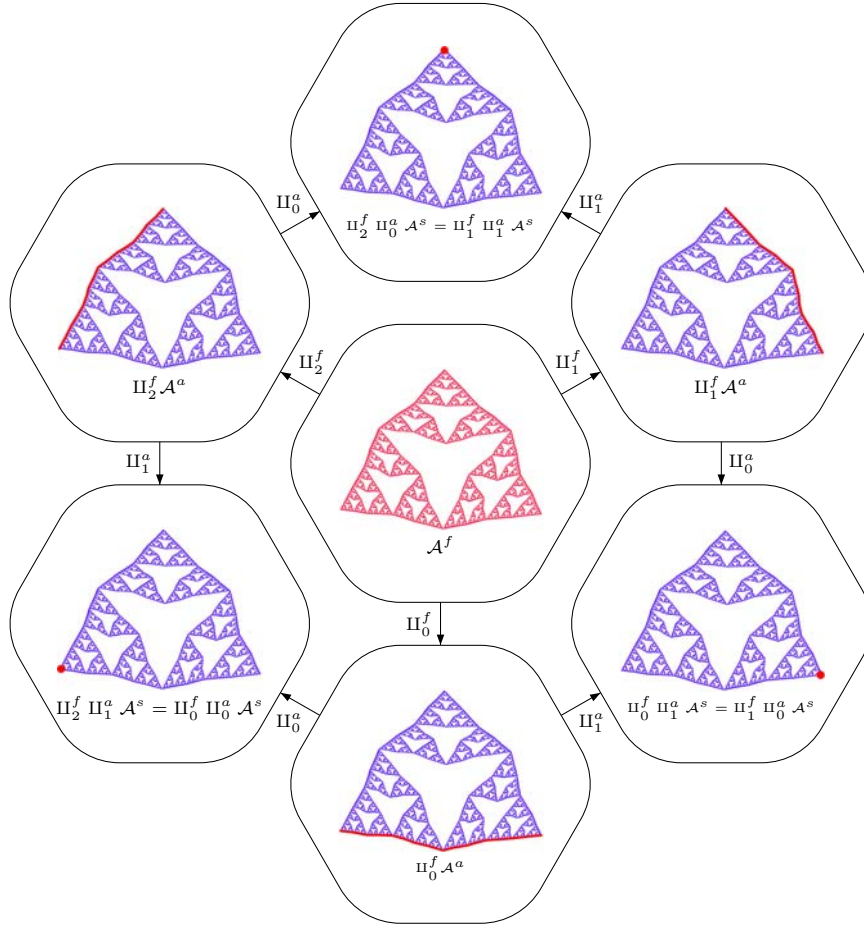
Les deux sommets bordant chaque arête correspondent au plongement de  $\mathcal{A}^s$  dans  $E^a$  par deux opérateurs d'incidence  $(\Pi_i^a)_{i \in \{0,1\}}$ .

$$\begin{aligned} \mathcal{A}^a &\supset \Pi_i^a \mathcal{A}^s \\ \Pi_i^a &: E^a \leftarrow E^s \end{aligned}$$

Les relations d'incidence et d'adjacence peuvent alors s'exprimer au travers des opérateurs de subdivision et d'incidence.

### Équations d'adjacence sur les opérateurs d'incidence

Les arêtes bordant la face partagent certains sommets entre elles. La mise en commun des sommets est représentée sur la figure 3.7. Cela se traduit par des contraintes d'adjacence



**Fig. 3.7** – Décomposition du triangle de SIERPINSKI en face, arêtes et sommets définis par plongements

sur les opérateurs de plongement  $(\Pi_i^f)_{i \in \{0,1,2\}}$ .

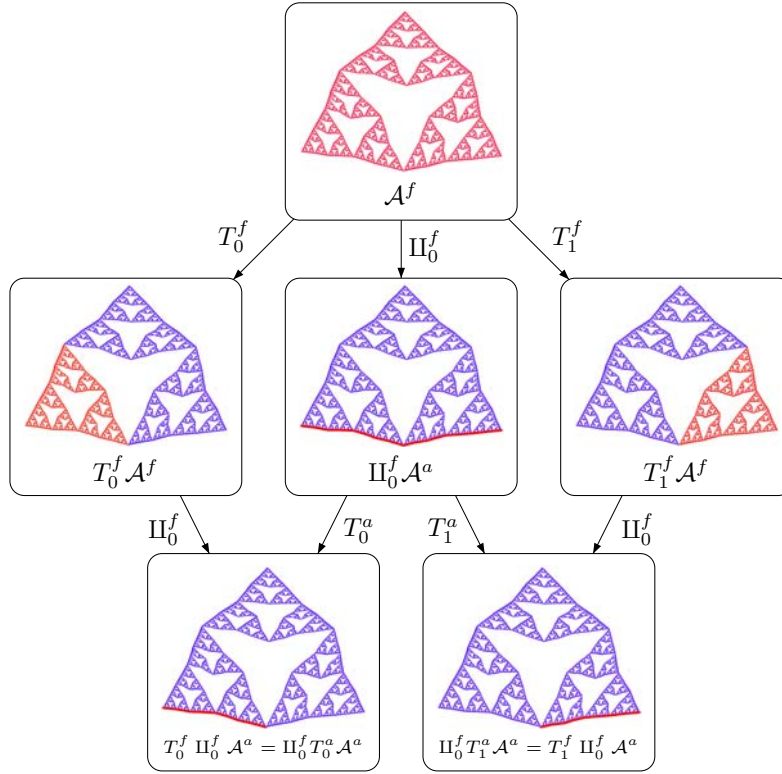
$$\begin{aligned} \Pi_0^f \Pi_1^alpha &= \Pi_1^f \Pi_0^alpha \\ \Pi_1^f \Pi_1^alpha &= \Pi_2^f \Pi_0^alpha \\ \Pi_2^f \Pi_1^alpha &= \Pi_0^f \Pi_0^alpha \end{aligned}$$

### Équations d'incidence

La face et les arêtes sont définis par leurs propres opérateurs de subdivision  $(T_i^f)_{i \in \{0,1,2\}}$  et  $(T_j^alpha)_{j \in \{0,1\}}$ . Il y a des correspondances entre la subdivision d'un bord et le bord d'une subdivision. Cette propriété est vérifiée car les opérateurs de subdivision de la face  $(T_i^f)_{i \in \{0,1,2\}}$  sont liés avec ceux d'une arête  $(T_j^alpha)_{j \in \{0,1\}}$  par un ensemble d'équations d'incidence.

Le bord  $\partial_0$  se subdivise en deux sous-arêtes. La première correspond au bord  $\partial_0$  de la sous-face de subdivision 0, et la seconde correspond au bord  $\partial_0$  de la sous-face de subdivision 1 (voir figure 3.8).

$$\begin{aligned} \Pi_0^f T_0^alpha &= T_0^f \Pi_0^f \\ \Pi_0^f T_1^alpha &= T_1^f \Pi_0^f \end{aligned}$$



**Fig. 3.8** – Équations d'incidence mettant en correspondance le bord d'une subdivision avec la subdivision du bord  $\partial_0$

Ces équations sur les matrices de subdivision permettent d'établir le plongement d'attracteurs  $\Pi_0^f \mathcal{A}^a = \mathcal{A}(\{T_0^f, T_1^f\})$ . En effet, on a :

$$T_0^f \Pi_0^f \mathcal{A}^a \cup T_1^f \Pi_0^f \mathcal{A}^a = \Pi_0^f T_0^a \mathcal{A}^a \cup \Pi_0^f T_1^a \mathcal{A}^a = \Pi_0^f (T_0^a \mathcal{A}^a \cup T_1^a \mathcal{A}^a) = \Pi_0^f \mathcal{A}^a$$

$$\Pi_0^f \mathcal{A}^a \in \mathcal{H}(E^f), \quad \Pi_0^f \mathcal{A}^a = T_0^f(\Pi_0^f \mathcal{A}^a) \cup T_1^f(\Pi_0^f \mathcal{A}^a) \Leftrightarrow \Pi_0^f \mathcal{A}^a = \mathcal{A}(\{T_0^f, T_1^f\})$$

$\mathcal{A}(\{T_0^f, T_1^f\})$  est un sous-attracteur de  $\mathcal{A}^f = \mathcal{A}(\{T_0^f, T_1^f, T_2^f\})$  :

$$\mathcal{A}^f = \mathcal{A}(\{T_0^f, T_1^f, T_2^f\}) = \varphi^f(\{0, 1, 2\}^\omega) \supset \varphi^f(\{0, 1\}^\omega) = \mathcal{A}(\{T_0^f, T_1^f\})$$

Ceci permettent d'établir l'inclusion  $\mathcal{A}^f \supset \Pi_0^f \mathcal{A}^a$  car :

$$\mathcal{A}^f = \mathcal{A}(\{T_0^f, T_1^f, T_2^f\}) \supset \mathcal{A}(\{T_0^f, T_1^f\}) = \Pi_0^f \mathcal{A}(\{T_0^a, T_1^a\}) = \Pi_0^f \mathcal{A}^a$$

De la même façon sur le bord  $\partial_1$ , les équations d'incidence

$$\begin{aligned} \Pi_1^f T_0^a &= T_1^f \Pi_1^f \\ \Pi_1^f T_1^a &= T_2^f \Pi_1^f \end{aligned}$$

permettent d'établir l'inclusion  $\mathcal{A}^f \supset \Pi_1^f \mathcal{A}^a$  car :

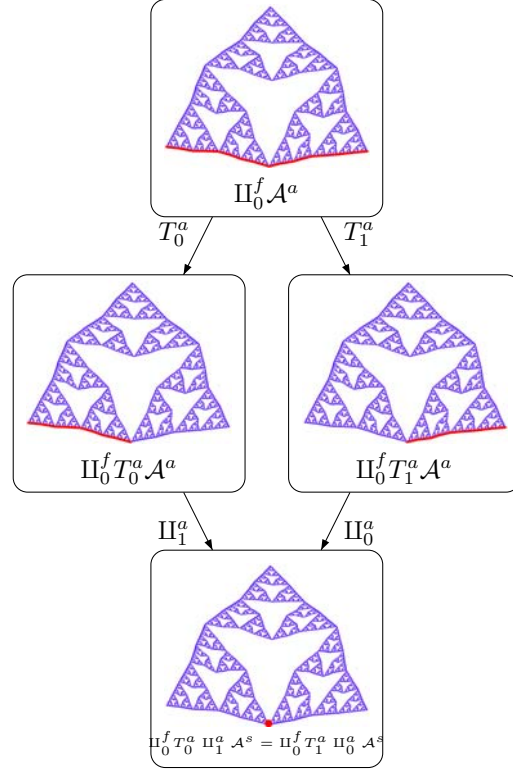
$$\mathcal{A}^f = \mathcal{A}(\{T_0^f, T_1^f, T_2^f\}) \supset \mathcal{A}(\{T_1^f, T_2^f\}) = \Pi_1^f \mathcal{A}(\{T_0^a, T_1^a\}) = \Pi_1^f \mathcal{A}^a$$

Enfin, sur le bord  $\partial_2$ , les équations d'incidence

$$\begin{aligned} \Pi_2^f T_0^a &= T_2^f \Pi_2^f \\ \Pi_2^f T_1^a &= T_0^f \Pi_2^f \end{aligned}$$

permettent d'établir l'inclusion  $\mathcal{A}^f \supset \Pi_2^f \mathcal{A}^a$  car :

$$\mathcal{A}^f = \mathcal{A}(\{T_0^f, T_1^f, T_2^f\}) \supset \mathcal{A}(\{T_0^f, T_2^f\}) = \Pi_2^f \mathcal{A}(\{T_0^a, T_1^a\}) = \Pi_2^f \mathcal{A}^a$$



**Fig. 3.9** – Équation d'adjacence entre les deux subdivisions d'un bord  $a$  de  $f$

### Équations d'adjacence sur les opérateurs de subdivision

Les sous-cellules issues de la subdivision d'une arête vérifient une relation d'adjacence.

$$T_0^a \Pi_1^a = T_1^a \Pi_0^a$$

Dans  $\mathcal{A}^a$ , on a alors

$$T_0^a \Pi_1^a \mathcal{A}^s = T_1^a \Pi_0^a \mathcal{A}^s$$

et dans  $\mathcal{A}^f$ , on a (voir figure 3.9)

$$\Pi_i^f T_0^a \Pi_1^a \mathcal{A}^s = \Pi_i^f T_1^a \Pi_0^a \mathcal{A}^s$$

### 3.1.3 Formalisation du modèle BCIFS

Dans la formulation BCIFS, les composantes sont liées entre elles par des opérateurs d'incidence et de subdivision associés à un automate. Les relations d'incidence et d'adjacence sont décrites par un ensemble de règles d'équivalence de chemins sur l'automate. Ces règles d'équivalence définissent un système de contraintes sur les opérateurs de subdivision.



## Automate

L'ensemble des états  $\mathcal{Q}$  est partitionné :

1. états initiaux correspondant aux figures composées (cercles, sphères, tores, cylindres, ...) :  
 $\mathcal{Q}_I = \{i; \dots\}$ ;
2. états terminaux correspondant aux cellules combinatoires, partitionnés selon leur dimension combinatoire  $\dim_C \mathcal{Q}_C = \mathcal{Q}_S \cup \mathcal{Q}_A \cup \mathcal{Q}_F$  avec :
  - $\mathcal{Q}_F = \{f; g; \dots\}$  faces,  $\dim_C(f) = 2$ ;
  - $\mathcal{Q}_A = \{a; b; \dots\}$  arêtes,  $\dim_C(a) = 1$ ;
  - $\mathcal{Q}_S = \{s; t; \dots\}$  sommets,  $\dim_C(s) = 0$ ;
à chaque état  $x$  est associée une figure primitive  $K^x$  (carré ou triangle, segment de droite, singleton),

L'ensemble des symboles  $\Sigma$  est partitionné en :

1. symboles de subdivision  $\Sigma_{\dot{\cdot}} = \{0; 1; 2; \dots\}$ ;
2. symboles d'incidence  $\Sigma_{\partial} = \{\partial_0; \partial_1; \partial_2; \dots\}$ .

En prenant  $f \in \mathcal{Q}_F$ ,  $a \in \mathcal{Q}_A$ ,  $s \in \mathcal{Q}_S$ , deux actions de transition sont distinguées :

1. subdivision :  $\delta(f, i) \in \mathcal{Q}_F$ ,  $\delta(a, i) \in \mathcal{Q}_A$ ,  $\delta(s, i) \in \mathcal{Q}_S$ ;
2. incidence :  $\delta(f, \partial_i) = a$ ,  $\delta(a, \partial_j) = s$ .

## Opérateurs

Les figures composées ou cellulaires de dimensions combinatoires différentes vont avoir des espaces avec de dimensions différentes. On introduit des opérateurs d'incidence permettant de plonger l'un dans l'autre  $\Pi_u^x : E^x \leftarrow E^{\delta(x,u)}$  pour tout symbole d'incidence  $u \in \Sigma_{\partial}$ .

$$u \in \Sigma_{\partial}^x \Rightarrow \dim_C(x) = \dim_C(\delta(x,u)) + 1$$

À chaque symbole de subdivision  $i \in \Sigma_{\dot{\cdot}}^x$  est associé l'opérateur de subdivision  $T_i^x : E^x \leftarrow E^{\delta(x,i)}$ .

De manière plus générale, à chaque symbole  $u \in \Sigma^x = \Sigma_{\dot{\cdot}}^x \cup \Sigma_{\partial}^x$  est associé l'opérateur  $\hat{T}_u^x : E^x \leftarrow E^{\delta(x,u)}$ .

$$\begin{aligned} \hat{T}_i^x &= T_i^x & \text{pour } i \in \Sigma_{\dot{\cdot}}^x \\ \hat{T}_{\partial_i}^x &= \Pi_i^x & \text{pour } \partial_i \in \Sigma_{\partial}^x \end{aligned}$$

## Expressions

L'automate engendre un ensemble de mots  $\mathcal{L}(x, \mathcal{Q}_C)$ , Ces mots représentent des expressions décrivant des figures.

En notant  $|\theta|_C$  le nombre de symboles de subdivision apparaissant dans  $\theta$ , pour un état  $x \in \mathcal{Q}_I$  et un niveau donné  $n$ , les mots de  $\mathcal{L}_n(x, \mathcal{Q}_C) = \{\theta \in \Sigma^* / \delta(x, \theta) \in \mathcal{Q}_C, |\theta|_C = n\}$  représentent des expressions décrivant des composantes de  $K_{(n)}^x$ .

À chaque chemin  $\gamma_{\beta}^x$  est associé une figure calculée à l'aide de produits d'opérateurs de subdivision et d'opérateurs de côté :

$$K_{\beta}^x = \hat{T}_{\beta}^x K^{\delta(x,\beta)}$$

Deux chemins — ou deux expressions formelles — seront dits géométriquement équivalents si :

$$\gamma_{\beta}^x \simeq \gamma_{\beta'}^x \stackrel{\text{def}}{\Leftrightarrow} \delta(x, \beta) = \delta(x, \beta') \text{ et } \hat{T}_{\beta}^x = \hat{T}_{\beta'}^x$$

Deux expressions géométriquement équivalentes décrivent la même figure car :

$$\begin{aligned} \delta(x, \theta) &= \delta(x, \theta') \text{ et } \hat{T}_\theta^x = \hat{T}_{\theta'}^x \\ \Rightarrow \hat{T}_\theta^x K^{\delta(x, \theta)} &= \hat{T}_{\theta'}^x K^{\delta(x, \theta')} \\ \Rightarrow K_\theta^x &= K_{\theta'}^x \end{aligned}$$

### Règles d'équivalence

En introduisant une relation d'équivalence formelle sur les mots de  $\mathcal{L}(x, \mathcal{Q}_C)$ , garantissant l'équivalence géométrique et l'égalité des figures, les classes de mots  $C^x(\theta) \in \mathcal{L}_n(x, \mathcal{Q}_C) / \equiv$  constituent une représentation formelle des composantes  $K_\theta^x$ .

Cette relation d'équivalence est définie à partir d'un système fini de règles d'équivalence et de contraintes sur les états et les opérateurs. Ce système est décrit par l'ensemble  $\Upsilon^x$ , qui contient les couples de chemins  $(\theta, \theta')$  équivalents.

$$\forall(\theta, \theta') \in \Upsilon^x \begin{cases} \delta(x, \theta) &= \delta(x, \theta') \\ \hat{T}_\theta^x &= \hat{T}_{\theta'}^x \end{cases}$$

La description de la structure topologique peut alors se faire de manière formelle, à travers un système de règles d'équivalence. Les côtés de  $C^x(\theta)$  sont les classes  $C^x(\theta v)$  avec  $v \in \Sigma_\partial$ .

On distingue dans  $\Upsilon^x$  trois types de contraintes.

Les couples de la forme  $(\partial_i \partial_j, \partial_k \partial_\ell) \in \Upsilon^x$  décrivent une équation d'adjacence sur les opérateurs d'incidence. Elles s'appliquent aux cellules ayant  $\dim_C(x) \geq 2$  lorsque deux cellules incidentes sont adjacentes (par exemple, deux arêtes bordant une face ont un sommet en commun, voir figure 3.7).

$$\forall(\partial_i \partial_j, \partial_k \partial_\ell) \in \Upsilon^x \begin{cases} \delta(x, \partial_i \partial_j) &= \delta(x, \partial_k \partial_\ell) \\ \Pi_i^x \Pi_j^{\delta(x, \partial_i)} &= \Pi_k^x \Pi_\ell^{\delta(x, \partial_k)} \end{cases} \quad (3.1)$$

Les couples de la forme  $(i \partial_j, k \partial_\ell) \in \Upsilon^x$  décrivent une équation d'adjacence sur les opérateurs de subdivisions. Ces équations établissent les raccords de subdivision entre cellules par mise en commun d'une sous-cellule de dimension inférieure (par exemple, une arête se subdivise en deux arêtes raccordées en un sommet, voir figure 3.9).

$$\forall(i \partial_j, k \partial_\ell) \in \Upsilon^x \begin{cases} \delta(x, i \partial_j) &= \delta(x, k \partial_\ell) \\ T_i^x \Pi_j^{\delta(x, i)} &= T_k^x \Pi_\ell^{\delta(x, k)} \end{cases} \quad (3.2)$$

Enfin, les couples de la forme  $(\partial_i j, k \partial_\ell) \in \Upsilon^x$  décrivent une équation d'incidence. Pour chaque opérateur d'incidence  $\Pi_i^x$  de type  $x$ , et pour chaque opérateur de subdivision  $T_j^{\delta(x, \partial_i)}$  de la cellule incidente, les équations d'incidence font correspondre la subdivision du bord  $\Pi_i^x T_j^{\delta(x, \partial_i)}$  avec le bord d'une subdivision  $T_k^x \Pi_\ell^{\delta(x, k)}$  (voir figure 3.8). Elles établissent le plongement d'un attracteur dans un autre.

$$\forall(\partial_i j, k \partial_\ell) \in \Upsilon^x \begin{cases} \delta(x, \partial_i j) &= \delta(x, k \partial_\ell) \\ \Pi_i^x T_j^{\delta(x, \partial_i)} &= T_k^x \Pi_\ell^{\delta(x, k)} \end{cases} \quad (3.3)$$

### 3.1.4 Arbre de décomposition, et graphe quotient

À partir d'un état  $x$ , l'automate définit un arbre de décomposition, dont les nœuds correspondent aux chemins  $\gamma_\theta^x$ . En introduisant la relation d'équivalence sur les chemins, on définit un graphe quotient, dont les nœuds correspondent aux classes d'équivalence  $C^x(\theta)$ .

Les nœuds  $\gamma_\theta^x$  de ces arbres sont étiquetés par les états  $\delta(x, \theta)$  et les branches  $(\gamma_\theta^x, \gamma_{\theta u}^x)$  sont étiquetées par les symboles  $u \in \Sigma$ . Les graphes quotients sont déduits des arbres par regroupement des nœuds équivalents. Les étiquettes restent inchangées car  $\gamma_\theta^x \simeq \gamma_{\theta'}^x \Rightarrow \delta(x, \theta) = \delta(x, \theta')$ .

Les arbres de décomposition et les graphes quotients associés à un BCIFS sont illustrés sur les exemples suivants.

## 3.2 Topologies classiques

Dans les exemples qui suivent, les cellules sont homéomorphes à des boules, et les opérateurs d'incidence sont des opérateurs de bord.

### 3.2.1 Arc de courbe

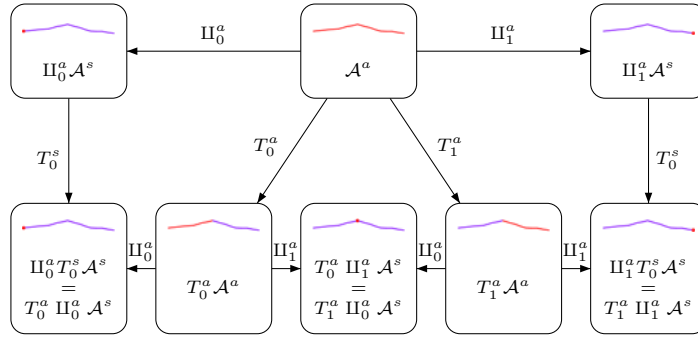


Fig. 3.10 – Structure de raccord d'un arc de courbe  $a$

L'exemple le plus simple de BCIFS décrivant une topologie filaire est celui d'un arc de courbe à deux subdivisions. Nous détaillons ici la description BCIFS correspondante.

On note  $a$  l'état associé à la courbe, et  $s$  l'état associé à ses extrémités. On établit dans un premier temps les relations d'incidence entre  $a$  et  $s$ .

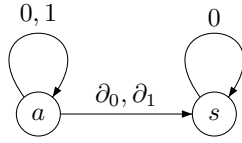
$$\begin{aligned}\delta(a, \partial_0) &= s \\ \delta(a, \partial_1) &= s\end{aligned}$$

pour spécifier qu'une arête de type  $a$  est bordée par deux sommets de type  $s$ .

On déclare également dans l'automate les deux relations de subdivision de  $a$ , ainsi qu'une relation de subdivision pour  $s$ .

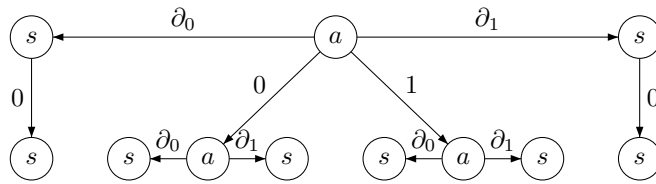
$$\begin{aligned}\delta(a, 0) &= a & \delta(s, 0) &= s \\ \delta(a, 1) &= a\end{aligned}$$

On obtient un automate contenant les relations de bordure, ainsi que les relations de subdivision, comme le montre la figure 3.11.



**Fig. 3.11** – Automate complet

On obtient d'après cet automate un arbre de décomposition à partir de l'état  $a$  représenté sur la figure 3.12.



**Fig. 3.12** – Arbre de décomposition associé à l'automate

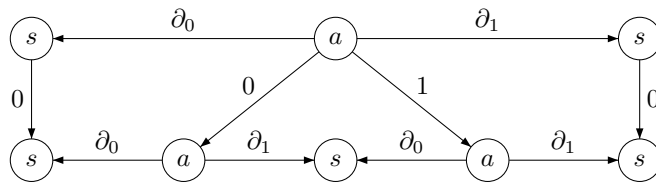
Nous introduisons maintenant les équivalences de chemins de manière à ce que la subdivision de l'arête vérifie les équations d'incidence 3.3 :

$$\Pi_0^a T_0^s = T_0^a \Pi_0^a \quad \Pi_1^a T_0^s = T_1^a \Pi_1^a$$

et d'adjacence 3.2 :

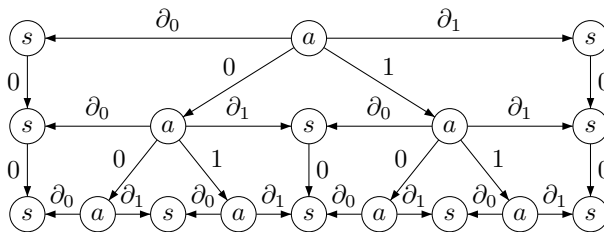
$$T_0^a \Pi_1^a = T_1^a \Pi_0^a$$

Ces équivalences de chemins se traduisent sur le graphe quotient représenté sur la figure 3.13.

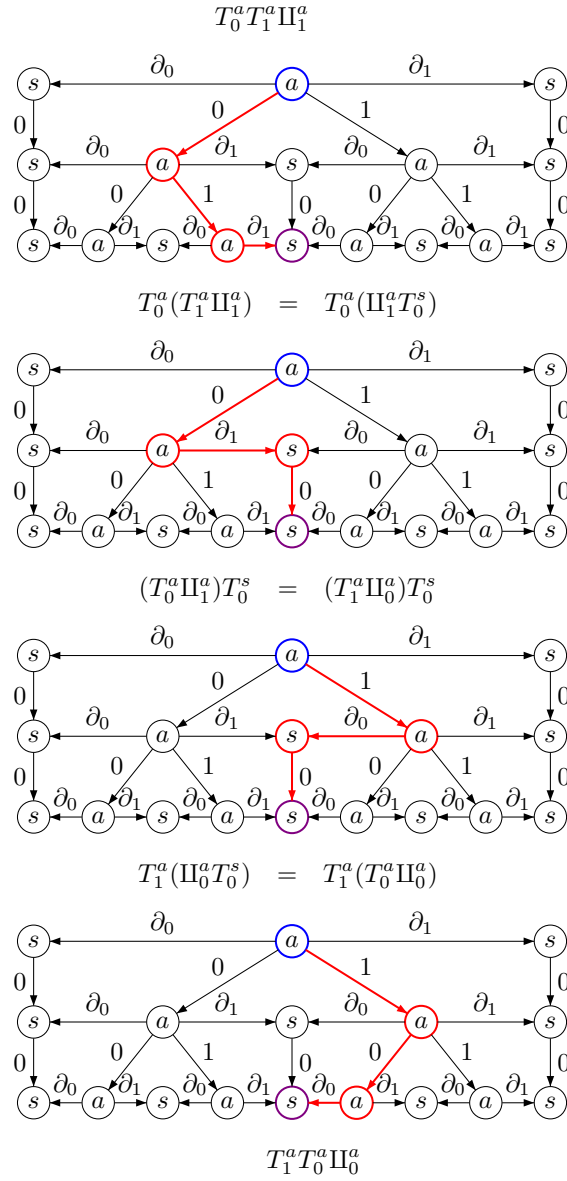


**Fig. 3.13** – Graphe quotient

Par induction, on remarque que la structure reste raccordée à tous les niveaux d'itération suivants, comme le montre la figure 3.14 au second niveau.



**Fig. 3.14** – Graphe quotient au deuxième niveau de subdivision



**Fig. 3.15** – Équivalence de chemins de deuxième niveau de subdivision, déduite des équivalences de premier niveau

Par exemple, l'équivalence

$$T_0^a T_1^a \Pi_1^a = T_1^a T_0^a \Pi_0^a$$

est implicitement vérifiée, car elle peut se déduire des trois équivalences que l'on a données (voir figure 3.15).

On peut conclure, par passage à la limite, que l'attracteur  $\mathcal{A}^a$  associé à l'état  $a$  est bien une courbe, sous réserve que les opérateurs de subdivision  $T_0^a$  et  $T_1^a$  soient contractants, et donc que l'attracteur  $\mathcal{A}^a$  existe et soit unique.

### 3.2.2 Carreau quadrangulaire

L'exemple suivant montre comment définir un BCIFS à topologie surfacique, en prenant l'exemple d'un carreau de surface à quatre subdivisions. On distinguera sur ce modèle deux types d'arêtes, selon leur orientation horizontale ou verticale sur la face. On note  $f$  l'état associé à la surface,  $a$  l'état associé aux arcs de courbes délimitant le carreau à gauche et à droite,  $b$  l'état associé aux arcs de courbes délimitant la surface en haut et en bas, et  $s$  l'état associé aux extrémités des courbes  $a$  et  $b$ .

#### Remarque

Du point de vue topologique, il n'est pas nécessaire de distinguer deux types de courbes  $a$  et  $b$ . On pourrait n'utiliser qu'un seul type de courbe pour chacun des quatre bords. Cependant, en distinguant ces deux types, on obtient un modèle de surface plus général, car d'un point de vue géométrique, les attracteurs  $\mathcal{A}^a$  et  $\mathcal{A}^b$  peuvent être définis indépendamment l'un de l'autre. Cela permet de modéliser des surfaces dont les bords verticaux sont d'un aspect différent des bords horizontaux (voir figure 4.35).

#### Description BCIFS du carreau

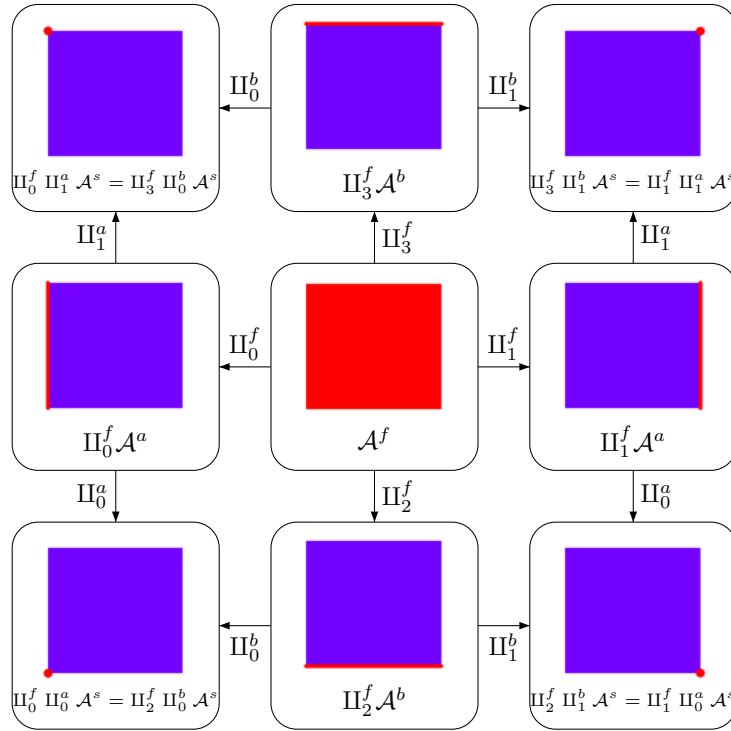


Fig. 3.16 – Face, arêtes et sommets d'un carreau définis par plongements

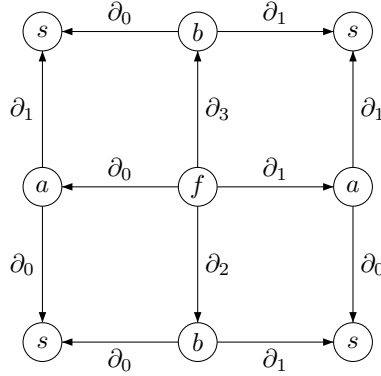
On établit dans un premier temps les relations d'incidence entre  $f$ ,  $a$ ,  $b$  et  $s$ .

$$\begin{aligned}
 \delta(f, \partial_0) &= a & \delta(a, \partial_0) &= s & \delta(b, \partial_0) &= s \\
 \delta(f, \partial_1) &= a & \delta(a, \partial_1) &= s & \delta(b, \partial_1) &= s \\
 \delta(f, \partial_2) &= b & & & & \\
 \delta(f, \partial_3) &= b & & & & 
 \end{aligned}$$

Les opérateurs d'incidence  $(\Pi_i^f)_{i \in \{0,1,2,3\}}$  vérifient les équations d'adjacence :

$$\begin{aligned} \Pi_0^f \Pi_1^a &= \Pi_3^f \Pi_0^b & \Pi_1^f \Pi_1^a &= \Pi_3^f \Pi_1^b \\ \Pi_0^f \Pi_0^a &= \Pi_2^f \Pi_0^b & \Pi_1^f \Pi_0^a &= \Pi_2^f \Pi_1^b \end{aligned}$$

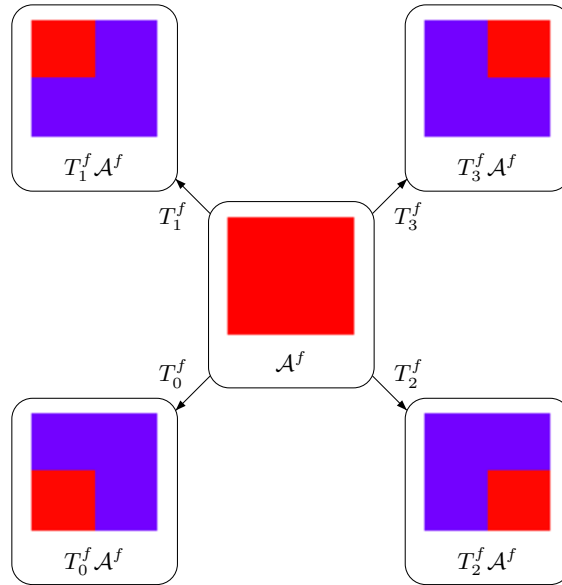
D'après ces équivalences de chemins sur les bordures, on obtient un graphe quotient représenté sur la figure 3.17.



**Fig. 3.17** – Graphe quotient sur les incidences

On déclare également dans l'automate quatre relations de subdivision pour  $f$ , deux relations de subdivision pour  $a$  et  $b$ , et une pour  $s$ .

$$\begin{aligned} \delta(f, 0) &= f & \delta(a, 0) &= a & \delta(b, 0) &= b & \delta(s, 0) &= s \\ \delta(f, 1) &= f & \delta(a, 1) &= a & \delta(b, 1) &= b & & \\ \delta(f, 2) &= f & & & & & & \\ \delta(f, 3) &= f & & & & & & \end{aligned}$$



**Fig. 3.18** – Structure de subdivision d'un carreau

L'automate contenant les relations de bordure ainsi que les relations de subdivision est donné sur la figure 3.19.

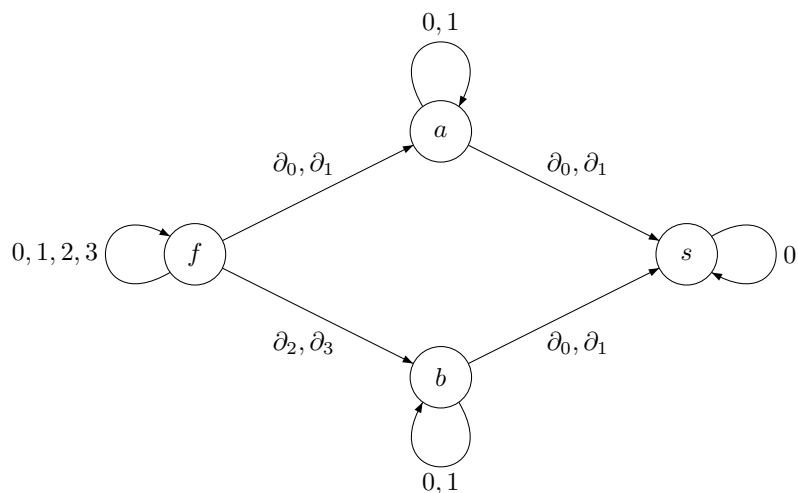


Fig. 3.19 – Automate complet

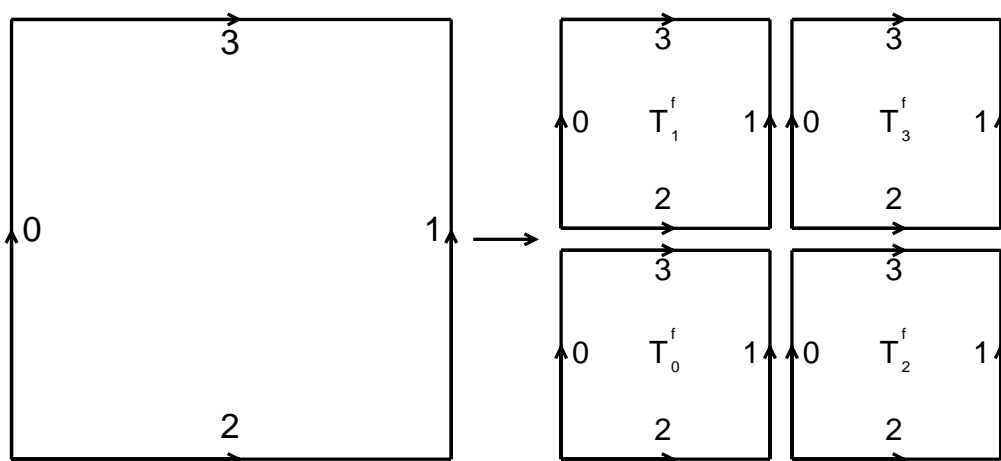
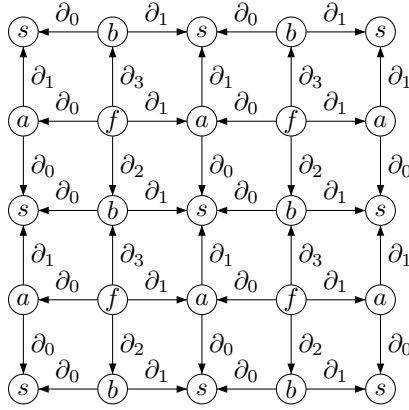


Fig. 3.20 – Orientations des bords et des subdivisions de  $f$

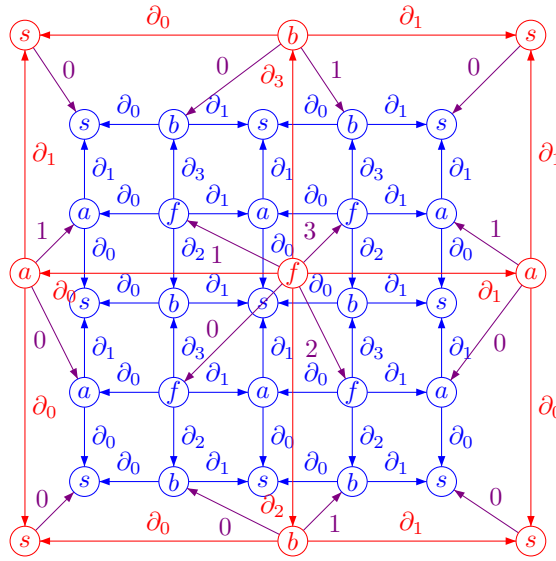
Nous allons ensuite établir les équivalences de chemins faisant intervenir les subdivisions des cellules. La structure de raccord donnée par la figure 3.20 s'exprime dans le graphe quotient de la figure 3.21.





**Fig. 3.21** – Structure de raccord du premier niveau de subdivision

Certaines cellules de cette structure de premier niveau de subdivision correspondent à l'image, par un opérateur de subdivision, d'une cellule de niveau zéro. On représente sur la figure 3.22 les relations de subdivision qui lient les cellules de niveau zéro avec celles de niveau un.



**Fig. 3.22** – Mise en correspondance entre la structure de raccord initiale (en rouge) et celle du premier niveau de subdivision (en bleu)

Nous décrivons maintenant la liste des équivalences de chemins nécessaires pour obtenir une telle structure topologique. Nous avons quatre équations d'adjacence, établissant la mise en commun d'arêtes entre les faces subdivisées.

$$\begin{aligned}
 T_0^f \Pi_3^f &= T_1^f \Pi_2^f & T_1^f \Pi_1^f &= T_3^f \Pi_0^f & T_2^f \Pi_3^f &= T_3^f \Pi_2^f \\
 & & T_0^f \Pi_1^f &= T_2^f \Pi_0^f & &
 \end{aligned}$$

Nous avons également un ensemble d'équations d'incidence, qui identifient la subdivision d'une arête avec l'arête d'une face subdivisée.

$$\begin{array}{llll}
T_1^f \Pi_0^f = \Pi_0^f T_1^a & T_1^f \Pi_3^f = \Pi_3^f T_0^b & T_3^f \Pi_3^f = \Pi_3^f T_1^b & T_3^f \Pi_1^f = \Pi_1^f T_1^a \\
T_0^f \Pi_0^f = \Pi_0^f T_0^a & T_0^f \Pi_2^f = \Pi_2^f T_0^b & T_2^f \Pi_2^f = \Pi_2^f T_1^b & T_2^f \Pi_1^f = \Pi_1^f T_0^a
\end{array}$$

Enfin, on note également les contraintes décrivant une structure topologique de courbe pour les cellules  $a$  et  $b$ .

$$\begin{array}{lll}
\Pi_0^a T_0^s = T_0^a \Pi_0^a & T_0^a \Pi_1^a = T_1^a \Pi_0^a & \Pi_1^a T_0^s = T_1^a \Pi_1^a \\
\Pi_0^b T_0^s = T_0^b \Pi_0^b & T_0^b \Pi_1^b = T_1^b \Pi_0^b & \Pi_1^b T_0^s = T_1^b \Pi_1^b
\end{array}$$

### 3.2.3 Carreau triangulaire

Un autre exemple de BCIFS à topologie surfacique est celui d'un triangle à quatre subdivisions. On note  $f$  l'état associé à la surface,  $a$ ,  $b$  et  $c$  l'état associé à chaque courbe délimitant la surface, et  $s$  l'état associé aux extrémités des courbes.

On définit dans un premier temps les relations d'incidence entre  $f$ ,  $a$ ,  $b$ ,  $c$  et  $s$ .

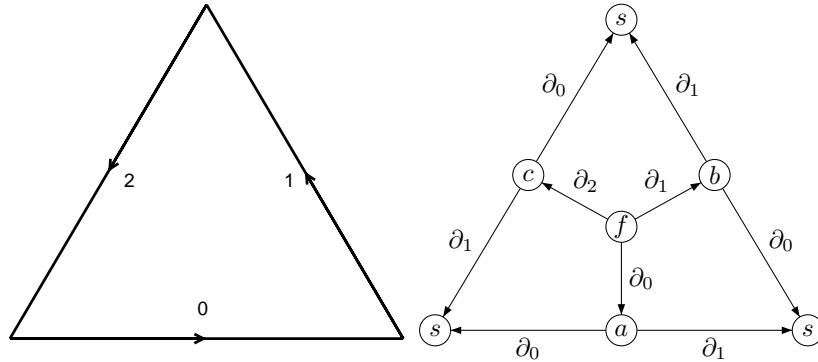
$$\begin{array}{llll}
\delta(f, \partial_0) = a & \delta(a, \partial_0) = s & \delta(b, \partial_0) = s & \delta(c, \partial_0) = s \\
\delta(f, \partial_1) = b & \delta(a, \partial_1) = s & \delta(b, \partial_1) = s & \delta(c, \partial_1) = s \\
\delta(f, \partial_2) = c & & & 
\end{array}$$

pour spécifier qu'une face de type  $f$  est bordée par trois arêtes de type  $a$ ,  $b$  et  $c$ , chacune bordée par deux sommets de type  $s$ .

On peut d'ores et déjà établir une structure de raccord pour les bordures de la cellule  $f$ . Les opérateurs d'incidence  $(\Pi_i^f)_{i=\{0,1,2\}}$  vérifient les équations d'adjacence :

$$\begin{array}{l}
\Pi_1^f \Pi_1^b = \Pi_2^f \Pi_0^c \\
\Pi_2^f \Pi_1^c = \Pi_0^f \Pi_0^a \quad \Pi_0^f \Pi_1^a = \Pi_1^f \Pi_0^b
\end{array}$$

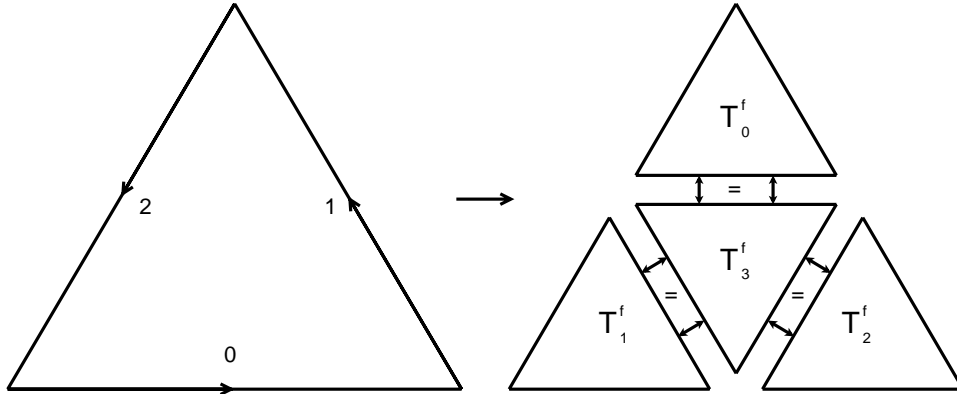
D'après ces équivalences de chemins sur les bordures, on obtient un graphe quotient représenté sur la figure 3.23.



**Fig. 3.23** – Graphe quotient, exprimant la structure de raccord des arêtes d'un carreau triangulaire

On déclare également dans l'automate quatre relations de subdivision pour  $f$  (voir figure 3.24), et une relation de subdivision pour  $s$ . On ne précise pas encore les relations de subdivision des arêtes  $a$ ,  $b$  et  $c$ . On les déterminera après avoir résolu certains problèmes d'orientation des faces, et défini la structure de raccord du premier niveau de subdivision.

$$\begin{aligned}
\delta(f, 0) &= f & \delta(s, 0) &= s \\
\delta(f, 1) &= f \\
\delta(f, 2) &= f \\
\delta(f, 3) &= f
\end{aligned}$$

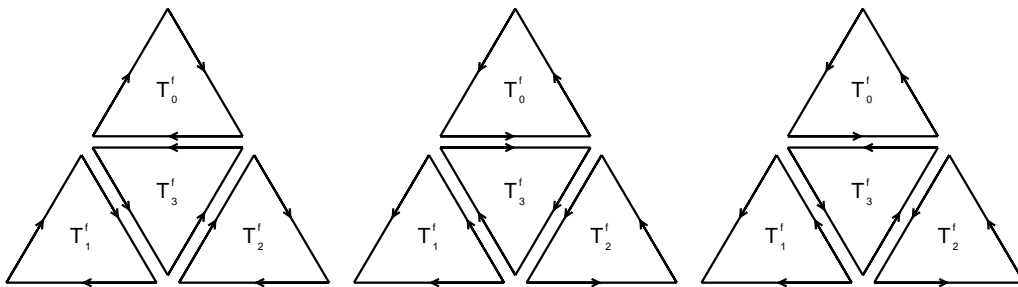


**Fig. 3.24** – Relations de subdivision du carreau triangulaire en quatre sous-carreaux triangulaires

### Choix de la structure de raccord de premier niveau de subdivision

Nous allons ensuite définir la structure de raccord du premier niveau de subdivision. Il est nécessaire pour cela de faire un certain nombre de choix concernant l'orientation des faces et de arêtes, car il existe de multiples façons de former un triangle par quatre subdivisions. Nous détaillons ici la description d'un exemple parmi d'autres, en illustrant brièvement pour chaque choix que l'on fait certaines alternatives possibles. Le premier choix concerne l'orientation des arêtes, et le second choix l'orientation des faces.

Il existe de multiples manières d'orienter les arêtes des faces subdivisées. La figure 3.25 en montre trois exemples.



**Fig. 3.25** – Exemples d'orientations des arêtes possibles au premier niveau de subdivision. On développe par la suite le premier exemple.

Nous choisissons d'orienter les arêtes de manière à ce que chaque couple d'arêtes mises en commun soient orientées dans le même sens, en retournant les trois faces de subdivision situées aux extrémités, comme le montre la figure 3.26. Les structures de raccord avec des bordures orientées différemment, telles que le troisième exemple de la figure 3.25 sont un peu plus compliquées à décrire dans le formalisme BCIFS, et nous aborderons ce type de problèmes dans la partie 3.3.

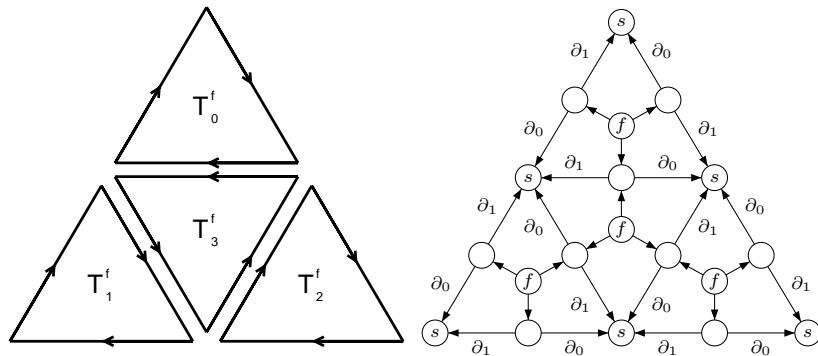


Fig. 3.26 – Structure de raccord avec la même orientation des arêtes mises en commun, obtenue en retournant les trois faces aux extrémités

On précise maintenant l'orientation des faces que l'on choisit, de manière à ce que les arêtes mises en commun soient de même type  $a$ ,  $b$  ou  $c$ . Cela peut se faire de trois manières différentes (voir figure 3.27).

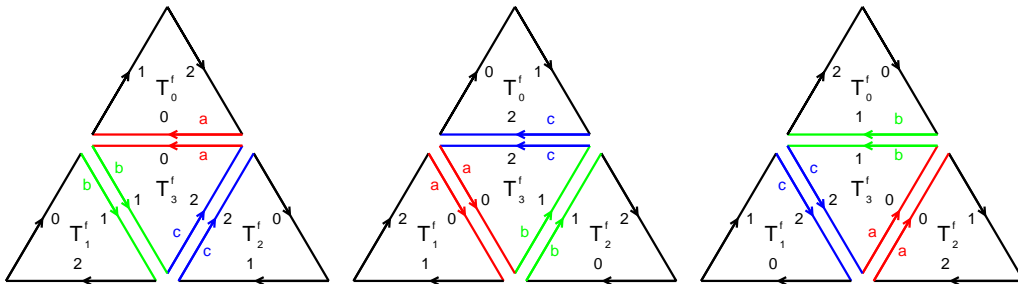


Fig. 3.27 – Orientations des faces possibles au premier niveau de subdivision. On développe par la suite le premier exemple.

Nous choisissons la première de ces trois possibilités. Ceci détermine alors la structure de raccord du premier niveau de subdivision (voir figure 3.28).

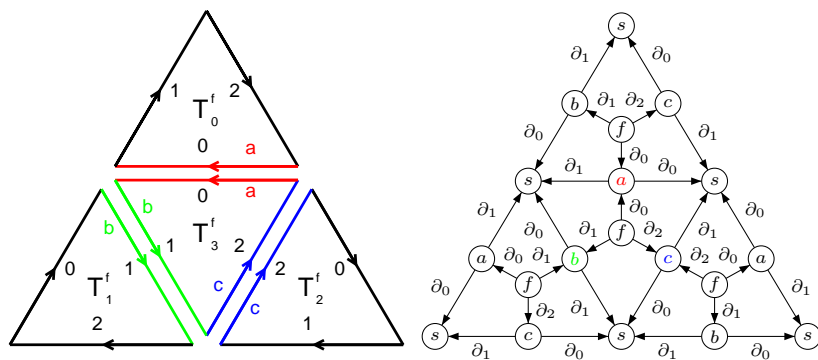


Fig. 3.28 – Structure de raccord du premier niveau de subdivision

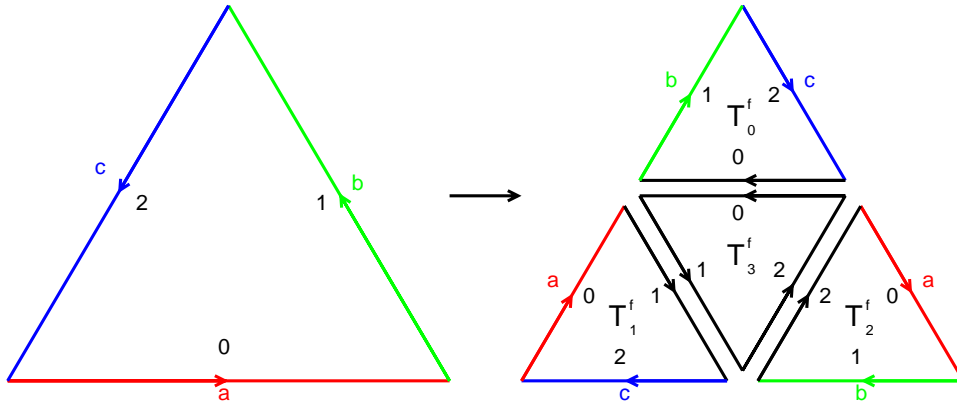
## Description BCIFS

Nous décrivons maintenant la liste des équivalences de chemins nécessaires pour obtenir la structure topologique définie sur la figure 3.28. Ces équivalences peuvent se regrouper en plusieurs familles. Nous avons d'abord un ensemble de trois équations d'adjacence, correspondant à la mise en commun d'arêtes entre les faces subdivisées adjacentes.

$$\begin{aligned} T_0^f \Pi_0^f &= T_3^f \Pi_0^f \\ T_1^f \Pi_1^f &= T_3^f \Pi_1^f & T_2^f \Pi_2^f &= T_3^f \Pi_2^f \end{aligned}$$

Nous avons également un ensemble d'équations d'incidence, qui identifient la subdivision d'une arête avec l'arête d'une face subdivisée. On note que d'après les orientations des faces que l'on a définies, les arêtes  $a$ ,  $b$  et  $c$  ne sont pas des courbes autosimilaires mais interdépendantes, c'est-à-dire qu'elles correspondent à un CIFS (voir figure 3.29).

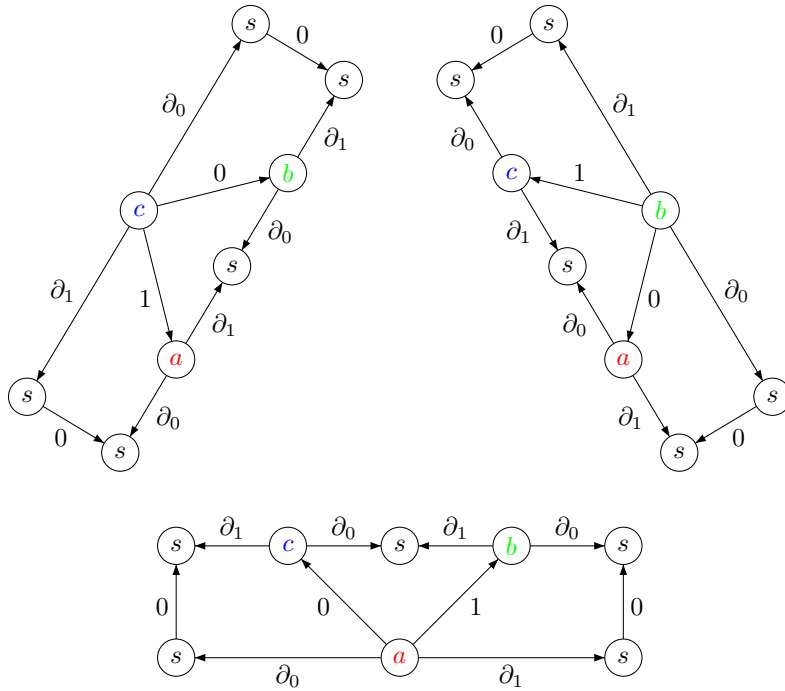
$$\begin{aligned} T_1^f \Pi_2^f &= \Pi_0^f T_0^a & T_2^f \Pi_1^f &= \Pi_0^f T_1^a \\ T_2^f \Pi_0^f &= \Pi_1^f T_0^b & T_0^f \Pi_2^f &= \Pi_1^f T_1^b \\ T_0^f \Pi_1^f &= \Pi_2^f T_0^c & T_1^f \Pi_0^f &= \Pi_2^f T_1^c \end{aligned}$$



**Fig. 3.29** – Mises en correspondance entre les arêtes de niveau 0 et de niveau 1, permettant d'établir les relations d'incidence de  $f$ , ainsi que la structure de subdivision des arêtes  $a$ ,  $b$  et  $c$

On déclare enfin les contraintes décrivant la structure topologique des courbes pour les cellules  $a$ ,  $b$  et  $c$  (voir 3.30).

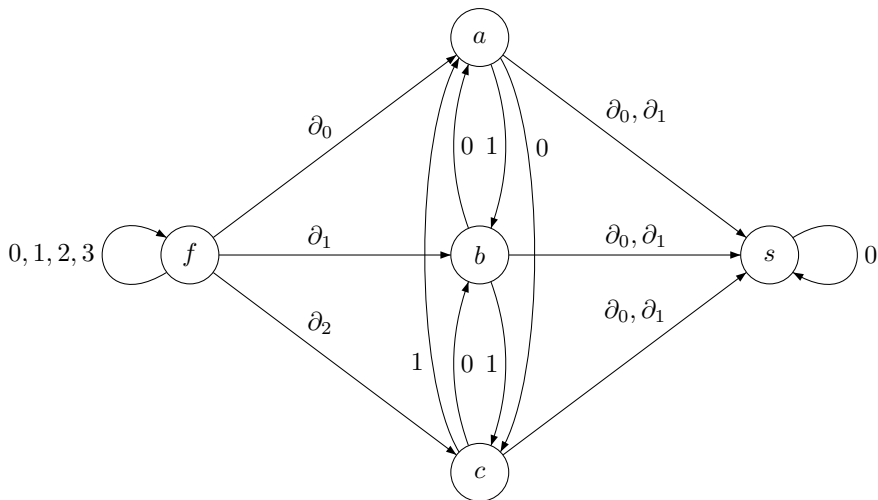
$$\begin{aligned} \Pi_0^a T_0^s &= T_0^a \Pi_1^c & T_0^a \Pi_0^c &= T_1^a \Pi_1^b & \Pi_1^a T_0^s &= T_1^a \Pi_0^b \\ \Pi_0^b T_0^s &= T_0^b \Pi_1^a & T_0^b \Pi_0^a &= T_1^b \Pi_1^c & \Pi_1^b T_0^s &= T_1^b \Pi_0^c \\ \Pi_0^c T_0^s &= T_0^c \Pi_1^b & T_0^c \Pi_0^b &= T_1^c \Pi_1^a & \Pi_1^c T_0^s &= T_1^c \Pi_0^a \end{aligned}$$



**Fig. 3.30** – Structure topologique des courbes  $a$ ,  $b$  et  $c$

L'automate contenant les relations de bordure ainsi que les relations de subdivision est donné sur la figure 3.31. Nous n'avons pas décrit l'automate avant car les règles de subdivision des courbes  $a$ ,  $b$  et  $c$  ont été déterminées par la structure de raccord de niveau un vue précédemment.

$$\begin{array}{lll} \delta(a,0) = c & \delta(b,0) = a & \delta(c,0) = b \\ \delta(a,1) = b & \delta(b,1) = c & \delta(c,1) = a \end{array}$$



**Fig. 3.31** – Automate complet

### 3.3 Problèmes d'orientation et de symétrie

Dans l'exemple précédent, nous avons évoqué la possibilité de définir des structures de raccord avec des arêtes mises en commun orientées différemment les unes des autres, telle que sur la figure 3.35.

Pour définir une telle structure dans le formalisme BCIFS, on introduit un nouveau type d'opérateur : les opérateurs de symétrie. Cet opérateur est une involution qui inverse l'orientation d'une arête en permutant chacun de ses deux sommets.

On introduit également une contrainte de symétrie. Cette contrainte permet d'établir qu'une arête est confondu avec son image par l'opérateur de symétrie défini précédemment. Ainsi, il est possible de raccorder deux faces adjacentes en mettant en commun deux arêtes orientées différemment l'une de l'autre.

Enfin, on donne la description BCIFS complète d'un triangle avec mises en commun d'arêtes orientées différemment, en utilisant des opérateurs et des contraintes de symétries présentées précédemment.

#### 3.3.1 Opérateurs de symétrie

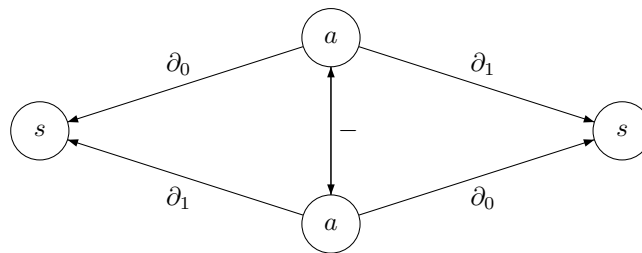
On définit l'opérateur  $S_-^a : E^a \leftarrow E^a$  comme une involution ( $S_-^a S_-^a = Id$ ) qui vérifie les propriétés suivantes :

$$S_-^a \Pi_1^a = \Pi_0^a \quad S_-^a \Pi_0^a = \Pi_1^a$$

Comme  $S_-^a$  est une involution, ces deux équations sont équivalentes :

$$\begin{aligned} S_-^a \Pi_1^a &= \Pi_0^a \\ S_-^a S_-^a \Pi_1^a &= S_-^a \Pi_0^a \\ \Pi_1^a &= S_-^a \Pi_0^a \end{aligned}$$

L'opérateur  $S_-^a$  est associé dans l'automate au symbole  $-$ . Les deux équations sur  $S_-^a$  s'expriment sous la forme d'un graphe quotient illustré sur la figure 3.32.



**Fig. 3.32** – Définition de l'involution  $S_-^a$ , inversant l'orientation d'une arête  $a$

### 3.3.2 Contraintes de symétrie

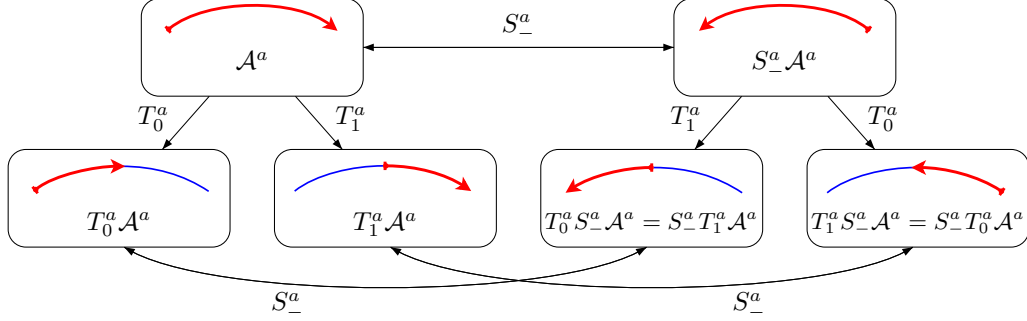


Fig. 3.33 – Contraintes de symétrie de l'arête  $a$

On établit enfin des contraintes permettant d'établir la mise en commun entre une arête avec son arête inversée. Les courbes doivent vérifier une propriété de symétrie, à savoir  $\mathcal{A}^a = S_-^a \mathcal{A}^a$ . Cette condition est vérifiée si les opérateurs de subdivision de la courbe  $T_0^a$  et  $T_1^a$  vérifient :

$$T_0^a S_-^a = S_-^a T_1^a \quad T_1^a S_-^a = S_-^a T_0^a$$

car on a alors :

$$S_-^a \mathcal{A}^a = S_-^a (T_0^a \mathcal{A}^a \cup T_1^a \mathcal{A}^a) = S_-^a T_0^a \mathcal{A}^a \cup S_-^a T_1^a \mathcal{A}^a = T_1^a S_-^a \mathcal{A}^a \cup T_0^a S_-^a \mathcal{A}^a$$

$$S_-^a \mathcal{A}^a \in \mathcal{H}(E^a), \quad S_-^a \mathcal{A}^a = T_0^a (S_-^a \mathcal{A}^a) \cup T_1^a (S_-^a \mathcal{A}^a) \Leftrightarrow S_-^a \mathcal{A}^a = \mathcal{A}^a$$

D'autre part, comme  $S_-^a$  est une involution, les deux équations  $T_0^a S_-^a = S_-^a T_1^a$  et  $T_1^a S_-^a = S_-^a T_0^a$  sont équivalentes :

$$\begin{aligned} T_0^a S_-^a &= S_-^a T_1^a \\ S_-^a (T_0^a S_-^a) S_-^a &= S_-^a (S_-^a T_1^a) S_-^a \\ (S_-^a T_0^a) (S_-^a S_-^a) &= (S_-^a S_-^a) (T_1^a S_-^a) \\ S_-^a T_0^a &= T_1^a S_-^a \end{aligned}$$

La symétrie de  $a$  est donc établie par une seule contrainte (voir figure 3.34).

$$T_0^a S_-^a = S_-^a T_1^a$$

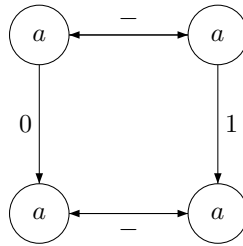
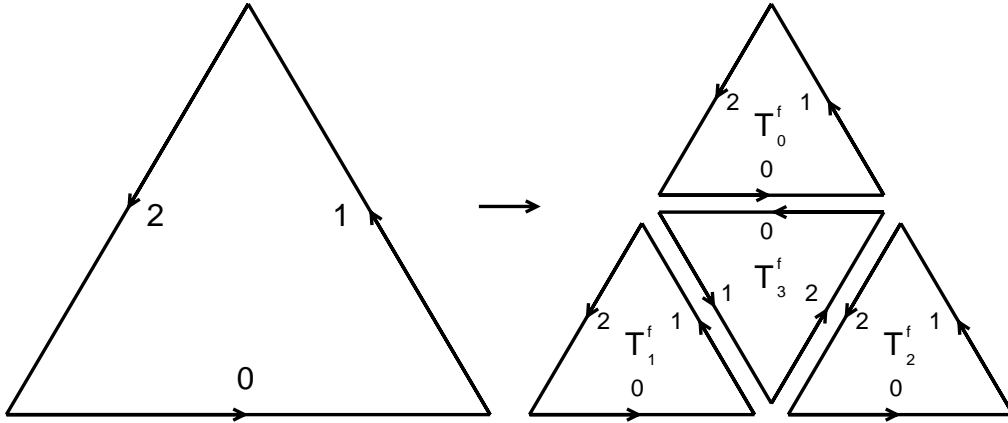


Fig. 3.34 – Contrainte de symétrie de la courbe  $a$



### 3.3.3 Exemple du carreau triangulaire

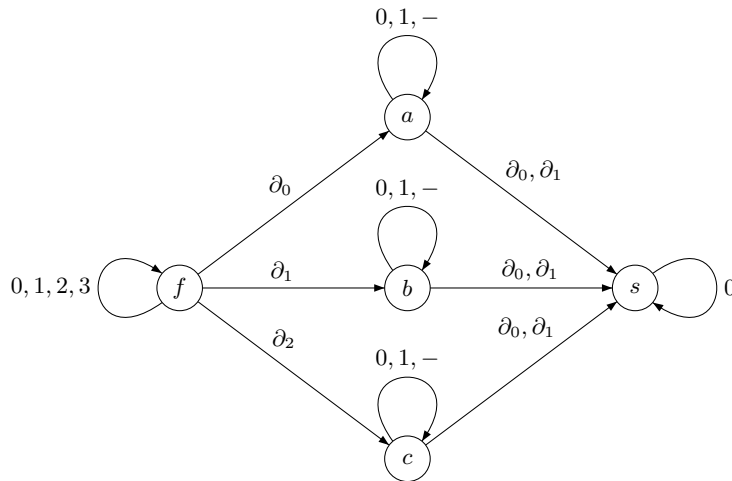
En utilisant ce qui précède, on peut décrire dans le formalisme BCIFS un carreau triangulaire dont les arêtes mises en commun sont orientées différemment (voir figure 3.35).



**Fig. 3.35** – Structure de raccord dont les arêtes mises en commun sont orientées différemment

Sur cet exemple, les arcs sont autosimilaires : les arêtes  $a$  se subdivisent en  $a$ , même chose pour  $b$  et  $c$ . La figure 3.36 décrit l'automate BCIFS correspondant.

$$\begin{aligned} \delta(f, 0) &= a & \delta(a, 0) &= a & \delta(b, 0) &= b & \delta(c, 0) &= c & \delta(s, 0) &= s \\ \delta(f, 1) &= b & \delta(a, 1) &= a & \delta(b, 1) &= b & \delta(c, 1) &= c & & \\ \delta(f, 2) &= c & & & & & & & & \\ \delta(f, 3) &= d & & & & & & & & \end{aligned}$$



**Fig. 3.36** – Automate

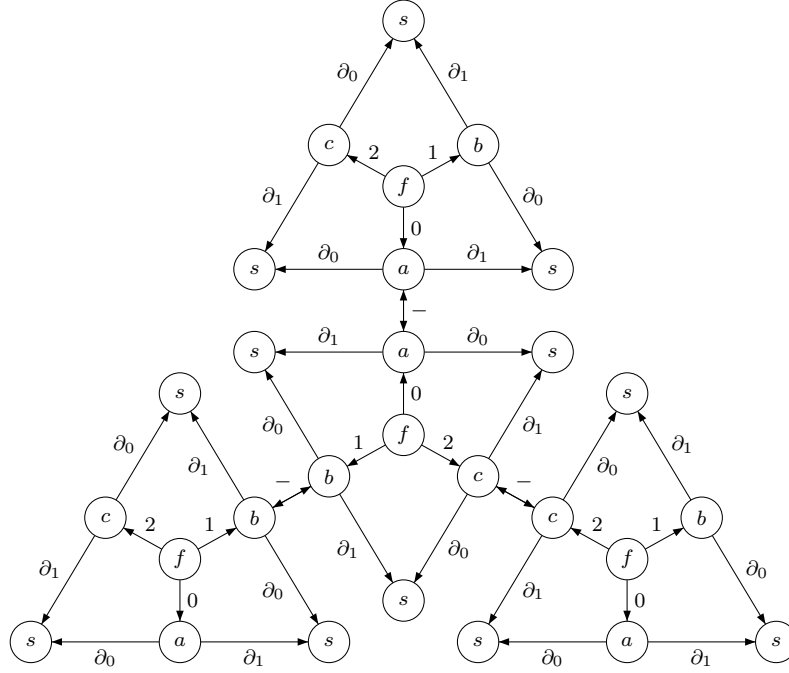
Chaque arête  $a$ ,  $b$  et  $c$  est munie d'un opérateur de symétrie, qui est une involution vérifiant :

$$S^a \Pi_1^a = \Pi_0^a \quad S^b \Pi_1^b = \Pi_0^b \quad S^c \Pi_1^c = \Pi_0^c$$

On établit une contrainte de symétrie sur chaque courbe :

$$T_0^a S_-^a = S_-^a T_1^a \quad T_0^b S_-^b = S_-^b T_1^b \quad T_0^c S_-^c = S_-^c T_1^c$$

Les opérateurs de symétrie permettent d'établir la structure de raccord de premier niveau du triangle (voir figure 3.37).



**Fig. 3.37** – Structure de raccord avec mise en commun d'arêtes orientées différemment

Les raccords sur les arêtes sont alors définis par les équations d'adjacence suivantes :

$$T_0^f \Pi_0^f = T_3^f \Pi_0^f S_-^a$$

$$T_1^f \Pi_1^f = T_3^f \Pi_1^f S_-^b \quad T_2^f \Pi_2^f = T_3^f \Pi_2^f S_-^c$$

Ces équations d'adjacence établissent la mise en commun entre une arête et son arête symétrique : le bord d'une face est égal au symétrique du bord d'une autre face. Ils coïncident en raison des contraintes de symétrie des courbes.

On établit les équations d'incidence de  $f$  :

$$\begin{aligned} T_1^f \Pi_0^f &= \Pi_0^f T_0^a & T_2^f \Pi_0^f &= \Pi_0^f T_1^a \\ T_2^f \Pi_1^f &= \Pi_1^f T_0^b & T_0^f \Pi_1^f &= \Pi_1^f T_1^b \\ T_0^f \Pi_2^f &= \Pi_2^f T_0^c & T_1^f \Pi_2^f &= \Pi_2^f T_1^c \end{aligned}$$

On déclare enfin les contraintes décrivant la structure topologique des courbes pour les cellules  $a$ ,  $b$  et  $c$ .

$$\begin{aligned} \Pi_0^a T_0^s &= T_0^a \Pi_0^a & T_0^a \Pi_1^a &= T_1^a \Pi_0^a & \Pi_1^a T_0^s &= T_1^a \Pi_1^a \\ \Pi_0^b T_0^s &= T_0^b \Pi_0^b & T_0^b \Pi_1^b &= T_1^b \Pi_0^b & \Pi_1^b T_0^s &= T_1^b \Pi_1^b \\ \Pi_0^c T_0^s &= T_0^c \Pi_0^c & T_0^c \Pi_1^c &= T_1^c \Pi_0^c & \Pi_1^c T_0^s &= T_1^c \Pi_1^c \end{aligned}$$

## 3.4 Topologies fractales

Dans les exemples qui suivent, les faces ne sont plus homéomorphes à des disques, mais les arêtes restent homéomorphes à l'intervalle  $[0, 1]$ . Dans le premier exemple, les opérateurs d'incidence peuvent encore s'interpréter comme des opérateurs de bord externe. Dans le second, ce n'est plus le cas.

### 3.4.1 Triangle de Sierpinski

L'exemple suivant montre comment définir un BCIFS dont la structure topologique correspond à celle d'un triangle de SIERPINSKI, qui a été présenté dans la partie 3.1.1. Nous verrons que les propriétés du BCIFS sont assez similaires à celles du triangle classique, décrit dans la partie 3.2.3, à la différence près que l'on supprime une face et ses équations d'adjacence.

On note  $f$  l'état associé à la face,  $a$  l'état associé aux arêtes, et  $s$  l'état associé aux sommets.

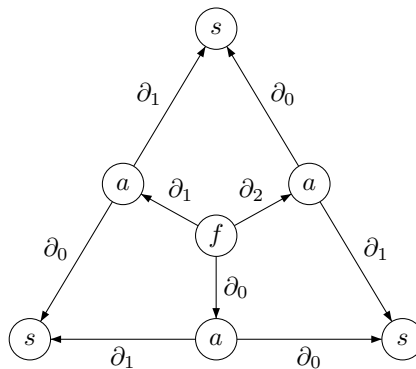
La structure de raccord de ses arêtes est la même que celle des carreaux triangulaires.

$$\begin{aligned} \delta(f, \partial_0) &= a & \delta(a, \partial_0) &= s \\ \delta(f, \partial_1) &= a & \delta(a, \partial_1) &= s \\ \delta(f, \partial_2) &= a & & \end{aligned}$$

Les opérateurs d'incidence  $(\Pi_i^f)_{i \in \{0,1,2\}}$  vérifient les équations d'adjacence :

$$\begin{aligned} \Pi_0^f \Pi_1^a &= \Pi_1^f \Pi_0^a \\ \Pi_1^f \Pi_1^a &= \Pi_2^f \Pi_1^b \\ \Pi_2^f \Pi_1^a &= \Pi_0^f \Pi_0^a \end{aligned}$$

D'après ces équivalences de chemins sur les bordures, on obtient le graphe quotient représenté sur la figure 3.38.

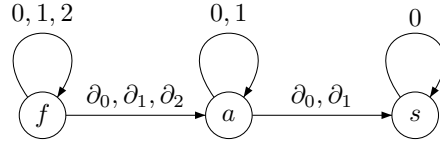


**Fig. 3.38** – Graphe quotient, selon les relations de bordures uniquement

On déclare également dans l'automate quatre relations de subdivision pour  $f$ , deux relations de subdivision pour  $a$ , et une pour  $s$ .

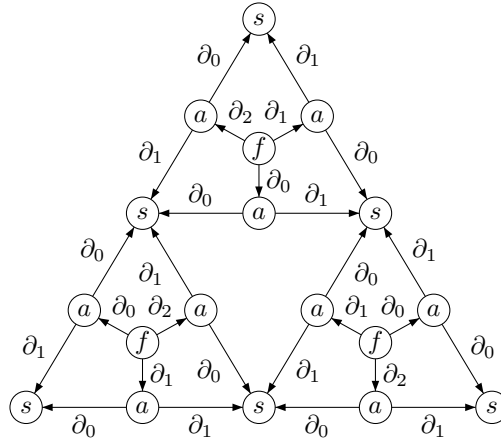
$$\begin{aligned} \delta(f, 0) &= f & \delta(a, 0) &= a & \delta(s, 0) &= s \\ \delta(f, 1) &= f & \delta(a, 1) &= a & & \\ \delta(f, 2) &= f & & & & \end{aligned}$$

L'automate comprenant les symboles d'incidence et de subdivision est donné sur la figure 3.39.



**Fig. 3.39** – Automate complet

Nous allons ensuite établir les équivalences de chemins faisant intervenir les subdivisions des cellules. La structure de raccord du premier niveau de subdivision est représentée sur la figure 3.40.



**Fig. 3.40** – Structure de raccord du premier niveau de subdivision

Nous décrivons maintenant les règles d'équivalence de chemins nécessaires pour obtenir une telle structure topologique. Contrairement au triangle classique, nous n'avons plus ici d'équations d'adjacence entre les faces subdivisées.

Nous avons un ensemble d'équations d'incidence, qui mettent en correspondance la subdivision d'une arête avec l'arête d'une face subdivisée.

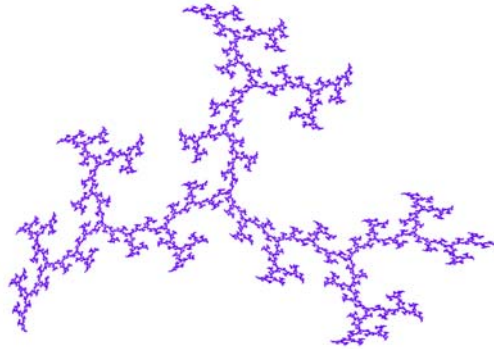
$$\begin{aligned}
 T_1^f \Pi_1^f &= \Pi_0^f T_0^a & T_2^f \Pi_2^f &= \Pi_0^f T_1^a \\
 T_2^f \Pi_0^f &= \Pi_1^f T_0^b & T_0^f \Pi_1^f &= \Pi_1^f T_1^b \\
 T_0^f \Pi_2^f &= \Pi_2^f T_0^c & T_1^f \Pi_0^f &= \Pi_2^f T_1^c
 \end{aligned}$$

Enfin, on déclare les contraintes décrivant la structure topologique d'une courbe pour la cellule  $a$ .

$$\Pi_0^a T_0^s = T_0^a \Pi_0^a \quad T_0^a \Pi_1^a = T_1^a \Pi_0^a \quad \Pi_1^a T_0^s = T_1^a \Pi_1^a$$

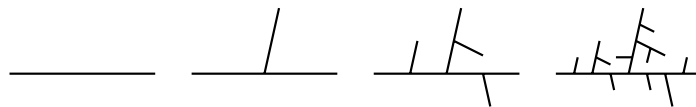
### 3.4.2 Arbre

Un autre exemple de topologie fractale est celui d'une structure arborescente (voir figure 3.41).



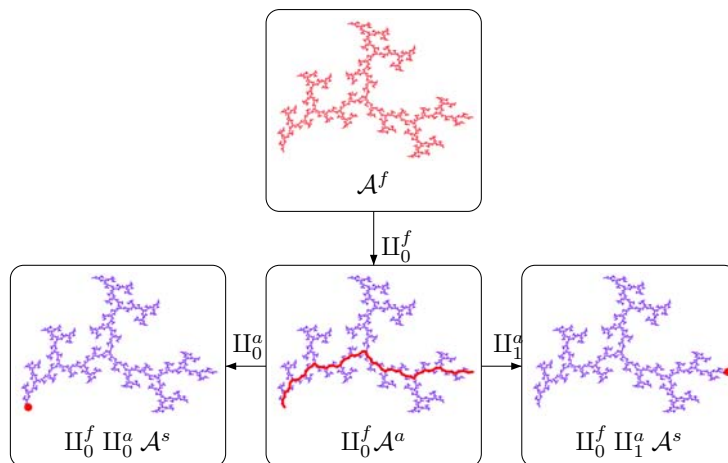
**Fig. 3.41** – Exemples d'attracteur  $\mathcal{A}^f$  ayant une structure d'arbre

De la même manière que le triangle de SIERPINSKI correspond à un triangle classique auquel on supprime une subdivision, on peut considérer l'exemple de l'arbre comme une courbe classique à laquelle on insère une troisième subdivision. Son procédé de construction peut se décrire dans un formalisme similaire aux L-Systèmes [ZTA08] [Mor96] [PH91] [Pru86] (voir figure 3.42).



**Fig. 3.42** – Étapes successives de construction d'un arbre

Cet arbre contient une arête incidente  $a$ , bordée par deux sommets  $s$  (voir figure 3.43). La dimension combinatoire de l'arbre est de 2, car il a une arête incidente de dimension combinatoire 1. On associe donc cet arbre à un état  $f$ .



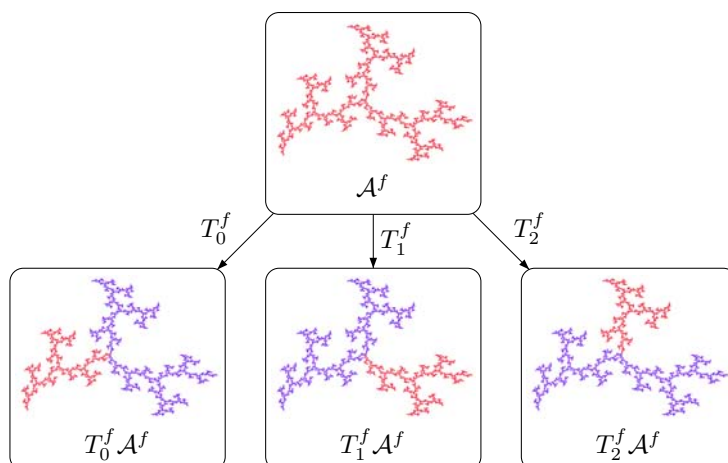
**Fig. 3.43** – Décomposition de l'arbre  $f$  en face, arêtes et sommets

On introduit les relations d'incidence entre  $f$ ,  $a$  et  $s$ .

$$\begin{aligned} \delta(f, \partial_0) &= a & \delta(a, \partial_0) &= s \\ & & \delta(a, \partial_1) &= s \end{aligned}$$

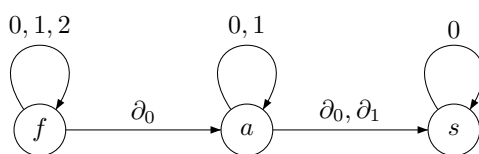
On introduit également dans l'automate les trois relations de subdivision de  $f$  (figure 3.44), deux pour  $a$  et une pour  $s$ .

$$\begin{aligned} \delta(f, 0) &= f & \delta(a, 0) &= a & \delta(s, 0) &= s \\ \delta(f, 1) &= f & \delta(a, 1) &= a & & \\ \delta(f, 2) &= f & & & & \end{aligned}$$



**Fig. 3.44** – Structure de subdivision de l'arbre  $f$

On obtient un automate contenant les relations d'incidence, ainsi que les relations de subdivision, comme le montre la figure 3.45.



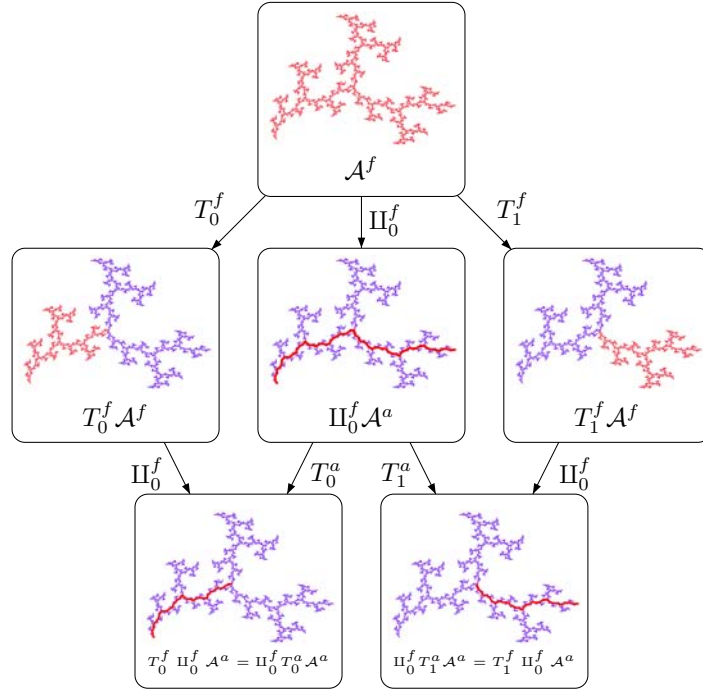
**Fig. 3.45** – Automate complet

On introduit également les équations d'incidence et d'adjacence décrivant une structure topologique de courbe pour les cellules  $a$ .

$$\Pi_0^a T_0^s = T_0^a \Pi_0^a \quad T_0^a \Pi_1^a = T_1^a \Pi_0^a \quad \Pi_1^a T_0^s = T_1^a \Pi_1^a$$

Nous déclarons maintenant les équations d'incidence de  $f$  (voir figure 3.46).

$$\Pi_0^f T_0^a = T_0^f \Pi_0^f \quad \Pi_0^f T_1^a = T_1^f \Pi_0^f$$



**Fig. 3.46** – Équations d'incidence mettant en correspondance le bord d'une subdivision avec la subdivision d'un bord

Enfin, on introduit une équation d'adjacence sur les subdivisions qui permet de raccorder  $T_2^f$  au sommet commun de  $T_0^f$  et  $T_1^f$  (voir figure 3.47).

$$T_0^f \Pi_0^f \Pi_1^a = T_1^f \Pi_0^f \Pi_0^a = T_2^f \Pi_0^f \Pi_1^a$$

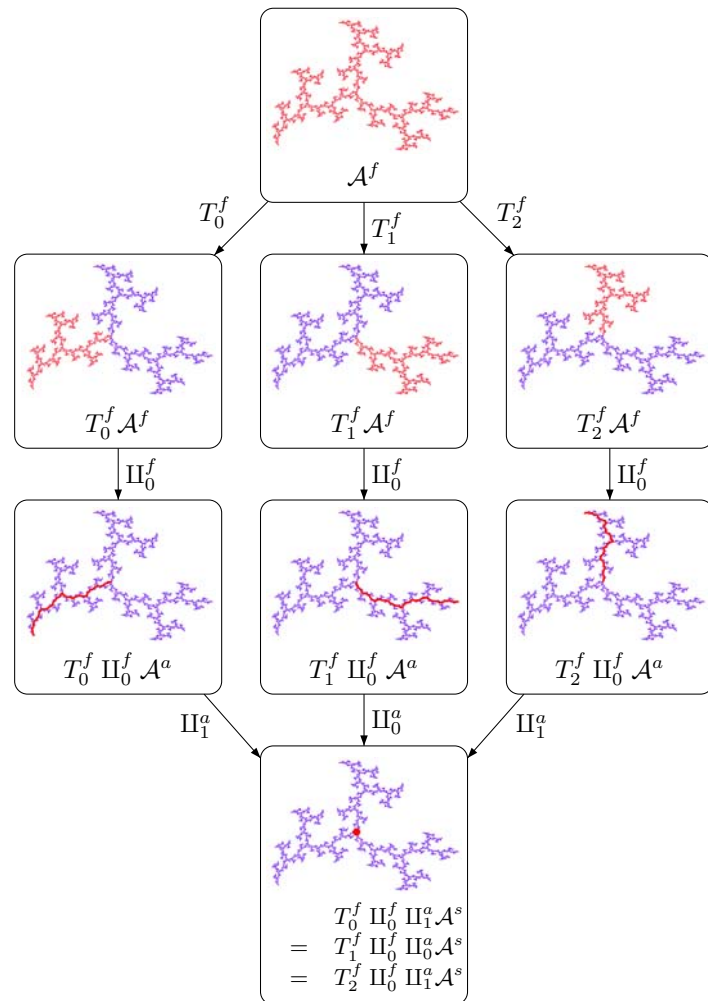
On remarque que l'équation

$$T_0^f \Pi_0^f \Pi_1^a = T_1^f \Pi_0^f \Pi_0^a$$

est implicitement vérifiée car elle peut se déduire de l'équation d'adjacence de  $a$  et des équations d'incidence de  $f$ , données précédemment.

$$T_0^f \Pi_0^f \Pi_1^a = \Pi_0^f T_0^a \Pi_1^a = \Pi_0^f T_0^a \Pi_1^a = T_1^f \Pi_0^f \Pi_0^a$$

Pour éviter les équations redondantes, on peut donc seulement raccorder  $T_2^f \Pi_0^f \Pi_1^a$  avec  $T_0^f \Pi_0^f \Pi_1^a$  ou bien avec  $T_1^f \Pi_0^f \Pi_0^a$ .



**Fig. 3.47** – Équations d'adjacence sur les subdivisions, raccordant un sommet de  $T_2^f$  avec le sommet du milieu



# Chapitre 4

## Plongement géométrique

### 4.1 Introduction

Après avoir décrit les propriétés formelles des BCIFS, nous nous intéressons ici à leur plongement géométrique. Les BCIFS sont décrits par un automate muni de relations d'équivalence sur les chemins. Le plongement géométrique des BCIFS correspond au fait que les transitions de l'automate sont associées à des opérateurs géométriques. Cela nécessite également d'avoir une interprétation géométrique des espaces  $E^x$  associés aux états de l'automate.

#### 4.1.1 Conventions

Nous donnons tout d'abord un certain nombre de conventions, permettant une interprétation géométrique des BCIFS proche de la formulation des formes à pôles ou des schémas de subdivision.

##### Choix des espaces

À chaque état de l'automate, on fait correspondre un certain espace. Nous affecterons à l'état initial  $\natural$  un type d'espace différent que pour les autres états.

L'espace  $E^\natural$  associé à l'état initial  $\natural$  est l'espace dans lequel les objets modélisés seront définis. Par commodité, on suppose que cet espace est un espace affine bidimensionnel ou tridimensionnel,  $E^\natural = \mathbb{R}^2$  ou  $\mathbb{R}^3$ , ou éventuellement un espace projectif  $P(\mathbb{R}^4)$ .

Hormis pour l'état initial  $\natural$ , on associera à chaque état  $x$  un espace barycentrique  $E^x = B(\mathbb{R}^{J^x})$ , où  $J^x$  est un ensemble fini d'indices associés aux points de contrôle. En pratique, on prend  $J^x = \{0, \dots, m^x - 1\}$ , où  $m^x$  désigne le nombre de points de contrôle associés à l'état  $x$ .

##### Interprétation géométrique des opérateurs

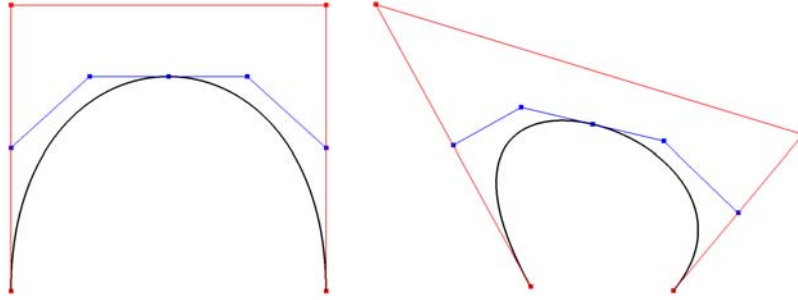
À chaque transition  $\delta(x, i) = y$  dans l'automate correspond un opérateur géométrique  $T_i^x : E^x \leftarrow E^y$ . On représente chaque opérateur par une matrice homogène.

Le choix des espaces associés aux états a des conséquences sur l'interprétation géométrique des opérateurs et sur leur représentation matricielle.

De la même manière que nous avons distingué l'espace de modélisation  $E^\natural$  des espaces barycentriques  $E^x = B(\mathbb{R}^{J^x})$ , nous devons distinguer les opérateurs  $T_i^\natural$  (correspondant aux

transitions issues de l'état initial) des autres opérateurs  $T_i^x$ . Cela est dû aux types d'espaces d'arrivée différents entre ces deux familles d'opérateurs.

Les opérateurs  $T_i^{\natural}$  sont de la forme  $T_i^{\natural} : E^{\natural} \leftarrow B(\mathbb{R}^{J^x})$ . Chaque élément  $e_j$  de la base canonique associée à  $B(\mathbb{R}^{J^x})$  correspond à un point de contrôle. La matrice  $T_i^{\natural}$  représente la façon dont ces points de contrôle sont positionnés dans l'espace de modélisation. Chaque colonne de  $T_i^{\natural}$  représente un point de contrôle de  $\delta(\natural, i)$ . Par la suite, nous désignerons ce type de matrices par le terme "matrice de points de contrôle". La modification des points de contrôle entraîne une déformation globale de l'attracteur (voir figure 4.1).



**Fig. 4.1** – Déformation globale d'un attracteur dû à la modification de points de contrôle (en rouge)

Les opérateurs  $T_i^x$  sont de la forme  $T_i^x : R^{J^x} \leftarrow R^{J^y}$ . Par la suite, nous désignerons ce type de matrices par le terme "matrice de subdivision". Ces matrices étant barycentriques, la somme des coefficients sur chaque colonne est égale à un.

#### 4.1.2 Lien avec les formes à pôles

D'après les propriétés que nous venons de décrire précédemment, nous pouvons faire un lien entre notre modèle, et celui des formes à pôles.

Dans les modèles classiques de formes à pôles, l'utilisateur dispose d'un ensemble de points de contrôles  $p_k$  qu'il peut positionner dans l'espace. La figure paramétrée  $h(t)$  est calculé en fonction de la position de ces points et d'un ensemble de fonctions de mélange  $H_k$ .

$$h(t) = \sum_k p_k H_k(t)$$

Ces figures sont généralement des courbes ou des surfaces lisses, car les fonctions de mélange sont généralement polynomiales, ou rationnelles. Le déplacement des points de contrôle entraîne une déformation plus ou moins globale de la figure, tout en conservant son aspect local lisse.

Dans le modèle BCIFS, chaque état  $x$  correspond un certain attracteur  $A^x$  défini dans  $E^x$ . Une matrice de la forme  $T_i^{\natural}$ , avec  $\delta(\natural, i) = x$  définit une certaine projection de cet attracteur  $T_i^{\natural} A^x$  dans l'espace de modélisation  $E^{\natural}$ . Chaque colonne  $T_i^{\natural} e_k$  de  $T_i^{\natural}$  correspond à un point de contrôle  $p_k$ . La modification d'un point de contrôle se traduit par la modification d'une colonne de  $T_i^{\natural}$ , ce qui entraîne une déformation globale de  $T_i^{\natural} A^x$ , de la même manière que pour les formes à pôles. Cependant, à la différence des formes à pôles classiques, les fonctions de mélange  $H_k^x(t)$  ne sont pas prédéfinies, mais sont définies récursivement, selon les opérateurs  $T_i^x$  du BCIFS.

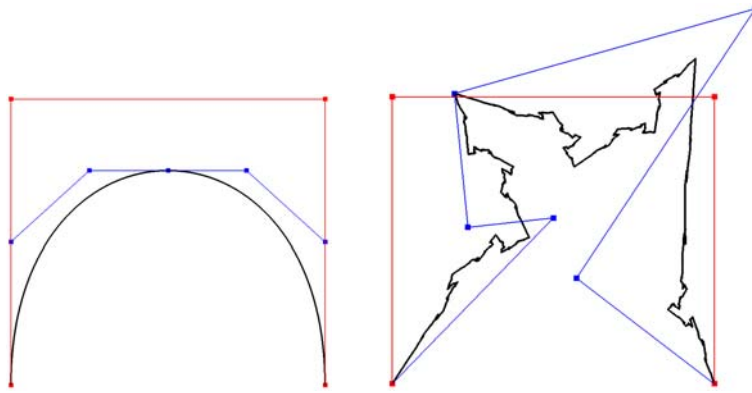
La manipulation des matrices de subdivision peut se faire par l'intermédiaire de "points de subdivision". De la même manière que l'on associe un point de contrôle à chaque colonne

d'une matrice de contrôle, on associe un point de subdivision à chaque colonne d'une matrice de subdivision.

Cependant, chacune de ces colonnes  $q_j$  est définie dans  $E^x = B(\mathbb{R}^{J^x})$ . Pour représenter un point de subdivision  $q_j$  dans l'espace de modélisation  $E^h$ , nous définissons  $p_j$  comme l'image de  $q_j$  par l'opérateur  $T_i^h$ .

$$p_j = T_i^h q_j$$

Comme nous le verrons par la suite, la modification des points de subdivision entraîne des déformations de l'objet très différentes des déformations obtenues par la modification d'un point de contrôle. Ces déformations n'ont plus forcément un effet global, mais affectent généralement l'aspect local des formes. Ces déformations permettent notamment de modifier une forme lisse pour obtenir une forme fractale plus ou moins rugueuse (voir figure 4.2).

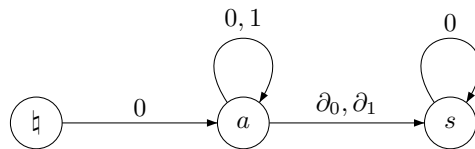


**Fig. 4.2** – Déformations locales d'un attracteur dues à la modification de points de subdivision (en bleu)

### 4.1.3 Exemple d'une courbe

Nous commençons ici par illustrer sur un exemple, les éléments que nous venons de décrire, ainsi que certaines problématiques liés au plongement géométrique des BCIFS. Ces problématiques concernent la visualisation et la saisie interactive des paramètres des BCIFS, ainsi que la construction des espaces de contrôle et le traitement des contraintes de raccord. Chacun de ces points sera développé plus en détail dans les parties suivantes, et nous verrons comment les automatiser.

L'exemple que l'on considère est celui d'une courbe simple à deux subdivisions et trois points de contrôle. L'automate correspondant est illustré sur la figure 4.3.



**Fig. 4.3** – Représentation de l'automate pour une courbe simple à deux subdivisions

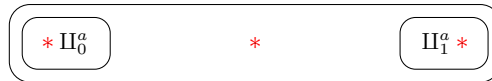
On souhaite avoir une courbe à trois points de contrôle. On définit l'espace associé à l'état  $a$  comme un espace barycentrique de dimension deux  $E^a = B(\mathbb{R}^3)$ .

La matrice de points de contrôle  $T_0^b$  est une matrice  $4 \times 3$  dont chaque colonne représente les coordonnées d'un point de contrôle de la courbe.

$$T_0^b = \left( (p_1), (p_2), (p_3) \right) = \begin{pmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{pmatrix}$$

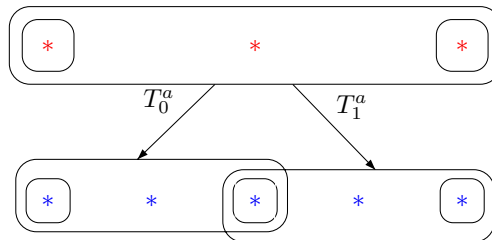
On considère que le premier et le troisième point de  $E^a$  correspondent aux bords  $\Pi_0^a$  et  $\Pi_1^a$ . L'espace associé à l'état  $s$  est alors un espace à un seul point de contrôle  $E^s = B(\mathbb{R})$ , et les opérateurs d'incidence sont définis ainsi :

$$\Pi_0^a = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \Pi_1^a = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$



**Fig. 4.4** – Représentation de l'ensemble des points de contrôle associés à  $a$

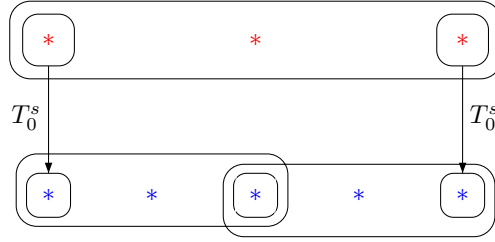
Les matrices  $T_0^a$  et  $T_1^a$  sont des matrices  $3 \times 3$  où chaque colonne correspond à l'image d'un point de contrôle par un opérateur de subdivision.



**Fig. 4.5** – Relations entre les points de contrôle et leur image par les opérateurs de subdivision  $T_0^a$  et  $T_1^a$

Enfin,  $T_0^s$  est l'opérateur de subdivision associé à un sommet  $s$ . C'est une matrice  $1 \times 1$ . L'opérateur étant supposé barycentrique,  $T_0^s$  est déterminé.

$$T_0^s = ( 1 )$$



**Fig. 4.6** – Relations entre les points de contrôle et leur image par les opérateurs de subdivision  $T_0^s$  et  $T_1^s$

Le BCIFS est muni de contraintes de raccords, permettant d'établir la topologie d'une courbe. Ces contraintes sont les suivantes :

$$\begin{aligned} T_0^a \Pi_1^a &= T_1^a \Pi_0^a \\ T_0^a \Pi_0^a &= \Pi_0^a T_0^s \\ T_1^a \Pi_1^a &= \Pi_1^a T_0^s \end{aligned}$$

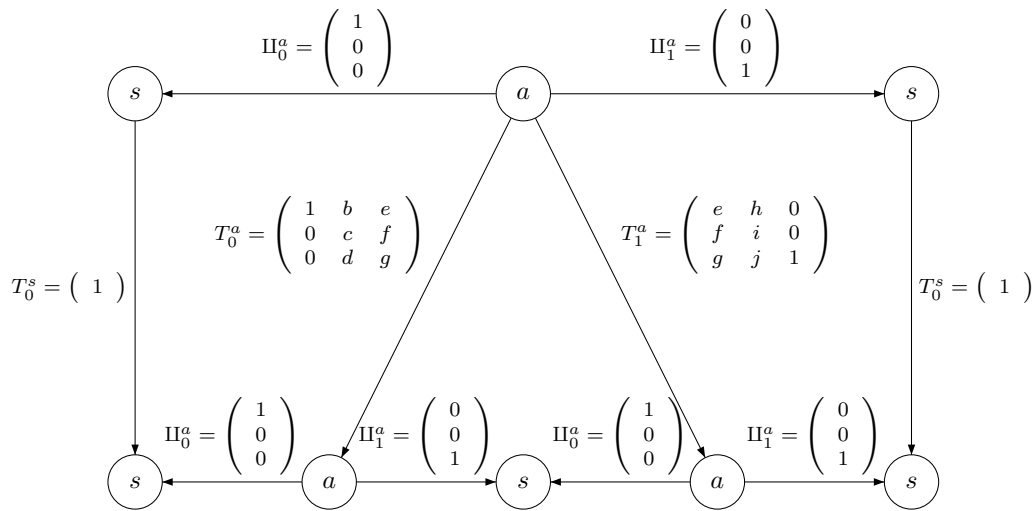
Les transformations géométriques étant représentées par des matrices, les contraintes de raccord se traduisent par des équations matricielles. En résolvant formellement ces équations matricielles, on obtient une forme générale des opérateurs, où certains coefficients sont nuls, et certains sont liés à d'autres.

$$T_0^a = \begin{pmatrix} 1 & b & e \\ 0 & c & f \\ 0 & d & g \end{pmatrix} \quad T_1^a = \begin{pmatrix} e & h & 0 \\ f & i & 0 \\ g & j & 1 \end{pmatrix}$$

Cette forme générale utilise un ensemble de coefficients indépendants qui satisfont les contraintes du BCIFS.

$$\begin{aligned} T_0^a \Pi_1^a = T_1^a \Pi_0^a &\Leftrightarrow \begin{pmatrix} 1 & b & e \\ 0 & c & f \\ 0 & d & g \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} e & h & 0 \\ f & i & 0 \\ g & j & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\ T_0^a \Pi_0^a = \Pi_0^a T_0^s &\Leftrightarrow \begin{pmatrix} 1 & b & e \\ 0 & c & f \\ 0 & d & g \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} (1) \\ T_1^a \Pi_1^a = \Pi_1^a T_0^s &\Leftrightarrow \begin{pmatrix} e & h & 0 \\ f & i & 0 \\ g & j & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} (1) \end{aligned}$$

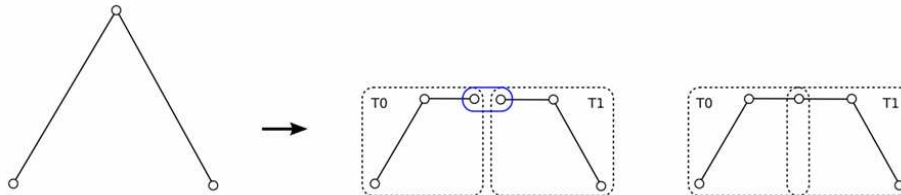
La figure 4.7 montre le graphe de subdivision avec la représentation matricielle des opérateurs.



**Fig. 4.7** – Forme générale des opérateurs munis des contraintes du BCIFS

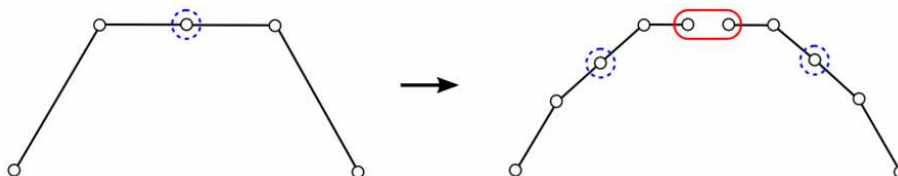
Parmi les contraintes de raccord définissant cette courbe, nous distinguons deux types de contraintes, qui sont les contraintes d'adjacence, et d'incidence.

La première contrainte  $T_0^a \Pi_1^a = T_1^a \Pi_0^a$  est une contrainte d'adjacence, entre les deux subdivisions de  $a$ . Elle a pour effet, matriciellement, d'identifier deux colonnes dans les matrices de subdivision  $T_0^a$  et  $T_1^a$ . Géométriquement, cela se traduit par la mise en commun de deux points de subdivision au premier niveau de subdivision, comme le montre la figure 4.8.



**Fig. 4.8** – Expression des contraintes de raccord internes au premier niveau de subdivision

Les deux autres contraintes  $T_0^a \Pi_0^a = \Pi_0^a T_0^s$  et  $T_1^a \Pi_1^a = \Pi_1^a T_0^s$  sont des contraintes d'incidence, portant sur la subdivision des sommets. Du point de vue matriciel, elles ont pour effet d'identifier certains blocs dans les matrices de subdivision  $T_0^a$  et  $T_1^a$ , et de créer des zéros sur les mêmes colonnes en dehors de ces blocs. Géométriquement, cela se traduit par la mise en commun de deux points de subdivision au milieu de la courbe au second niveau de subdivision, comme le montre la figure 4.8. Sans ces contraintes, le raccord au milieu de la courbe serait rompu au second niveau de subdivision, car c'est la mise en commun des subdivisions des bords de chacun des deux morceaux qui permet de maintenir le raccord.



**Fig. 4.9** – Effet des contraintes d'incidence au second niveau de subdivision

La forme générale des opérateurs de subdivision, obtenue par la résolution matricielle des contraintes de raccord, permet de décomposer les paramètres des matrices en un ensemble de paramètres indépendants. Nous disposons donc d'un ensemble de paramètres que nous pouvons moduler, afin de faire varier les propriétés géométriques de l'objet modélisé.

Ces paramètres, regroupés en colonnes, peuvent être représentés géométriquement comme un ensemble de points dans l'espace de modélisation. Ces points sont regroupés en deux familles, les points de contrôle et les points de subdivision, selon que les opérateurs soient directement issus de l'état initial ou non.

Dans notre exemple, nous avons trois points de contrôle, qui correspondent aux trois colonnes de la matrice  $T^b$ . Nous avons également deux opérateurs de subdivision  $T_0^a$  et  $T_1^a$  auxquels sont associés en tout six points de subdivision.

L'équation  $T_0^a \Pi_0^a = \Pi_0^a T_0^s$  fait que le point de subdivision associé à la première colonne de  $T_0^a$  n'est plus un paramètre, car cette colonne est entièrement déterminée; le premier point de subdivision de  $a$  par  $T_0^a$  est confondu avec le premier point de contrôle de  $a$ . De même pour l'équation  $T_0^a \Pi_0^a = \Pi_0^a T_0^s$ , le troisième point de subdivision de  $a$  par  $T_1^a$  est confondu avec le troisième point de contrôle de  $a$ . Enfin, d'après l'équation  $T_0^a \Pi_1^a = T_1^a \Pi_0^a$ , le troisième point de subdivision de  $T_0^a$ , bien qu'il ne soit pas déterminé, est confondu avec le premier point de subdivision de  $T_1^a$ .

Nous obtenons donc une courbe paramétrable par trois points de contrôle, et trois points de subdivision. Ces points permettent une saisie interactive des paramètres dans une interface graphique. Le déplacement d'un point de contrôle ou d'un point de subdivision entraîne la mise à jour de la colonne correspondante, en permettant de visualiser directement les effets de cette modification. La figure 4.10 montre les effets sur la courbe de la modification des points de subdivision.

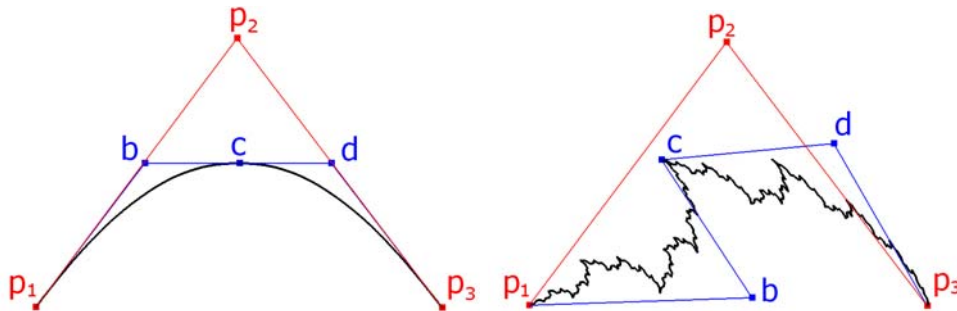


Fig. 4.10 – Plongements géométriques d'une courbe, avec des paramètres géométriques différents

## 4.2 Résolution des contraintes topologiques

L'algorithme de résolution des contraintes topologiques prend comme entrée la description d'un BCIFS, et génère les matrices associées aux transitions, ainsi que les contraintes matricielles auxquelles elles sont soumises. Cet algorithme se décompose en deux étapes. Chacune de ces étapes traite un sous-ensemble de contraintes dans  $(\Upsilon^x)_{x \in \mathcal{Q}}$ . On partitionne chaque  $\Upsilon^x$  en deux sous-ensembles  $\Upsilon^x = \tilde{\Upsilon}^x \cup \hat{\Upsilon}^x$  :

- $\tilde{\Upsilon}^x$  l'ensemble des couples de la forme  $(\partial_i \partial_j, \partial_k \partial_\ell)$  décrivant les équations d'adjacence sur les opérateurs d'incidence notées  $\doteq$  :

$$\forall (\partial_i \partial_j, \partial_k \partial_\ell) \in \tilde{\Upsilon}^x \begin{cases} \delta(x, \partial_i \partial_j) & = \delta(x, \partial_k \partial_\ell) \\ \Pi_i^x \Pi_j^{\delta(x, \partial_i)} & \doteq \Pi_k^x \Pi_\ell^{\delta(x, \partial_k)} \end{cases}$$

- $\hat{\Upsilon}^x$  l'ensemble des couples de la forme  $(i \partial_j, k \partial_\ell)$  et  $(\partial_i j, k \partial_\ell)$ , décrivant les équations

d'incidence et d'adjacence sur les opérateurs de subdivision notées  $\hat{=}$  :

$$\forall(i \partial_j, k \partial_\ell) \in \hat{\Upsilon}^x \begin{cases} \delta(x, i \partial_j) & = \delta(x, k \partial_\ell) \\ T_i^x \Pi_j^{\delta(x,i)} & \hat{=} T_k^x \Pi_\ell^{\delta(x,k)} \end{cases}$$

$$\forall(\partial_i j, k \partial_\ell) \in \Upsilon^x \begin{cases} \delta(x, \partial_i j) & = \delta(x, k \partial_\ell) \\ \Pi_i^x T_j^{\delta(x,\partial_i)} & \hat{=} T_k^x \Pi_\ell^{\delta(x,k)} \end{cases}$$

La première étape consiste à déterminer les espaces associés aux états. Cette étape prend en compte la “dimension interne”  $\hat{m}^x$  attribuée à chaque état, et le sous-ensemble de contraintes  $\hat{\Upsilon}^x$ . La dimension interne  $\hat{m}^x$  est le nombre de points de contrôles associés à  $x$  ne faisant pas partie des points de contrôles associés à ses bords. Ceci détermine la dimension totale de chaque espace  $E^x$ , définit entièrement les matrices d'incidence  $\Pi_i^x$ , et détermine la dimension des matrices de subdivision  $T_i^x$ .

La seconde étape prend en compte le sous-ensemble de contraintes  $\hat{\Upsilon}^x$ . Ces contraintes portent sur les matrices de subdivision. Elles se traduisent par l'identification de coefficients de certaines matrices avec d'autres coefficients ou des constantes nulles.

## 4.2.1 Construction des espaces

L'espace  $E^x$ , associé à une composante  $x$ , est l'espace engendré par son espace interne et les différents espaces de ses sous-cellules incidentes. Ces derniers ne sont généralement pas en somme directe. Les équations d'adjacence liant les opérateurs d'incidence 3.1 doivent être prises en compte pour le calcul de la dimension de  $E^x$ , et pour la construction des matrices de bordure  $\Pi_{\partial_i}^x$ , qui doivent vérifier ces équivalences.

La construction des espaces de contrôle est faite de manière récursive. Pour chaque composante  $x$ , on détermine les espaces de contrôle  $E^x$ , c'est-à-dire l'ensemble d'indices de contrôle  $J^x$ , et les opérateurs de bord associés  $\Pi_u^x$  et  $\eta_u^x$ .

On détermine ces ensembles après avoir défini les figures de dimension inférieure. On choisit donc en premier les composantes n'ayant pas d'incidence, c'est-à-dire les sommets  $J^s$ , puis on détermine les arêtes  $J^a$  d'après les sommets, et les faces  $J^f$  d'après les arêtes.

### Espaces internes des composantes

Lorsque deux composantes  $x$  et  $y$  sont liées par un opérateur d'incidence  $\Pi_i^x$ , cela implique que  $\Pi_i^x E^y$  est un sous-espace de  $E^x$

$$\partial_i \in \Sigma_\partial^x \Rightarrow E^x \supseteq \Pi_i^x E^y$$

ce qui se traduit en termes d'indices par

$$\partial_i \in \Sigma_\partial^x \Rightarrow J^x \supseteq \eta_i^x \left( J^{\delta(x,\partial_i)} \right)$$

où  $\eta_i^x$  est une fonction  $J^x \leftarrow J^{\delta(x,\partial_i)}$  associant les indices des éléments de  $E^{\delta(x,\partial_i)}$  correspondant à un élément de  $E^x$ , d'après l'opérateur  $\Pi_i^x : E^x \leftarrow E^{\delta(x,\partial_i)}$ .

$$e_{\eta_i^x(j)}^x = \Pi_i^x e_j^{\delta(x,i)}$$

De plus, on peut choisir de munir chaque composante  $x$  d'un sous-espace  $\dot{E}^x$  indépendant des sous-espaces de ses bordures.

$$J^x = J^x \setminus \bigcup_{i \in \Sigma_\partial^x} \eta_i^x \left( J^{\delta(x,\partial_i)} \right)$$

La dimension interne  $\hat{m}_x$  passée en paramètre pour chaque composante  $x$  est donc la dimension de l'espace interne  $\dot{J}_x$  correspondant.



## Résolution des contraintes d'adjacence liant les opérateurs d'incidence

On construit dans un premier temps un ensemble  $\tilde{J}^x$  en prenant l'union disjointe des occurrences des  $J^{\delta(x,u)}$  dans  $x$  :

$$\tilde{J}^x = \bigcup_{u \in \Sigma_{\partial}^x} \{u\} \times J^{\delta(x,u)}$$

Pour déterminer  $J^x$  on en prend le quotient de  $\tilde{J}^x$  par la relation d'équivalence associée au système d'équations défini par  $\tilde{Y}^x$ , complété par les indices de l'espace interne  $\tilde{J}^x$  :

$$J^x = \tilde{J}^x / \mathcal{E}^x \cup \tilde{J}^x$$

$\mathcal{E}^x$  est la fermeture réflexive symétrique transitive de la relation  $\mathcal{R}^x$  définie sur l'ensemble d'indices  $(u, j) \in \tilde{J}^x$ .

À chaque équation  $(u_1 u_2, v_1 v_2) \in \tilde{Y}^x$  on associe l'état  $z = \delta(x, u_1 u_2) = \delta(x, v_1 v_2)$  et l'ensemble de couples :

$$\mathcal{R}_{(u_1 u_2, v_1 v_2)}^x = \bigcup_{\ell \in J^z} \left\{ \left( (u_1, \eta_{u_2}^{\delta(x, u_1)}(\ell)), (v_1, \eta_{v_2}^{\delta(x, v_1)}(\ell)) \right) \right\}$$

À l'état  $x$  est associé :

$$\mathcal{R}^x = \bigcup_{(u_1 u_2, v_1 v_2) \in \tilde{Y}^x} \mathcal{R}_{(u_1 u_2, v_1 v_2)}^x$$

$\mathcal{E}^x$  est alors donné par :

$$\mathcal{E}^x = (Id \cup \mathcal{R}^x \cup (\mathcal{R}^x)^{-1})^*$$

En notant  $C^x : J^x \leftarrow \tilde{J}^x$  la fonction associant chaque couple  $(u, j) \in \tilde{J}^x$  à sa classe d'équivalence selon  $\mathcal{E}^x$ , les fonctions  $\eta_u^x : J^x \leftarrow J^{\delta(x, u)}$  sont données par :

$$\eta_u^x = C^x \circ \tilde{\eta}_u^x$$

avec

$$\tilde{\eta}_u^x(j) = (u, j)$$

## Calcul matriciel booléen

Les ensembles d'indices peuvent être représentés par des vecteurs booléens. Les relations et les fonctions sur les ensembles d'indices sont représentés par des matrices booléennes.

La fonction  $\tilde{\eta}_i^x$  est représentée par la matrice booléenne  $\tilde{\Pi}_i^x$ .

$$\tilde{\Pi}_i^x : \tilde{J}^x \times J^{\delta(x, \partial_i)} \rightarrow \{0, 1\}$$

L'ensemble des matrices  $\tilde{\Pi}_i^x$  associent les indices  $J^{\delta(x, \partial_i)}$  à des sous-ensembles disjoints de couples  $(u, j) \in \tilde{J}^x$ . Un procédé simple pour construire ces matrices consiste à extraire des blocs issus d'une matrice identité sur  $\tilde{J}^x$ .

La relation d'équivalence  $\mathcal{E}^x$  est représentée par la matrice booléenne  $M^{x*} : \tilde{J}^x \times \tilde{J}^x \rightarrow \{0, 1\}$ .

$$M^{x*}[u, v] = 1 \Leftrightarrow u \mathcal{E}^x v$$

La matrice de projection  $\Pi^x : J^x \times \tilde{J}^x \rightarrow \{0, 1\}$  correspondant aux classes d'équivalences  $C^x$  est :

$$\Pi^x \tilde{e}_{(u, j)} = e_{C^x(u, j)}^x$$

La fonction  $\eta_u^x = C^x \circ \tilde{\eta}_u^x$  est représentée par  $\Pi_u^x = \Pi^x \tilde{\Pi}_u^x$ .

## Remarque

Les matrices booléennes  $\Pi_i^x : J^x \times J^{\delta(x, \partial_i)} \rightarrow \{0, 1\} \subset \mathbb{R}$  sont identifiables à des sous-ensembles de matrices numériques représentant des applications linéaires entre espaces homogènes  $\mathbb{R}^{J^x} \leftarrow \mathbb{R}^{J^{\delta(x, \partial_i)}}$  et les applications affines entre espaces barycentriques  $E^x = B(\mathbb{R}^{J^x}) \leftarrow E^{\delta(x, \partial_i)} = B(\mathbb{R}^{J^{\delta(x, \partial_i)}})$ . On a ainsi un calcul direct des opérateurs d'incidence  $\Pi_i^x$ .

## Algorithme de construction

On donne ici une méthode algorithmique pour déterminer les espaces associés aux états, et les opérateurs d'incidence. Cet algorithme suppose que le graphe des incidences ne contient aucun circuit.

```
ConstruireEspaces(Q)
début
  | pour tous les  $x \in Q$  faire
  |   | ConstruireEspace(x)
  |   fin
  fin

ConstruireEspace(x)
début
  | si espaceConstruit(x) alors
  |   | retourner
  |   fin
  |  $\tilde{m}^x = 0$ 
  | pour tous les  $u \in \Sigma_\partial^x$  faire
  |   | ConstruireEspace( $\delta(x, u)$ )
  |   |  $\tilde{m}^x = \tilde{m}^x + m^{\delta(x, u)}$ 
  |   fin
  |  $M = Id_{\tilde{m}^x}$ 
  | pour tous les  $u \in \Sigma_\partial^x$  faire
  |   | Initialiser( $\tilde{\Pi}_u^x, \tilde{\eta}_u^x$ )
  |   fin
  | pour tous les  $(u_1 u_2, v_1 v_2) \in \Upsilon^x$  faire
  |   | pour tous les  $j \in J^{\delta(x, u_1 u_2)}$  faire
  |     |  $M[\tilde{\eta}_{u_1}^x(\eta_{u_2}^{\delta(x, u_1)}(j))][\tilde{\eta}_{v_1}^x(\eta_{v_2}^{\delta(x, v_1)}(j))] = 1$ 
  |     fin
  |   fin
  |  $M^{x*} = FermetureSymetriqueTransitive(M)$ 
  |  $\Pi^x = SupprimerLignesMultiples(M^{x*})$ 
  |  $\Pi^x = AjouterLignesNulles(\Pi^x, \tilde{m}^x)$ 
  | pour tous les  $u \in \Sigma_\partial^x$  faire
  |   |  $\Pi_u^x = \Pi^x \tilde{\Pi}_u^x$ 
  |   fin
  |  $m^x = NombreDeLignes(M)$ 
  |  $J^x = \{0; \dots; m^x - 1\}$ 
fin
```

### Application sur l'exemple du carreau

La face, associée à l'état  $f$ , est bordée par quatre arêtes et quatre sommets  $s$ . On distingue deux types d'arêtes  $a$  et  $b$ . Chaque arête,  $a$  ou  $b$ , est bordée par deux sommets  $s$ . On a quatre opérateurs d'incidence sur  $f : \Pi_0^f, \Pi_1^f, \Pi_2^f, \Pi_3^f$ , 2 opérateurs d'incidence sur  $a : \Pi_0^a$  et  $\Pi_1^a$ , de même pour  $b : \Pi_0^b$  et  $\Pi_1^b$ .

La structure de raccord de la face est donnée par quatre équations d'adjacence liant les opérateurs d'incidence.

On dispose d'une structure de grille dont les nombres de points de contrôle sont paramétrables selon les dimensions internes  $\dot{m}_s, \dot{m}_a, \dot{m}_b$  et  $\dot{m}_f$ .

$$\begin{aligned} |J^s| &= m^s = \dot{m}_s \\ |J^a| &= m^a = \dot{m}^a + 2m^s = \dot{m}_a + 2\dot{m}_s \\ |J^b| &= m^b = \dot{m}^b + 2m^s = \dot{m}_b + 2\dot{m}_s \\ |J^f| &= m^f = \dot{m}^f + 2m^a + 2m^b - 4m^s \\ &= \dot{m}_f + 2(\dot{m}_a + 2\dot{m}_s) + 2(\dot{m}_b + 2\dot{m}_s) - 4\dot{m}_s \\ &= \dot{m}_f + 2\dot{m}_a + 2\dot{m}_b + 4\dot{m}_s \end{aligned}$$

On choisit d'attribuer à chaque état  $f, a, b$  et  $s$  une dimension interne de 1.

$$\dot{m}_f = \dot{m}_a = \dot{m}_b = \dot{m}_s = 1$$

On traite  $s$  avant  $a$  et  $b$ , puis  $a$  et  $b$  avant  $f$ .

L'état  $s$  ne contient aucune relation d'incidence. L'espace  $E^s$  associé est donc constitué uniquement de son espace interne  $\dot{E}^s$ , soit un espace barycentrique à un seul point de contrôle.

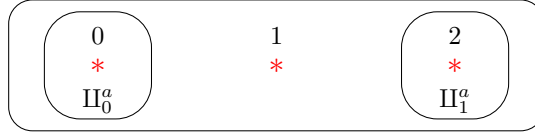
$$|J^s| = m^s = \dot{m}^s = 1 \Rightarrow E^s = B(\mathbb{R}^{J^s}) = B(\mathbb{R}^1)$$

Les états  $a$  et  $b$  sont chacun bordés par deux  $s$ . Aucune équation d'adjacence sur les incidences n'est associée à ces états. Les ensembles d'indices  $J^a$  et  $J^b$  sont donc chacun composés des indices  $\dot{J}^a$  et  $\dot{J}^b$  associés à leur espace interne, et de deux copies de  $J^s$ , ce qui engendre un espace barycentrique à trois points de contrôle.

$$\begin{aligned} |J^a| = m^a = \dot{m}^a + 2m^s = 1 + 2 \times 1 = 3 \quad , \quad J^a &= \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} \\ |J^b| = m^b = \dot{m}^b + 2m^s = 1 + 2 \times 1 = 3 \quad , \quad J^b &= \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} \end{aligned}$$

Les opérateurs de plongement  $\Pi_0^a : E^a \leftarrow E^s$  et  $\Pi_1^a : E^a \leftarrow E^s$  sont des matrices  $\mathbb{R}^{3 \times 1}$ , chacune extrayant un point de contrôle différent. Il en est de même pour ceux de  $b$ .

$$\begin{aligned} \Pi_0^a &= \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} & \Pi_1^a &= \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \\ \Pi_0^b &= \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} & \Pi_1^b &= \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \end{aligned}$$



**Fig. 4.11** – Représentation de l'ensemble des points de contrôle associés à  $a$  et de leur indice

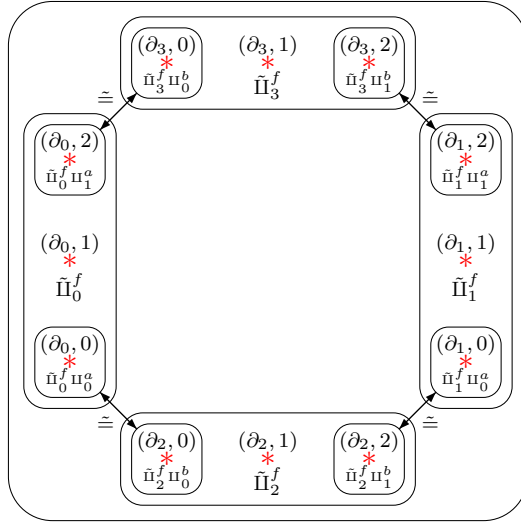
L'état  $f$  est bordé par deux  $a$  et deux  $b$ . Quatre équations d'adjacence sur les incidences définissent la structure de raccord de  $f$ .

$$\begin{aligned} \Pi_0^f \Pi_1^a &\cong \Pi_3^f \Pi_0^b & \Pi_1^f \Pi_1^a &\cong \Pi_3^f \Pi_1^b \\ \Pi_0^f \Pi_0^a &\cong \Pi_2^f \Pi_0^b & \Pi_1^f \Pi_0^a &\cong \Pi_2^f \Pi_1^b \end{aligned}$$

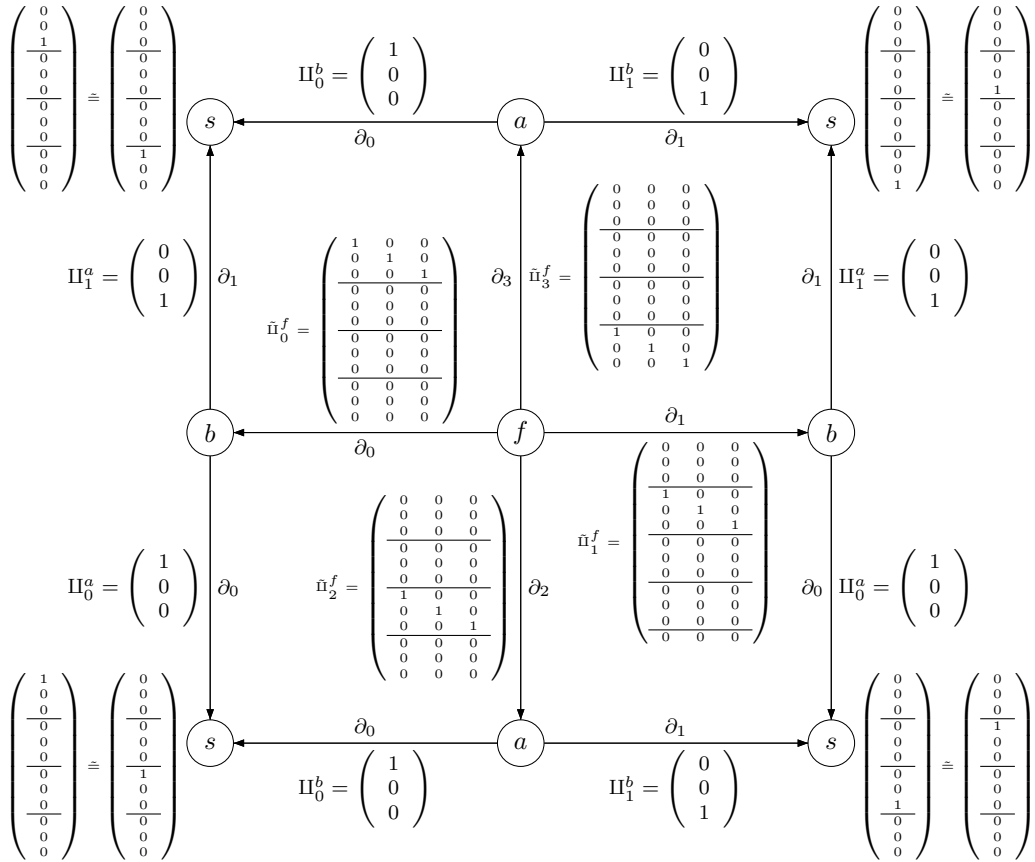
On définit dans un premier temps l'ensemble  $\tilde{J}^f$  et quatre matrices de plongement  $\tilde{\Pi}_i^f : \tilde{J}^f \times J^{\delta(f, \partial_i)} \rightarrow \{0, 1\}$ .

$$|\tilde{J}^f| = 2m^a + 2m^b = 2 \times 3 + 2 \times 3 = 12 \quad , \quad \tilde{J}^f = \begin{pmatrix} (\partial_0, 0) \\ (\partial_0, 1) \\ (\partial_0, 2) \\ \hline (\partial_1, 0) \\ (\partial_1, 1) \\ (\partial_1, 2) \\ \hline (\partial_2, 0) \\ (\partial_2, 1) \\ (\partial_2, 2) \\ \hline (\partial_3, 0) \\ (\partial_3, 1) \\ (\partial_3, 2) \end{pmatrix}$$

$$\tilde{\Pi}_0^f = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \tilde{\Pi}_1^f = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \tilde{\Pi}_2^f = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \tilde{\Pi}_3^f = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



**Fig. 4.12** – Représentation de l'ensemble des indices de  $\tilde{J}^f$ , et des équations d'adjacence à prendre en compte pour la construction de  $J^f$



**Fig. 4.13** – Structure de raccord d'un carreau, avec matrices de plongement initiales

Ces matrices ne vérifient pas les relations d'équivalence (voir figure 4.13), mais permettent d'identifier les éléments de  $\tilde{J}^f$  qui doivent être mis en commun.

La matrice booléenne  $M^{f*}$  représente la relation d'équivalence  $\mathcal{E}^f$  sur  $\tilde{J}^f$ .

$$\mathcal{R}^f = \left\{ \begin{array}{l} ((\partial_0, 0), (\partial_2, 0)) \\ ((\partial_0, 1), (\partial_3, 0)) \\ ((\partial_1, 0), (\partial_2, 3)) \\ ((\partial_1, 1), (\partial_3, 3)) \end{array} \right\} \quad M^{f*} = \left( \begin{array}{ccc|ccc|ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right)$$

La matrice booléenne  $\Pi^f$  représente la projection  $C^f$ .

$$\Pi^f = \left( \begin{array}{ccc|ccc|ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right)$$

Les opérateurs d'incidence  $\Pi_i^f = \Pi^f \tilde{\Pi}_i^f$  représentent les fonctions  $J^f \leftarrow J^{\delta(f, \partial_i)}$  correspondant à  $\eta_i^f = C^f \circ \tilde{\eta}_i^f$ .

$$\Pi_0^f = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \Pi_1^f = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \Pi_2^f = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \Pi_3^f = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

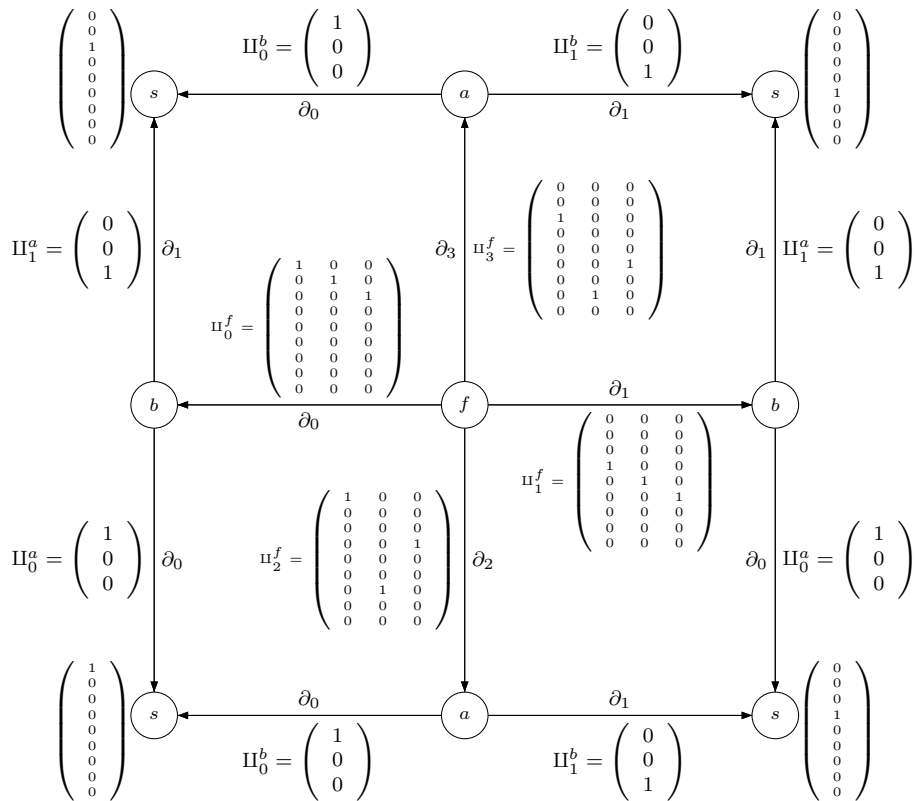


Fig. 4.14 – Structure de raccord d'un carreau, avec matrices de plongement vérifiant les équations d'adjacence sur les opérateurs d'incidence

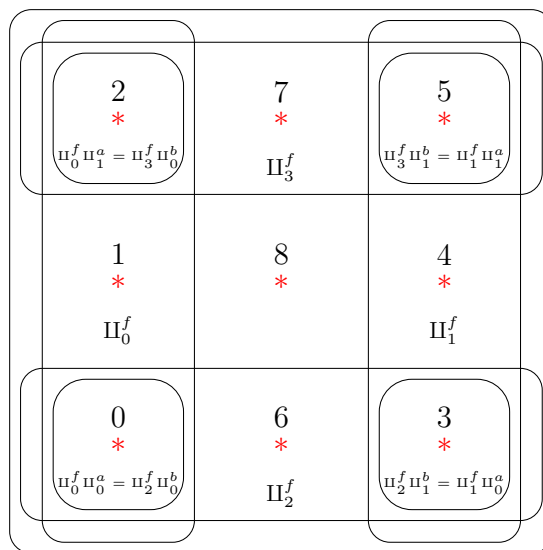


Fig. 4.15 – Représentation des opérateurs d'incidence associés à  $f$  et à ses bords

## 4.2.2 Matrices de subdivision

Le dimensionnement de chaque état  $E^x$  étant déterminé, on connaît la dimension des matrices de transformation  $T_i^x$ .

Les contraintes matricielles faisant intervenir des opérateurs de subdivision se traduisent par des équations sur les matrices de subdivision. Elles sont résolues en attribuant des valeurs formelles aux matrices. Ces équations ont pour effet l'identification de coefficients matriciels entre eux, ou bien leur fixation à une constante donnée.

### Calcul matriciel formel

On utilise une représentation formelle permettant de coder les coefficients des matrices de subdivision et d'incidence.

$$param(x) = const(0) | const(1) | var(x)$$

On définit les opérations formelles  $+$  et  $\times$  sur ces coefficients, qui sont utilisées pour calculer les produits matriciels.

$$\begin{aligned} const(0) + param(x) &= param(x) \\ const(0) \times param(x) &= const(0) \\ const(1) \times param(x) &= param(x) \end{aligned}$$

Certaines opérations ne sont pas définies, car elles n'apparaissent pas dans les équations. Les matrices d'incidence ont, par hypothèse, un seul 1 par colonne, ce qui élimine la possibilité de voir apparaître des expressions de la forme  $const(1) + param(x)$  ou bien  $var(x) + var(y)$ . Les produits matriciels sont composés, au plus, d'une matrice de subdivision, ce qui élimine les cas  $var(x) \times var(y)$ .

Après le calcul des produits matriciels sur chacun des membres, les équations matricielles se traduisent par l'identification de certains coefficients entre eux, qui peuvent être l'identification de deux variables entre elles, ou bien l'identification d'une variable avec une constante nulle.

$$\begin{aligned} var(x) &\hat{=} var(y) \\ var(x) &\hat{=} const(0) \end{aligned}$$

### Cas d'erreurs possibles

Lorsque l'utilisateur spécifie un ensemble de contraintes, la résolution formelle des équations peut aboutir à certaines contradictions.

Toutes les identifications entre coefficients matriciels ne sont pas autorisées. L'apparition du cas  $const(0) \hat{=} const(1)$  résulterait d'une erreur dans les équations matricielles fournies en entrée. De plus, lorsqu'une matrice contient une colonne entièrement nulle, cela constitue également un cas d'erreur, qui doit être détecté par l'algorithme et signalé à l'utilisateur.

### Réduction de paramètres sur les matrices de subdivision

On distinguera deux types de contraintes, dont on peut interpréter différemment les effets en termes de réductions de paramètres.



## Matrice globale

Les contraintes d'adjacence sur les opérateurs de subdivision, du type 3.2

$$T_i^x \Pi_j^y \cong T_k^x \Pi_\ell^z$$

se traduisent par l'identification de colonnes entières de matrices de subdivision. L'ensemble des contraintes de ce type permet de déterminer une matrice associée à l'état  $x$  notée  $T_x^x$  que l'on appellera "matrice globale" de  $x$ . Les colonnes de cette matrice représentent les classes d'équivalence des colonnes des matrices de subdivision de  $x$ . Pour la saisie des paramètres, la matrice globale représente l'ensemble des points de subdivision associés à un état.

## Masques

Les contraintes d'incidence, du type 3.3

$$\Pi_k^x T_\ell^z \cong T_i^x \Pi_j^y$$

se traduisent par l'identification des coefficients de  $T_\ell^z$  avec un bloc de  $T_i^x$ , et la création de constantes nulles sur les mêmes colonnes en dehors de ces blocs dans  $T_i^x$ . L'ensemble des paramètres non nuls dans les matrices de subdivision détermine les masques de subdivision associées à l'état  $x$ .

## Paramètres indépendants

Au niveau des coefficients de matrices, on obtient trois types de paramètres formels :

- un coefficient peut être sans aucune contrainte, auquel cas il peut être modifié librement, et indépendamment des autres paramètres ;
- un coefficient peut être lié avec un autre, ce qui signifie que la modification d'un paramètre doit entraîner la mise à jour des autres paramètres liés ;
- un coefficient peut être fixé à une constante nulle, et ce coefficient n'est donc pas un paramètre que l'on peut modifier, faute de quoi les relations d'adjacence ne pourraient plus être satisfaites.

Les paramètres peuvent s'interpréter non plus comme un ensemble de coefficients, mais comme un ensemble de colonnes. Ces paramètres représentent alors un ensemble de points de contrôle et de points de subdivision, permettant leur affichage et leur saisie dans une interface graphique. Les points de contrôle correspondent aux colonnes issues d'une matrice de projection dans l'espace de modélisation  $E^b$ . Les points de subdivision correspondent aux colonnes des autres matrices de subdivision.

### 4.2.3 Exemple d'une courbe

Soit une courbe dont la topologie est définie par les opérateurs de subdivision  $T_0^a$ ,  $T_1^a$  et  $T_0^s$ , et une matrice de points de contrôle  $T_0^b$ .

On rappelle les équations d'incidence et d'adjacence définissant les courbes :

$$\Pi_0^a T_0^s \cong T_0^a \Pi_0^a \quad T_0^a \Pi_1^a \cong T_1^a \Pi_0^a \quad \Pi_1^a T_0^s \cong T_1^a \Pi_1^a$$

#### Courbe avec raccord d'ordre 0

Une courbe avec un raccord géométrique d'ordre 0 s'obtient en établissant des raccords sur un point de contrôle. On attribue donc à chaque sommet  $s$  un seul point de contrôle.

On choisit la dimension interne  $\dot{m}_a$  de manière à obtenir une courbe à trois points de contrôle. Sachant que  $a$  possède déjà un point de contrôle pour chaque sommet, on attribue un point de contrôle interne à  $a$ .

$$\begin{aligned}\dot{m}_s &= 1 \\ \dot{m}_a &= 1 \\ m_s &= \dot{m}_s = 1 \\ m_a &= \dot{m}_a + 2m_s = 3\end{aligned}$$

On définit les opérateurs de plongement suivants :

$$\Pi_0^a = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \Pi_1^a = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Avant de résoudre les contraintes matricielles, on initialise les matrices de subdivision avec un ensemble de paramètres supposés *a priori* variables et indépendants.

$$\begin{aligned}T_0^a &= \begin{pmatrix} p_{0,w} & p_{1,w} & p_{2,w} \\ p_{0,x} & p_{1,x} & p_{2,x} \\ p_{0,y} & p_{1,y} & p_{2,y} \\ p_{0,z} & p_{1,z} & p_{2,z} \end{pmatrix} \\ T_0^a &= \begin{pmatrix} a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{pmatrix} \\ T_1^a &= \begin{pmatrix} d_0 & e_0 & f_0 \\ d_1 & e_1 & f_1 \\ d_2 & e_2 & f_2 \end{pmatrix} \\ T_0^s &= (g_0)\end{aligned}$$

La résolution des contraintes matricielles se traduit par l'identification de certains coef-

ficients. Pour cela, on exprime formellement les équations  $\hat{=}$  sous forme matricielle.

$$\hat{=} \begin{cases} T_0^a \Pi_1^a = \begin{pmatrix} a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} \\ T_1^a \Pi_0^a = \begin{pmatrix} d_0 & e_0 & f_0 \\ d_1 & e_1 & f_1 \\ d_2 & e_2 & f_2 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} d_0 \\ d_1 \\ d_2 \end{pmatrix} \end{cases}$$

$$\hat{=} \begin{cases} T_0^a \Pi_0^a = \begin{pmatrix} a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} \\ \Pi_0^a T_0^s = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} (g_0) = \begin{pmatrix} g_0 \\ 0 \\ 0 \end{pmatrix} \end{cases}$$

$$\hat{=} \begin{cases} T_1^a \Pi_1^a = \begin{pmatrix} d_0 & e_0 & f_0 \\ d_1 & e_1 & f_1 \\ d_2 & e_2 & f_2 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \end{pmatrix} \\ \Pi_1^a T_0^s = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} (g_0) = \begin{pmatrix} 0 \\ 0 \\ g_0 \end{pmatrix} \end{cases}$$

On obtient des identifications entre certains coefficients.

$$\begin{aligned} c_0 &\hat{=} d_0 & a_0 &\hat{=} f_2 &\hat{=} g_0 \\ c_1 &\hat{=} d_1 & a_1 &\hat{=} a_2 &\hat{=} f_0 &\hat{=} f_1 &\hat{=} 0 \\ c_2 &\hat{=} d_2 \end{aligned}$$

Ceci permet d'obtenir une représentation formelle de la matrice de points de contrôles  $T_0^{\sharp}$ , des matrices de subdivision  $T_0^a$ ,  $T_1^a$  et  $T_0^s$ , et de la matrice globale  $T_{\div}^a$ .

$$T_0^{\sharp} = \begin{pmatrix} p_{0,w} & p_{1,w} & p_{2,w} \\ p_{0,x} & p_{1,x} & p_{2,x} \\ p_{0,y} & p_{1,y} & p_{2,y} \\ p_{0,z} & p_{1,z} & p_{2,z} \end{pmatrix}$$

$$T_0^a = \begin{pmatrix} a_0 & b_0 & c_0 \\ 0 & b_1 & c_1 \\ 0 & b_2 & c_2 \end{pmatrix}$$

$$T_1^a = \begin{pmatrix} c_0 & e_0 & 0 \\ c_1 & e_1 & 0 \\ c_2 & e_2 & a_0 \end{pmatrix}$$

$$T_0^s = (a_0)$$

$$T_{\div}^a = \begin{pmatrix} a_0 & b_0 & c_0 & e_0 & 0 \\ 0 & b_1 & c_1 & e_1 & 0 \\ 0 & b_2 & c_2 & e_2 & a_0 \end{pmatrix}$$

## 4.3 Exemples de formes à pôles

### 4.3.1 Courbe avec raccord d'ordre 2

Cet exemple a en commun la structure topologique de l'exemple précédent, dans la partie 4.2.3, mais se distingue par la dimension des espaces associés aux états. Lorsque l'on définit une courbe  $a$  bordée par deux sommets  $s$ , l'état  $s$  sert à établir les raccords nécessaires pour garantir la continuité. Cependant, lorsque l'on élève le nombre de point de contrôle de  $s$ , on élève l'ordre de "raccord géométrique".

Dans l'approche classique des schémas de subdivision, l'ordre de raccord géométrique correspond à l'ordre de différentiabilité. Dans l'approche fractale, cette notion correspond à une mise en commun d'un motif.

Dans cet exemple, on cherche à obtenir un raccord géométrique d'ordre 2. On obtient ceci en établissant des raccords sur trois points de contrôle. On attribue donc à chaque sommet  $s$  un seul point de contrôle.

#### Construction des espaces de contrôle

On définit la dimension interne  $\dot{m}_a$  de manière à obtenir une courbe à sept points de contrôle. Sachant que  $a$  possède déjà trois points de contrôle pour chaque sommet, on attribue un point de contrôle interne à  $a$ .

$$\begin{aligned} \dot{m}_s &= 3 \\ \dot{m}_a &= 1 \\ m_s &= \dot{m}_s = 3 \\ m_a &= \dot{m}_a + 2m_s = 7 \end{aligned}$$

On définit les opérateurs de plongement suivants :

$$\Pi_0^a = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \Pi_1^a = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



Fig. 4.16 – Représentation des points de contrôle associés à  $a$  et à ses bords

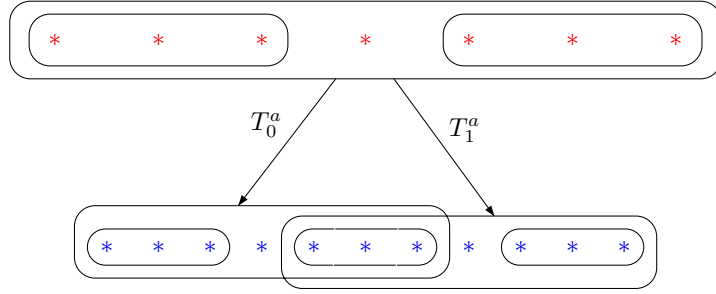


Fig. 4.17 – Représentation des points de subdivision de  $a$

### Matrices de subdivision

Avant de résoudre les contraintes matricielles, on initialise les matrices de subdivision avec un ensemble de paramètres supposés *a priori* variables et indépendants.

$$T_0^a = \begin{pmatrix} p_{0,w} & p_{1,w} & p_{2,w} & p_{3,w} & p_{4,w} & p_{5,w} & p_{6,w} \\ p_{0,x} & p_{1,x} & p_{2,x} & p_{3,x} & p_{4,x} & p_{5,x} & p_{6,x} \\ p_{0,y} & p_{1,y} & p_{2,y} & p_{3,y} & p_{4,y} & p_{5,y} & p_{6,y} \\ p_{0,z} & p_{1,z} & p_{2,z} & p_{3,z} & p_{4,z} & p_{5,z} & p_{6,z} \end{pmatrix}$$

$$T_0^a = \begin{pmatrix} a_0 & b_0 & c_0 & d_0 & e_0 & f_0 & g_0 \\ a_1 & b_1 & c_1 & d_1 & e_1 & f_1 & g_1 \\ a_2 & b_2 & c_2 & d_2 & e_2 & f_2 & g_2 \\ a_3 & b_3 & c_3 & d_3 & e_3 & f_3 & g_3 \\ a_4 & b_4 & c_4 & d_4 & e_4 & f_4 & g_4 \\ a_5 & b_5 & c_5 & d_5 & e_5 & f_5 & g_5 \\ a_6 & b_6 & c_6 & d_6 & e_6 & f_6 & g_6 \end{pmatrix}$$

$$T_1^a = \begin{pmatrix} h_0 & i_0 & j_0 & k_0 & l_0 & m_0 & n_0 \\ h_1 & i_1 & j_1 & k_1 & l_1 & m_1 & n_1 \\ h_2 & i_2 & j_2 & k_2 & l_2 & m_2 & n_2 \\ h_3 & i_3 & j_3 & k_3 & l_3 & m_3 & n_3 \\ h_4 & i_4 & j_4 & k_4 & l_4 & m_4 & n_4 \\ h_5 & i_5 & j_5 & k_5 & l_5 & m_5 & n_5 \\ h_6 & i_6 & j_6 & k_6 & l_6 & m_6 & n_6 \end{pmatrix}$$

$$T_0^s = \begin{pmatrix} o_0 & p_0 & q_0 \\ o_1 & p_1 & q_1 \\ o_2 & p_2 & q_2 \end{pmatrix}$$

La résolution des contraintes matricielles se traduit par l'identification de certains coefficients.

$$\begin{aligned}
T_0^a \Pi_1^a &= \begin{pmatrix} a_0 & b_0 & c_0 & d_0 & e_0 & f_0 & g_0 \\ a_1 & b_1 & c_1 & d_1 & e_1 & f_1 & g_1 \\ a_2 & b_2 & c_2 & d_2 & e_2 & f_2 & g_2 \\ a_3 & b_3 & c_3 & d_3 & e_3 & f_3 & g_3 \\ a_4 & b_4 & c_4 & d_4 & e_4 & f_4 & g_4 \\ a_5 & b_5 & c_5 & d_5 & e_5 & f_5 & g_5 \\ a_6 & b_6 & c_6 & d_6 & e_6 & f_6 & g_6 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} e_0 & f_0 & g_0 \\ e_1 & f_1 & g_1 \\ e_2 & f_2 & g_2 \\ e_3 & f_3 & g_3 \\ e_4 & f_4 & g_4 \\ e_5 & f_5 & g_5 \\ e_6 & f_6 & g_6 \end{pmatrix} \\
&\hat{=} \\
T_1^a \Pi_0^a &= \begin{pmatrix} h_0 & i_0 & j_0 & k_0 & l_0 & m_0 & n_0 \\ h_1 & i_1 & j_1 & k_1 & l_1 & m_1 & n_1 \\ h_2 & i_2 & j_2 & k_2 & l_2 & m_2 & n_2 \\ h_3 & i_3 & j_3 & k_3 & l_3 & m_3 & n_3 \\ h_4 & i_4 & j_4 & k_4 & l_4 & m_4 & n_4 \\ h_5 & i_5 & j_5 & k_5 & l_5 & m_5 & n_5 \\ h_6 & i_6 & j_6 & k_6 & l_6 & m_6 & n_6 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} h_0 & i_0 & j_0 \\ h_1 & i_1 & j_1 \\ h_2 & i_2 & j_2 \\ h_3 & i_3 & j_3 \\ h_4 & i_4 & j_4 \\ h_5 & i_5 & j_5 \\ h_6 & i_6 & j_6 \end{pmatrix} \\
&\hat{=} \\
T_0^a \Pi_0^a &= \begin{pmatrix} a_0 & b_0 & c_0 & d_0 & e_0 & f_0 & g_0 \\ a_1 & b_1 & c_1 & d_1 & e_1 & f_1 & g_1 \\ a_2 & b_2 & c_2 & d_2 & e_2 & f_2 & g_2 \\ a_3 & b_3 & c_3 & d_3 & e_3 & f_3 & g_3 \\ a_4 & b_4 & c_4 & d_4 & e_4 & f_4 & g_4 \\ a_5 & b_5 & c_5 & d_5 & e_5 & f_5 & g_5 \\ a_6 & b_6 & c_6 & d_6 & e_6 & f_6 & g_6 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \\ a_4 & b_4 & c_4 \\ a_5 & b_5 & c_5 \\ a_6 & b_6 & c_6 \end{pmatrix} \\
&\hat{=} \\
\Pi_0^a T_0^s &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} o_0 & p_0 & q_0 \\ o_1 & p_1 & q_1 \\ o_2 & p_2 & q_2 \end{pmatrix} = \begin{pmatrix} o_0 & p_0 & q_0 \\ o_1 & p_1 & q_1 \\ o_2 & p_2 & q_2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\
&\hat{=} \\
T_1^a \Pi_1^a &= \begin{pmatrix} h_0 & i_0 & j_0 & k_0 & l_0 & m_0 & n_0 \\ h_1 & i_1 & j_1 & k_1 & l_1 & m_1 & n_1 \\ h_2 & i_2 & j_2 & k_2 & l_2 & m_2 & n_2 \\ h_3 & i_3 & j_3 & k_3 & l_3 & m_3 & n_3 \\ h_4 & i_4 & j_4 & k_4 & l_4 & m_4 & n_4 \\ h_5 & i_5 & j_5 & k_5 & l_5 & m_5 & n_5 \\ h_6 & i_6 & j_6 & k_6 & l_6 & m_6 & n_6 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} l_0 & m_0 & n_0 \\ l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \\ l_4 & m_4 & n_4 \\ l_5 & m_5 & n_5 \\ l_6 & m_6 & n_6 \end{pmatrix} \\
&\hat{=} \\
\Pi_1^a T_0^s &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} o_0 & p_0 & q_0 \\ o_1 & p_1 & q_1 \\ o_2 & p_2 & q_2 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ o_0 & p_0 & q_0 \\ o_1 & p_1 & q_1 \\ o_2 & p_2 & q_2 \end{pmatrix}
\end{aligned}$$

Ceci permet d'obtenir une représentation formelle des matrices de subdivision et de la

matrice globale.

$$\begin{aligned}
T_0^h &= \begin{pmatrix} p_{0,w} & p_{1,w} & p_{2,w} & p_{3,w} & p_{4,w} & p_{5,w} & p_{6,w} \\ p_{0,x} & p_{1,x} & p_{2,x} & p_{3,x} & p_{4,x} & p_{5,x} & p_{6,x} \\ p_{0,y} & p_{1,y} & p_{2,y} & p_{3,y} & p_{4,y} & p_{5,y} & p_{6,y} \\ p_{0,z} & p_{1,z} & p_{2,z} & p_{3,z} & p_{4,z} & p_{5,z} & p_{6,z} \end{pmatrix} \\
T_0^a &= \begin{pmatrix} a_0 & b_0 & c_0 & d_0 & e_0 & f_0 & g_0 \\ a_1 & b_1 & c_1 & d_1 & e_1 & f_1 & g_1 \\ a_2 & b_2 & c_2 & d_2 & e_2 & f_2 & g_2 \\ 0 & 0 & 0 & d_3 & e_3 & f_3 & g_3 \\ 0 & 0 & 0 & d_4 & e_4 & f_4 & g_4 \\ 0 & 0 & 0 & d_5 & e_5 & f_5 & g_5 \\ 0 & 0 & 0 & d_6 & e_6 & f_6 & g_6 \end{pmatrix} \\
T_1^a &= \begin{pmatrix} e_0 & f_0 & g_0 & j_0 & 0 & 0 & 0 \\ e_1 & f_1 & g_1 & j_1 & 0 & 0 & 0 \\ e_2 & f_2 & g_2 & j_2 & 0 & 0 & 0 \\ e_3 & f_3 & g_3 & j_3 & 0 & 0 & 0 \\ e_4 & f_4 & g_4 & j_4 & a_0 & b_0 & c_0 \\ e_5 & f_5 & g_5 & j_5 & a_1 & b_1 & c_1 \\ e_6 & f_6 & g_6 & j_6 & a_2 & b_2 & c_2 \end{pmatrix} \\
T_0^s &= \begin{pmatrix} a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{pmatrix} \\
T_{\frac{1}{2}}^a &= \begin{pmatrix} a_0 & b_0 & c_0 & d_0 & e_0 & f_0 & g_0 & j_0 & 0 & 0 & 0 \\ a_1 & b_1 & c_1 & d_1 & e_1 & f_1 & g_1 & j_1 & 0 & 0 & 0 \\ a_2 & b_2 & c_2 & d_2 & e_2 & f_2 & g_2 & j_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & d_3 & e_3 & f_3 & g_3 & j_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & d_4 & e_4 & f_4 & g_4 & j_4 & a_0 & b_0 & c_0 \\ 0 & 0 & 0 & d_5 & e_5 & f_5 & g_5 & j_5 & a_1 & b_1 & c_1 \\ 0 & 0 & 0 & d_6 & e_6 & f_6 & g_6 & j_6 & a_2 & b_2 & c_2 \end{pmatrix}
\end{aligned}$$

## Raccord géométrique

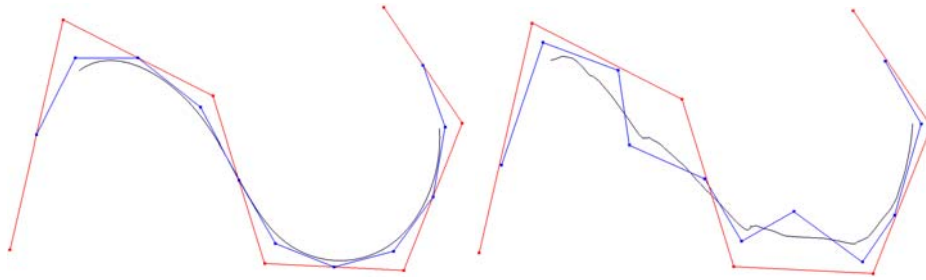
La mise en commun de plusieurs points de subdivision permet, sous certaines conditions, d'établir un certain raccord géométrique, plus fort que le simple raccord topologique. Ces conditions portent sur les valeurs propres des matrices de subdivision.

L'ensemble des valeurs propres de  $T_0^s$  est un sous-ensemble des valeurs propres de  $T_0^a$  et de  $T_1^a$ . Ces matrices homogènes, représentant des opérateurs affines contractants, ont comme plus grande valeur propre 1, car ce sont de matrices barycentriques. Le vecteur homogène associé représente le centre de chaque opérateur (leur point fixe). Les autres valeurs propres sont celles de l'opérateur linéaire associé  $L$ , et déterminent le comportement vectoriel local autour des centres  $c$  des opérateurs affines.

$$(T_0^s)^n (c + \vec{u}) = c + L^n(\vec{u})$$

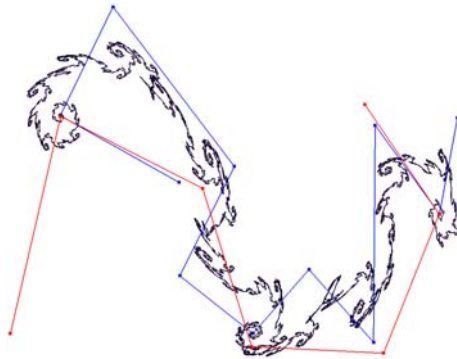
Si les valeurs propres maximales de  $T_0^a$  et de  $T_1^a$  sont celles de  $T_0^s$ , alors le comportement vectoriel local aux extrémités est le même et le raccord à l'intérieur de la courbe se fait sur ce comportement commun. Il y a plusieurs types de raccords géométriques itératifs. Nous donnons deux exemples.

Dans les premiers exemples (voir figure 4.18), les deuxièmes et troisièmes valeurs propres sont réelles et positives. La courbe est lisse. Le raccord se fait sur les deux vecteurs propres associés à  $L$ . Le premier correspond à la tangente [BGN07][BGN08]. De plus, si ces deux valeurs propres sont  $(r, r^2)$ , la fonction associée est localement  $C^2$  [DL92] [PM87] [MP89] [Pra89] [Pra98].



**Fig. 4.18** – Courbes d'ordre  $G^2$ . La courbe de droite n'est pas une courbe polynomiale ou rationnelle

Dans le deuxième exemple, les deuxièmes et troisièmes valeurs propres sont complexes conjuguées. On obtient alors un motif spiralé autour du centre des opérateurs (voir figure 4.19).



**Fig. 4.19** – Courbes dont les sommets ont un comportement spiralé

### 4.3.2 Courbe avec raccord d'ordre 2 et avec chevauchements

Un exemple de courbe plus intéressant pour l'étude des masques, est celui d'une courbe à quatre points de contrôle raccordée sur trois points de contrôle.

On obtient un type de courbe qui est une généralisation du schéma de courbes de CATMULL-CLARK.

#### Plongements

On introduit dans l'automate un nouvel état  $p$ , de dimension 1, représentant un point de contrôle. On notera également  $\pi_i$  la transition permettant d'extraire le  $i^{\text{ème}}$  point de contrôle interne d'un état  $x$ , et  $\Pi_{\pi_i}^x$  la matrice correspondante.

Les opérateurs  $\Pi_{\pi_i}^s$  effectuent l'extraction du  $i^{\text{ème}}$  point de contrôle.

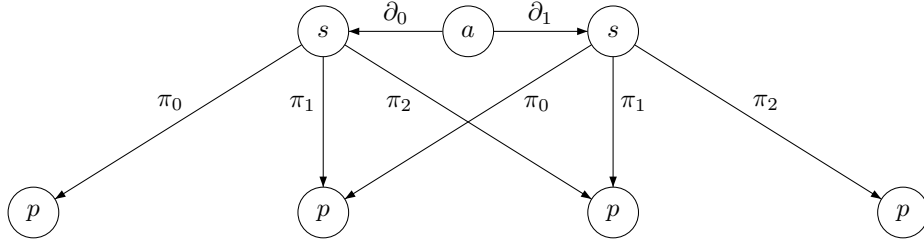
$$\Pi_{\pi_0}^s = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \Pi_{\pi_1}^s = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \Pi_{\pi_2}^s = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

On définit une certaine structure des points de contrôle de  $a$ , dans laquelle il y a un



chevauchement des plongements de  $E^s$  dans  $E^a$ .

$$\begin{aligned}\Pi_0^a \Pi_{\pi_1}^s &= \Pi_1^a \Pi_{\pi_0}^s \\ \Pi_0^a \Pi_{\pi_2}^s &= \Pi_1^a \Pi_{\pi_1}^s\end{aligned}$$



**Fig. 4.20** – Structure des points de contrôle de  $a$

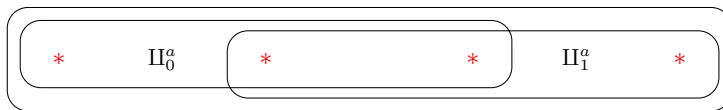
Les points de contrôle de  $a$  sont uniquement composés de ceux de  $s$ . On attribue une dimension interne nulle à  $a$ .

$$\begin{aligned}\dot{m}_s &= 3 \\ \dot{m}_a &= 0\end{aligned}$$

$$\begin{aligned}m_s &= \dot{m}_s = 3 \\ m_a &= \dot{m}_a + 2m_s - 2 = 4\end{aligned}$$

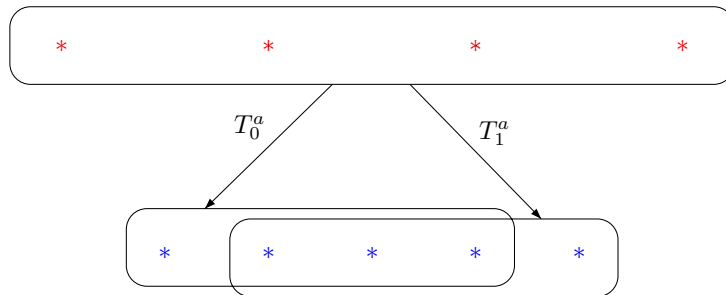
On définit l'ensemble des opérateurs de plongement conformément aux relations d'équivalence.

$$\Pi_0^a = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \quad \Pi_1^a = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



**Fig. 4.21** – Représentation des points de contrôle de  $a$

## Matrices de subdivision



**Fig. 4.22** – Représentation des points de subdivision de  $a$

On initialise les matrices de subdivision avec des paramètres formels variables et indépendants.

$$T_0^h = \begin{pmatrix} p_{0,w} & p_{1,w} & p_{2,w} & p_{3,w} \\ p_{0,x} & p_{1,x} & p_{2,x} & p_{3,x} \\ p_{0,y} & p_{1,y} & p_{2,y} & p_{3,y} \\ p_{0,z} & p_{1,z} & p_{2,z} & p_{3,z} \end{pmatrix}$$

$$T_0^a = \begin{pmatrix} a_0 & b_0 & c_0 & d_0 \\ a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \end{pmatrix}$$

$$T_1^a = \begin{pmatrix} e_0 & f_0 & g_0 & h_0 \\ e_1 & f_1 & g_1 & h_1 \\ e_2 & f_2 & g_2 & h_2 \\ e_3 & f_3 & g_3 & h_3 \end{pmatrix}$$

$$T_0^s = \begin{pmatrix} i_0 & j_0 & k_0 \\ i_1 & j_1 & k_1 \\ i_2 & j_2 & k_2 \end{pmatrix}$$

On calcule matriciellement les équations :

$$\hat{=} \left\{ \begin{array}{l} T_0^a \Pi_1^a = \begin{pmatrix} a_0 & b_0 & c_0 & d_0 \\ a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} b_0 & c_0 & d_0 \\ b_1 & c_1 & d_1 \\ b_2 & c_2 & d_2 \\ b_3 & c_3 & d_3 \end{pmatrix} \\ T_1^a \Pi_0^a = \begin{pmatrix} e_0 & f_0 & g_0 & h_0 \\ e_1 & f_1 & g_1 & h_1 \\ e_2 & f_2 & g_2 & h_2 \\ e_3 & f_3 & g_3 & h_3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} e_0 & f_0 & g_0 \\ e_1 & f_1 & g_1 \\ e_2 & f_2 & g_2 \\ e_3 & f_3 & g_3 \end{pmatrix} \end{array} \right.$$

$$\hat{=} \left\{ \begin{array}{l} T_0^a \Pi_0^a = \begin{pmatrix} a_0 & b_0 & c_0 & d_0 \\ a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix} \\ \Pi_0^a T_0^s = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} i_0 & j_0 & k_0 \\ i_1 & j_1 & k_1 \\ i_2 & j_2 & k_2 \end{pmatrix} = \begin{pmatrix} i_0 & j_0 & k_0 \\ i_1 & j_1 & k_1 \\ i_2 & j_2 & k_2 \\ 0 & 0 & 0 \end{pmatrix} \end{array} \right.$$

$$\hat{=} \left\{ \begin{array}{l} T_1^a \Pi_1^a = \begin{pmatrix} e_0 & f_0 & g_0 & h_0 \\ e_1 & f_1 & g_1 & h_1 \\ e_2 & f_2 & g_2 & h_2 \\ e_3 & f_3 & g_3 & h_3 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} f_0 & g_0 & h_0 \\ f_1 & g_1 & h_1 \\ f_2 & g_2 & h_2 \\ f_3 & g_3 & h_3 \end{pmatrix} \\ \Pi_1^a T_0^s = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} i_0 & j_0 & k_0 \\ i_1 & j_1 & k_1 \\ i_2 & j_2 & k_2 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ i_0 & j_0 & k_0 \\ i_1 & j_1 & k_1 \\ i_2 & j_2 & k_2 \end{pmatrix} \end{array} \right.$$

Après identification des coefficients correspondants, on obtient une représentation formelle des opérateurs de subdivision et de la matrice globale de  $a$ .

$$T_0^h = \begin{pmatrix} p_{0,w} & p_{1,w} & p_{2,w} & p_{3,w} \\ p_{0,x} & p_{1,x} & p_{2,x} & p_{3,x} \\ p_{0,y} & p_{1,y} & p_{2,y} & p_{3,y} \\ p_{0,z} & p_{1,z} & p_{2,z} & p_{3,z} \end{pmatrix}$$

$$T_0^s = \begin{pmatrix} a_0 & b_0 & 0 \\ a_1 & b_1 & a_0 \\ 0 & b_2 & a_1 \end{pmatrix}$$

$$T_0^a = \begin{pmatrix} a_0 & b_0 & 0 & 0 \\ a_1 & b_1 & a_0 & b_0 \\ 0 & b_2 & a_1 & b_1 \\ 0 & 0 & 0 & b_2 \end{pmatrix}$$

$$T_1^a = \begin{pmatrix} b_0 & 0 & 0 & 0 \\ b_1 & a_0 & b_0 & 0 \\ b_2 & a_1 & b_1 & a_0 \\ 0 & 0 & b_2 & a_1 \end{pmatrix}$$

$$T_{\div}^a = \begin{pmatrix} a_0 & b_0 & 0 & 0 & 0 \\ a_1 & b_1 & a_0 & b_0 & 0 \\ 0 & b_2 & a_1 & b_1 & a_0 \\ 0 & 0 & 0 & b_2 & a_1 \end{pmatrix}$$

Une représentation de ces matrices sur le graphe quotient est donnée sur la figure 4.23.

### Lien avec les schémas de subdivision

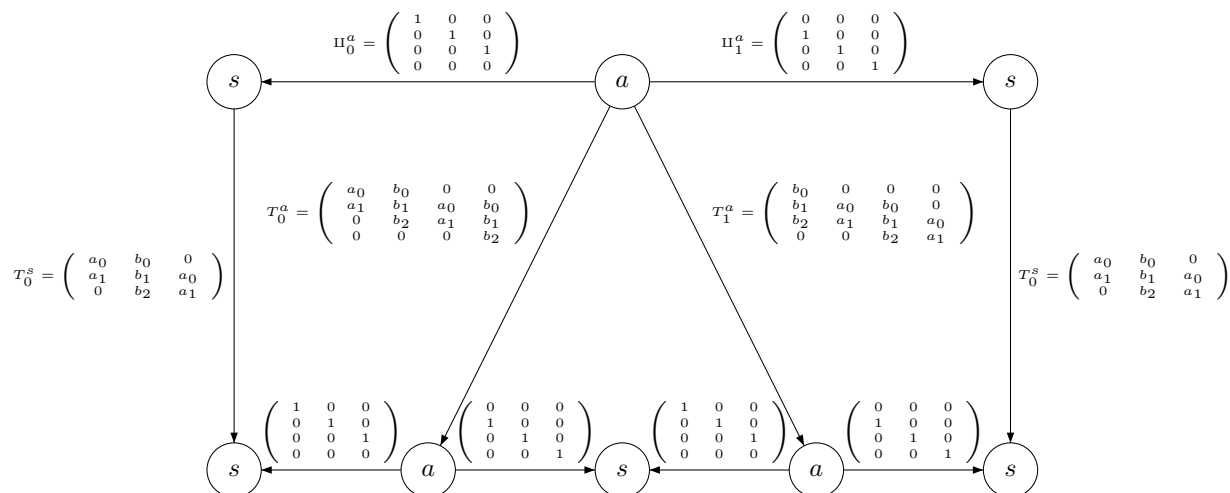


Fig. 4.23 – Forme générale des opérateurs munis des contraintes du BCIFS

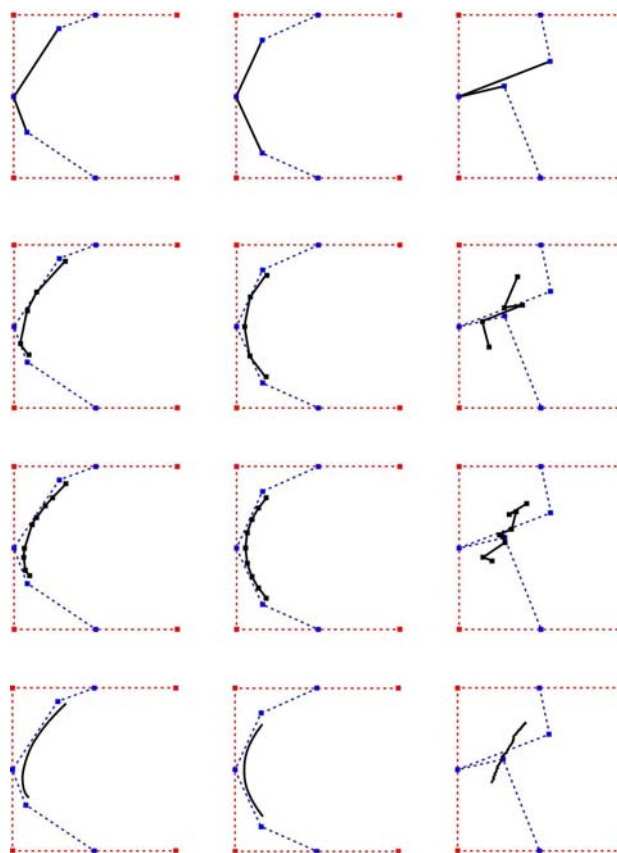


Fig. 4.24 – Variations de la courbe selon le nombre d'itérations et les paramètres de subdivision. Le schéma de CATMULL-CLARK est représenté sur la colonne centrale.

Les paramètres de subdivision sont réduits à cinq paramètres, correspondant aux masques  $(a_0, a_1)^T$  et  $(b_0, b_1, b_2)^T$ .

Les courbes de CATMULL-CLARK sont un cas particulier de celles-ci, avec des masques fixés aux valeurs suivantes :

$$\begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}$$

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} 1/8 \\ 6/8 \\ 1/8 \end{pmatrix}$$

On retrouve alors la matrice globale suivante :

$$T_{\div}^a = \begin{pmatrix} 1/2 & 1/8 & 0 & 0 & 0 \\ 1/2 & 6/8 & 1/2 & 1/8 & 0 \\ 0 & 1/8 & 1/2 & 6/8 & 1/2 \\ 0 & 0 & 0 & 1/8 & 1/2 \end{pmatrix}$$

### 4.3.3 Courbe symétrique

On reprend ici le premier exemple d'une courbe à trois points de contrôles et deux subdivisions décrit dans la section 4.2.3, mais en introduisant ici une contrainte de symétrie.

L'opérateur  $S_{\div}^a$  inversant l'orientation de  $a$  permute les extrémités et conserve le point interne (voir figure 4.25) :

$$S_{\div}^a = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

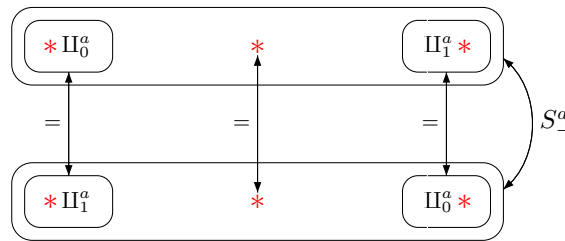


Fig. 4.25 – Mise en correspondance des points de contrôle de  $a$  avec son symétrique

La contrainte de symétrie est décrite par l'équation  $T_0^a S_{\div}^a = S_{\div}^a T_1^a$ . Sans cette équation, on rappelle que la forme générale des opérateurs  $T_0^a$  et  $T_1^a$  est :

$$T_0^a = \begin{pmatrix} a_0 & b_0 & c_0 \\ 0 & b_1 & c_1 \\ 0 & b_2 & c_2 \end{pmatrix} \quad T_1^a = \begin{pmatrix} c_0 & e_0 & 0 \\ c_1 & e_1 & 0 \\ c_2 & e_2 & a_0 \end{pmatrix}$$

L'équation  $T_0^a S_-^a = S_-^a T_1^a$  s'exprime formellement sous forme matricielle :

$$\hat{=} \begin{cases} T_0^a S_-^a = \begin{pmatrix} a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} c_0 & b_0 & a_0 \\ c_1 & b_1 & 0 \\ c_2 & b_2 & 0 \end{pmatrix} \\ S_-^a T_1^a = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} c_0 & e_0 & 0 \\ c_1 & e_1 & 0 \\ c_2 & e_2 & a_0 \end{pmatrix} = \begin{pmatrix} c_2 & e_2 & a_0 \\ c_1 & e_1 & 0 \\ c_0 & e_0 & 0 \end{pmatrix} \end{cases}$$

On obtient des identifications entre certains coefficients.

$$\begin{aligned} b_0 &\hat{=} e_2 & c_0 &\hat{=} c_2 \\ b_1 &\hat{=} e_1 \\ b_2 &\hat{=} e_0 \end{aligned}$$

Cela donne la forme générale des opérateurs  $T_0^a$  et  $T_1^a$  :

$$T_0^a = \begin{pmatrix} a_0 & b_0 & c_0 \\ 0 & b_1 & c_1 \\ 0 & b_2 & c_0 \end{pmatrix} \quad T_1^a = \begin{pmatrix} c_0 & b_2 & 0 \\ c_1 & b_1 & 0 \\ c_0 & b_0 & a_0 \end{pmatrix}$$

et de la matrice globale  $T_{\frac{\cdot}{\cdot}}^a$  :

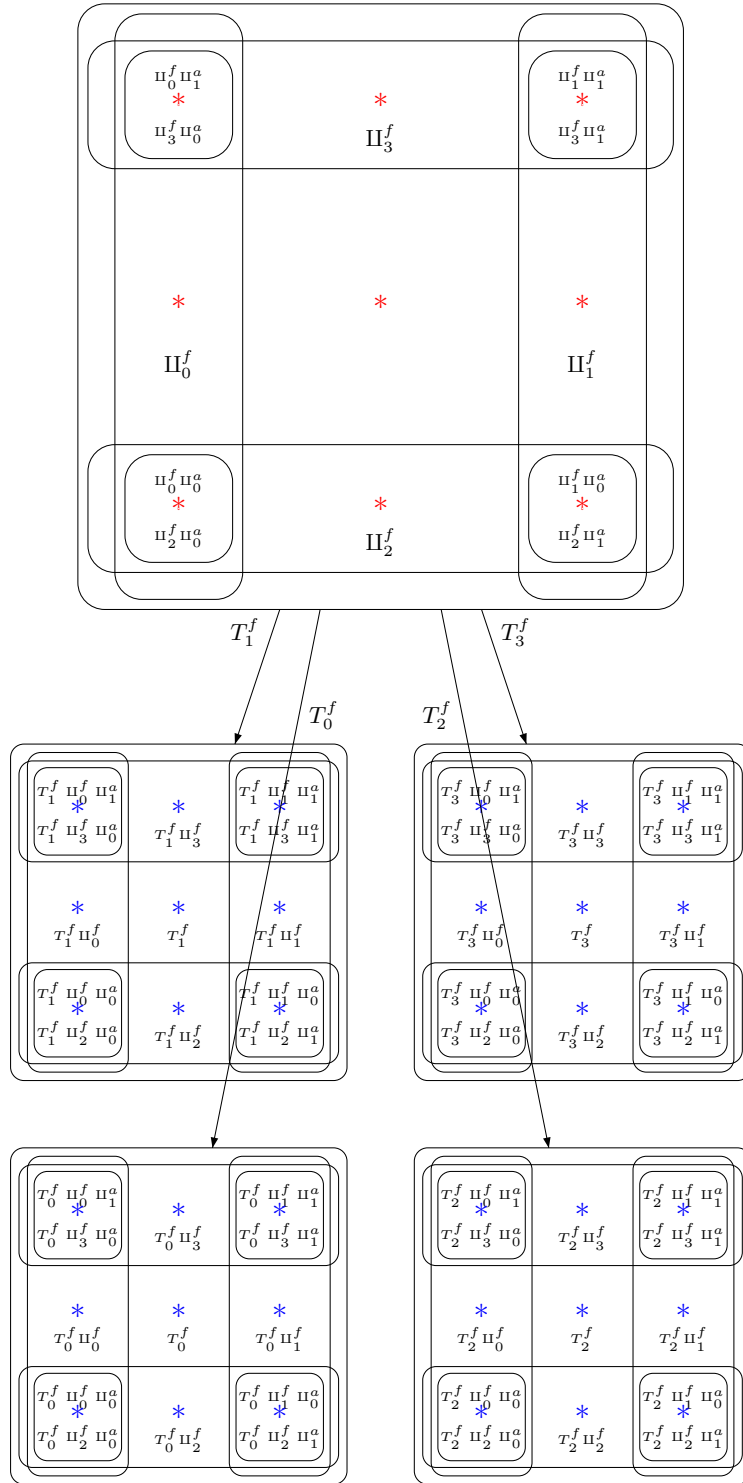
$$T_{\frac{\cdot}{\cdot}}^a = \begin{pmatrix} a_0 & b_0 & c_0 & b_2 & 0 \\ 0 & b_1 & c_1 & b_1 & 0 \\ 0 & b_2 & c_0 & b_0 & a_0 \end{pmatrix}$$

#### 4.3.4 Carreau quadrangulaire

On reprend ici l'exemple du carreau, dont on a étudié la construction de l'espace de contrôle dans la partie 4.2.1. On étudie ici les contraintes sur les opérateurs de subdivision. On ne présente plus ici la résolution des équations sous forme matricielle, mais graphiquement sous forme de grille de points.

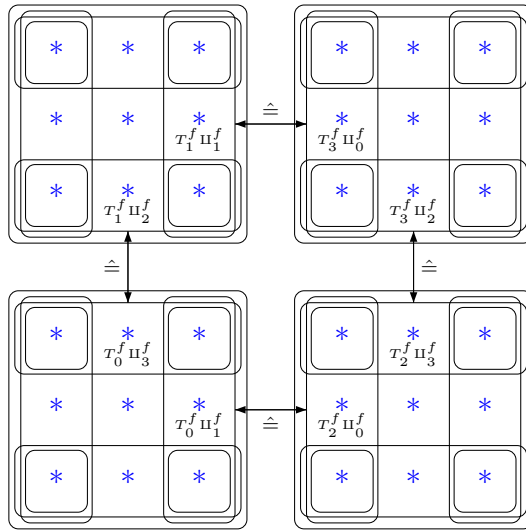
##### Description des contraintes

La face  $f$  possède 9 points de contrôle, munis d'une structure d'incidence par les opérateurs de plongement  $\Pi_i^f$ . La face se subdivise en quatre faces, ce qui génère quatre ensembles de 9 points de subdivision munis de la même structure d'incidence (voir figure 4.26).

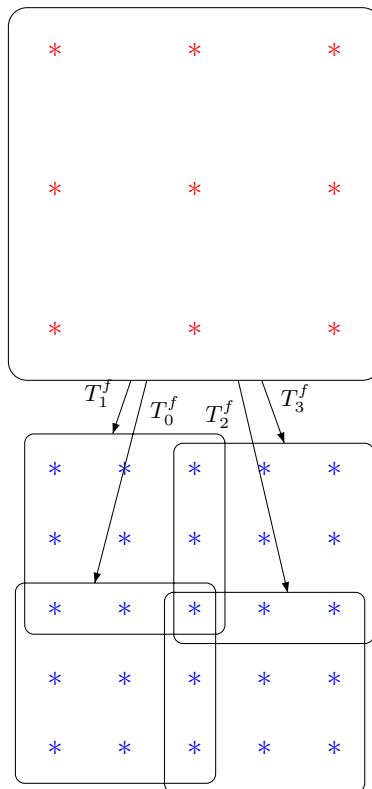


**Fig. 4.26** – Représentation des points de subdivision associés aux opérateurs  $T_i^f$

Les équations d'adjacence sur les opérateurs de subdivision établissent l'identification de certains points de subdivision, et réduisent leur nombre total à 25 points de subdivision (voir figures 4.27 et 4.28).



**Fig. 4.27** – Raccord des côtés du carreau : équations d'adjacence sur les opérateurs  $T_i^f$

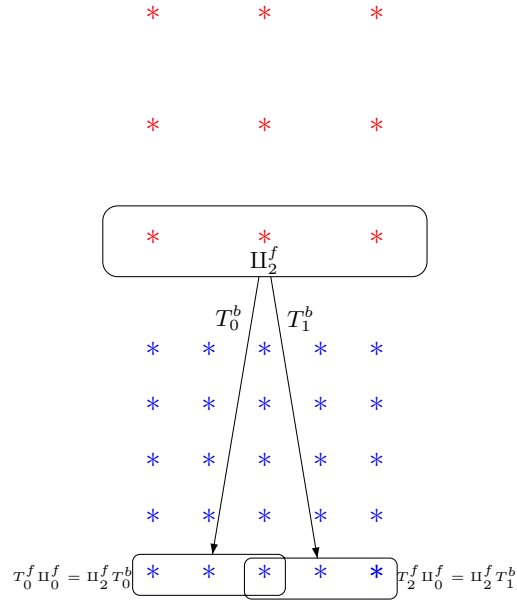


**Fig. 4.28** – Équations d'adjacence : mise en commun des points de subdivision associés aux opérateurs  $T_i^f$

Les équations d'incidence déterminent certains masques de subdivision.

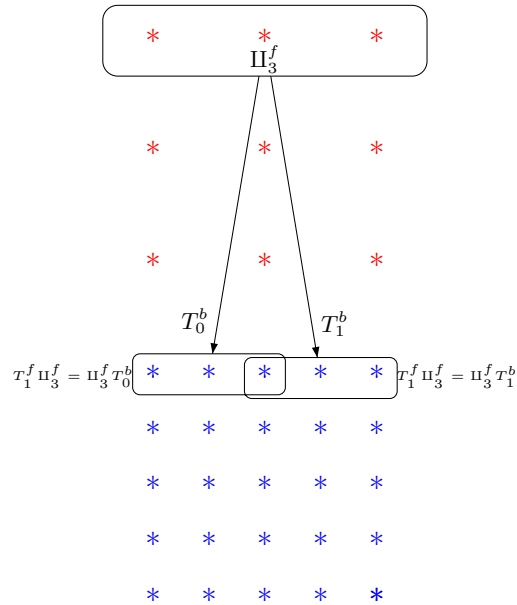
Les 5 points de subdivision du bord inférieur correspondent aux points de subdivision d'une arête  $b$  dont les points de contrôle sont les trois points du bord  $\partial_2$  de  $f$  (voir figure 4.29).





**Fig. 4.29** – Équations d'incidence associées à  $\Pi_2^f$

De même, les 5 points de subdivision du bord supérieur correspondent aux points de subdivision d'une arête  $b$  dont les points de contrôle sont les trois points du bord  $\partial_3$  de  $f$  (voir figure 4.30).



**Fig. 4.30** – Équations d'incidence associées à  $\Pi_3^f$

Il en est de même pour les deux colonnes de cinq points de subdivision de gauche et de droite. Chacune est définie par rapport à une colonne de trois points de contrôle par l'intermédiaire des opérateurs  $T_0^a$  et  $T_1^a$ .

Enfin, chaque sommet du premier niveau de subdivision correspond à la subdivision d'un sommet  $s$  de la face initiale (voir figure 4.31).

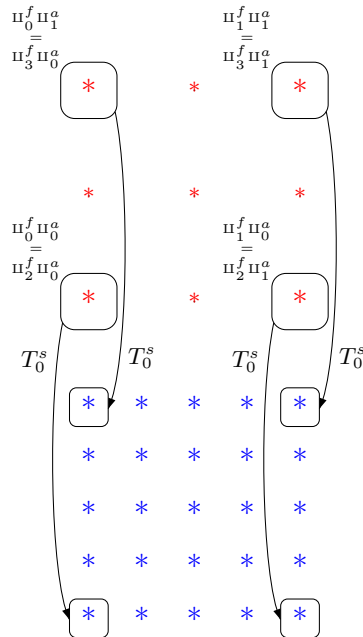


Fig. 4.31 – Représentation des points de subdivision de  $f$  associés aux sommets

### Contrôle de l'aspect

Comme nous l'avons vu dans la partie topologie, les carreaux sont bordés par quatre arêtes correspondant à deux types d'arcs, définis par deux IFS ( $T_i^a$  et  $T_i^b$ ). Si les paramètres de subdivision des carreaux sont définis par produit tensoriel [MP87] [ZT96] :

$$T_{i,j}^f = T_i^a \otimes T_j^b$$

les carreaux ont un aspect correspondant aux arcs de bord. La figure 4.32 montre un exemple d'un carreau de surface obtenu par produit tensoriel. La courbe  $a$  est définie comme un arc de BÉZIER et la courbe  $b$  une courbe de TAKAGI. La figure 4.33 est un carreau de BÉZIER, qui est un produit tensoriel de deux arcs de BÉZIER.

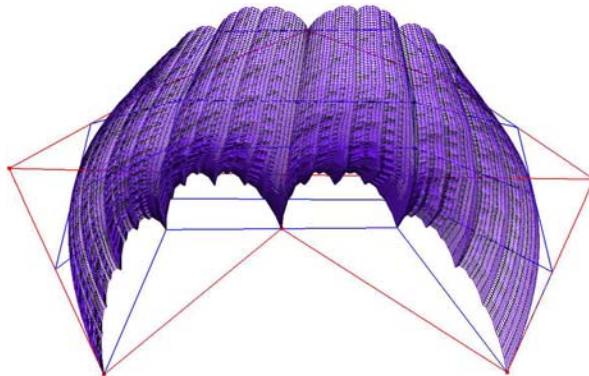
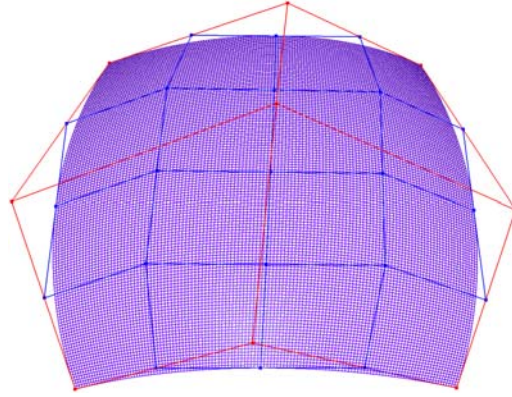


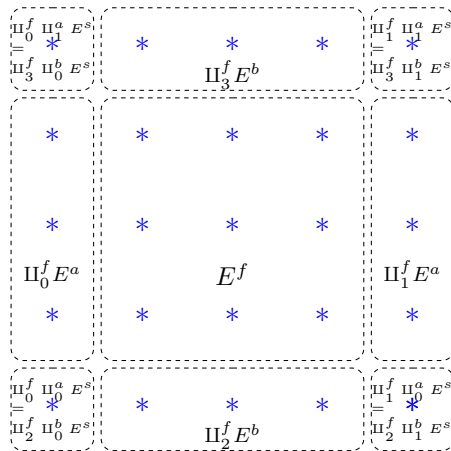
Fig. 4.32 – Carreau de surface obtenu par produit tensoriel entre une courbe de BÉZIER et une courbe de TAKAGI



**Fig. 4.33** – Carreau de BÉZIER : il est défini par produit tensoriel d’arcs de BÉZIER

La définition par contrainte est plus générale [Tos99a] [TGB02]. Les carreaux peuvent avoir un aspect différent entre le bord et le milieu (voir figures 4.35, 4.36 et 4.37).

Chaque point de subdivision correspond à un certain masque de subdivision, et dépend d’un certain sous-ensemble de points de contrôle. La figure 4.34 représente une classification des points de subdivision, selon le sous-ensemble de points de contrôle dont ils sont dépendants. Ces points de subdivision sont donc contraints d’appartenir à un certain sous-espace de  $E^f$ . Les sommets sont fixés, les bords sont contrôlés par les masques internes aux arcs et le milieu est contrôlé par les masques internes aux faces.

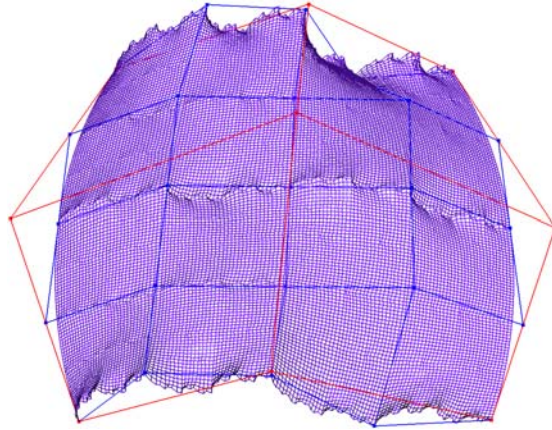


**Fig. 4.34** – Classification des points de subdivision de  $f$  selon le sous-espace auquel ils appartiennent

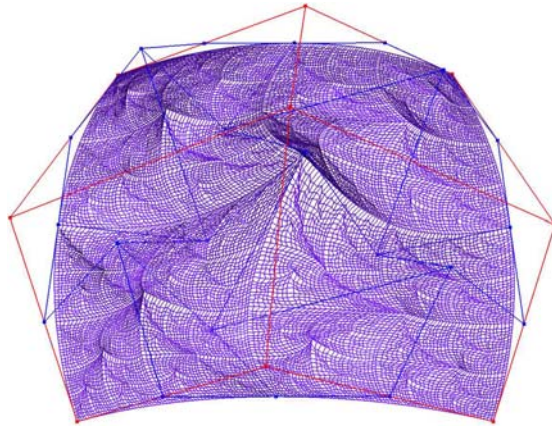
Sur la figure 4.35, les paramètres de subdivision du bord  $a$  et internes à la face sont ceux d’un carreau de BÉZIER. Les paramètres de subdivision de  $b$  ont été modifiés de manière à générer une courbe fractale. Ce motif apparaît sur les bords inférieurs et supérieurs du carreau, et se répète à l’intérieur du carreau lui-même.

Sur la figure 4.36, seuls les paramètres de subdivision de  $a$  et  $b$  sont ceux d’une courbe de BÉZIER. La surface n’est pas un carreau de BÉZIER mais contient une infinité de courbes de BÉZIER.

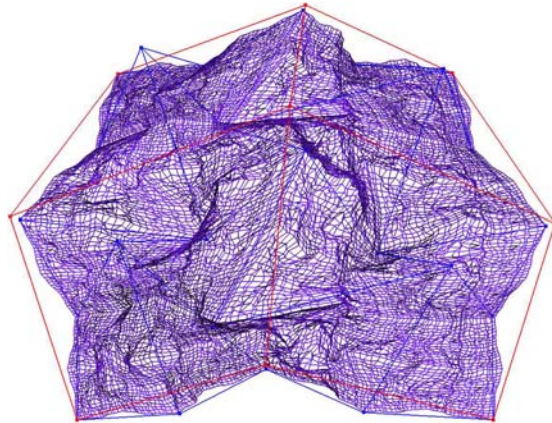
Enfin, la figure 4.37 est un exemple de carreau de surface dont tous les paramètres de subdivision sont quelconques.



**Fig. 4.35** – Carreau de surface dont un bord est un arc de BÉZIER, l'autre bord un arc fractal et les masques internes aux faces sont ceux d'un carreau de BÉZIER



**Fig. 4.36** – Carreau de surface dont les bords sont des courbes de BÉZIER et les masques internes aux faces sont différents de ceux du carreau de BÉZIER



**Fig. 4.37** – Carreau de surface avec des opérateurs de subdivision quelconques

### 4.3.5 Planaire de HILBERT

La courbe de HILBERT correspond à une suite de lignes brisées qui converge vers un carreau surfacique. Pour décrire cette courbe dans le formalisme BCIFS, on définit dans un premier temps les propriétés du carreau de surface qui constitue l'attracteur, et on complète la description en définissant les propriétés de la courbe qui parcourt la surface.

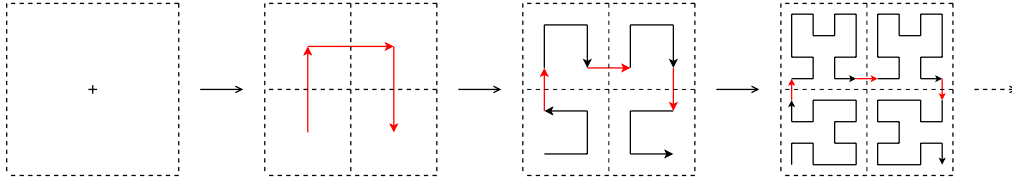


Fig. 4.38 – Schéma de la courbe de HILBERT

#### Description de la structure du carreau

Les propriétés du carreau sont décrites par les états  $f$ ,  $a$ , et  $s$  représentant les faces, arêtes et sommets.

En l'infini, la courbe de HILBERT est une fonction continue parcourant entièrement la surface  $f$ , et dont les points de départ et d'arrivée sont deux sommets  $s$ . La face  $f$  se subdivise en quatre faces. Les faces subdivisées sont orientées de manière à préserver la continuité de la courbe (voir figure 4.39).

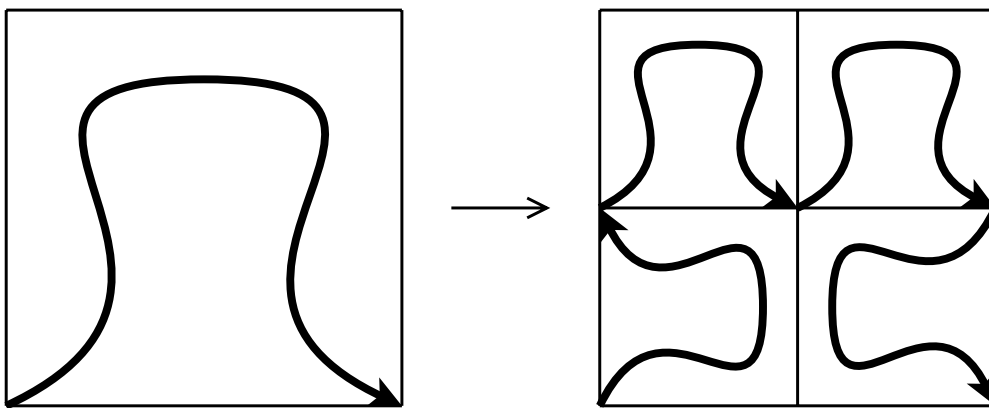


Fig. 4.39 – Sens de parcours de la courbe dans la face et ses quatre sous-faces

L'orientation des faces subdivisées étant déterminée, on établit la description BCIFS de  $f$  comme un carreau de surface (voir figure 4.40).

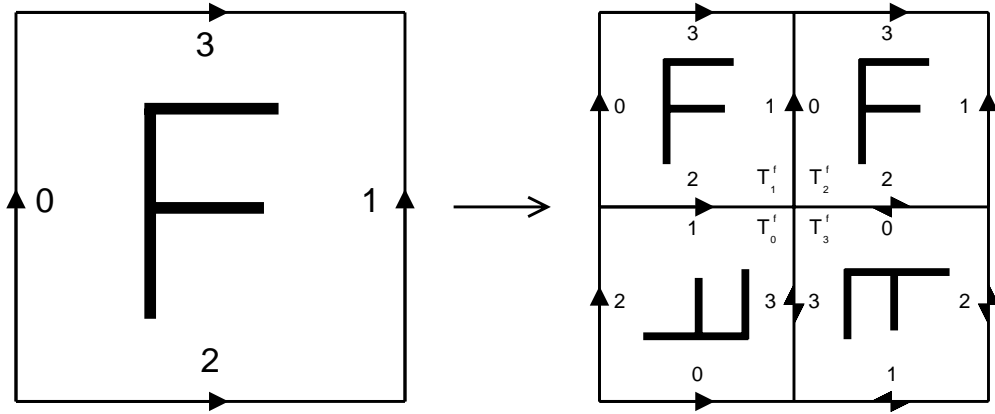


Fig. 4.40 – Subdivision de la face, avec changements d'orientation

Sur ce schéma, on remarque que le choix d'avoir un seul type d'arête  $a$  pour les quatre bords n'est pas nécessaire, étant données les arêtes mises en commun, mais qu'il est seulement nécessaire d'avoir le même type d'arête pour les bords  $\partial_0$ ,  $\partial_1$  et  $\partial_2$ . Nous donnons le même type au bord  $\partial_3$  uniquement pour simplifier la description.

L'orientation des faces subdivisées que l'on a établie implique la mise en commun d'arêtes orientées différemment entre elles. Il est donc nécessaire de définir une involution  $S_-^a$  qui inverse l'orientation des arêtes

$$\Pi_0^a = S_-^a \Pi_1^a$$

et d'établir une contrainte de symétrie sur  $a$

$$S_-^a T_0^a = T_1^a S_-^a$$

On définit également la structure topologique de la courbe  $a$ .

$$\Pi_0^a T_0^s = T_0^a \Pi_0^a \quad T_0^a \Pi_1^a = T_1^a \Pi_0^a \quad \Pi_1^a T_0^s = T_1^a \Pi_1^a$$

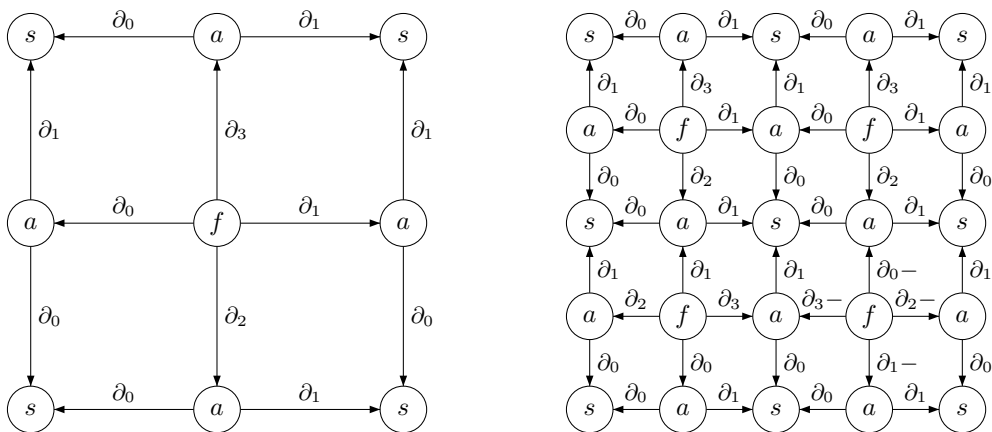
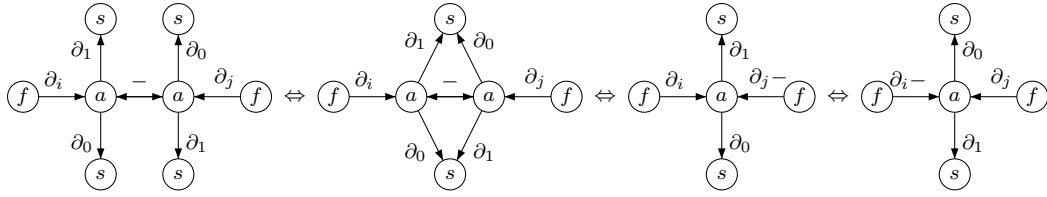


Fig. 4.41 – Topologie de la surface, au niveau 0 puis au niveau 1

On représente la structure de raccord aux niveaux de subdivision 0 et 1 sur la figure 4.41. Ce graphe utilise une représentation simplifiée pour les mises en commun d'arêtes orientées différemment, illustrée sur la figure 4.42.



**Fig. 4.42** – Représentations simplifiées de la mise en commun d'arêtes orientées différemment

La structure de raccord de niveau 0 établit les équations d'adjacence sur les opérateurs d'incidence :

$$\begin{aligned} \Pi_0^f \Pi_1^a &= \Pi_3^f \Pi_0^a & \Pi_1^f \Pi_1^a &= \Pi_3^f \Pi_1^a \\ \Pi_0^f \Pi_0^a &= \Pi_2^f \Pi_0^a & \Pi_1^f \Pi_0^a &= \Pi_2^f \Pi_1^a \end{aligned}$$

La structure de raccord de niveau 1 établit les équations d'adjacence sur les subdivisions de  $f$ .

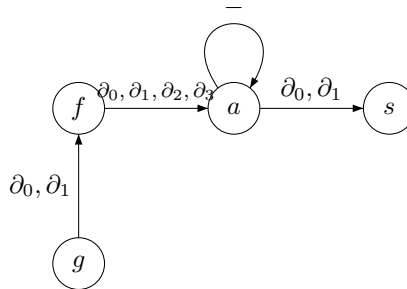
$$\begin{aligned} T_0^f \Pi_1^f &= T_1^f \Pi_2^f & T_1^f \Pi_1^f &= T_2^f \Pi_0^f & T_2^f \Pi_2^f &= T_3^f \Pi_0^f S_-^a \\ T_0^f \Pi_3^f &= T_3^f \Pi_3^f S_-^a \end{aligned}$$

La mise en correspondance des bords de la face, entre le niveau 0 et le niveau 1 établit les équations d'incidence de  $f$ .

$$\begin{aligned} \Pi_0^f T_1^a &= T_1^f \Pi_0^f & \Pi_3^f T_0^a &= T_1^f \Pi_3^f & \Pi_3^f T_1^a &= T_2^f \Pi_3^f & \Pi_1^f T_1^a &= T_2^f \Pi_1^f \\ \Pi_0^f T_0^a &= T_0^f \Pi_2^f & \Pi_2^f T_0^a &= T_0^f \Pi_0^f & \Pi_2^f T_1^a &= T_3^f \Pi_1^f S_-^a & \Pi_1^f T_0^a &= T_3^f \Pi_2^f S_-^a \end{aligned}$$

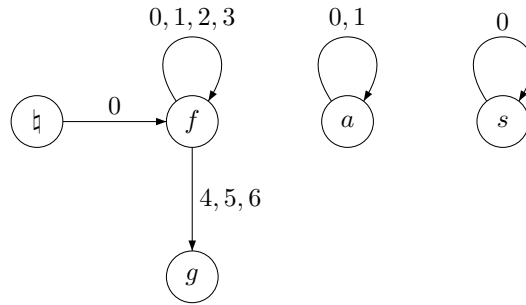
### Description de la courbe

On associe l'état  $g$  à chaque segment reliant deux faces. Chacune de ces deux faces définit une extrémité du segment. On établit donc deux relations d'incidence de  $g$  vers  $f$  (voir figure 4.43).



**Fig. 4.43** – Graphe des incidences et des permutations

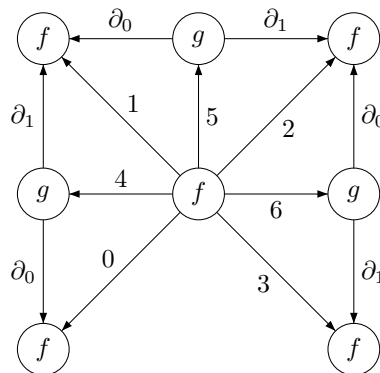
Sur les figures 4.38 et 4.46, on observe qu'à chaque étape de subdivision, une face génère trois nouveaux segments, et un segment génère un segment plus petit. On établit donc trois relations de subdivision de  $f$  vers  $g$ , et une de  $g$  vers  $g$  (voir figure 4.43).



**Fig. 4.44** – Graphe des subdivisions

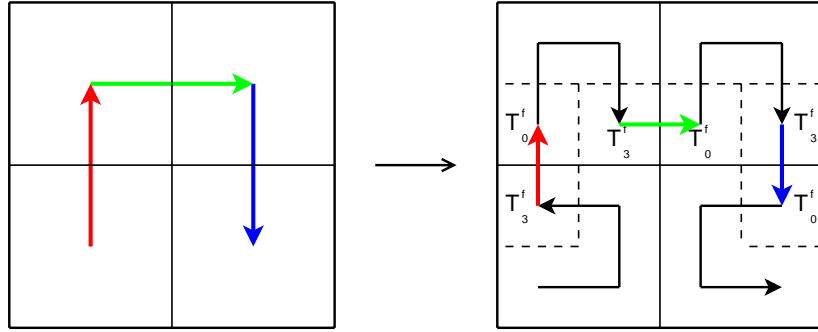
Chacun des trois segments  $g$  issus de la subdivision de  $f$  relie une certaine face à une autre (voir figure 4.45), ce qui se traduit par un ensemble d'équations d'adjacence.

$$\begin{aligned}
 T_4^f \Pi_1^g &= T_1^f = T_5^f \Pi_0^g & T_5^f \Pi_1^g &= T_2^f = T_6^f \Pi_0^g \\
 T_4^f \Pi_0^g &= T_0^f & T_6^f \Pi_1^g &= T_0^f
 \end{aligned}$$



**Fig. 4.45** – Subdivision de  $f$ , avec mises en commun des bords de  $g$

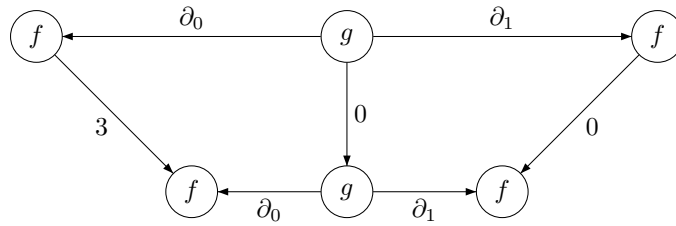




**Fig. 4.46** – *Subdivision des faces, chacune générant trois nouveaux segments, et subdivision des segments, chacune générant un segment plus petit. Le bord  $\partial_0$  d'un nouveau segment correspond à la subdivision 3 du bord  $\partial_0$  du segment initial, et le bord  $\partial_1$  d'un nouveau segment correspond à la subdivision 1 du bord  $\partial_1$  du segment initial.*

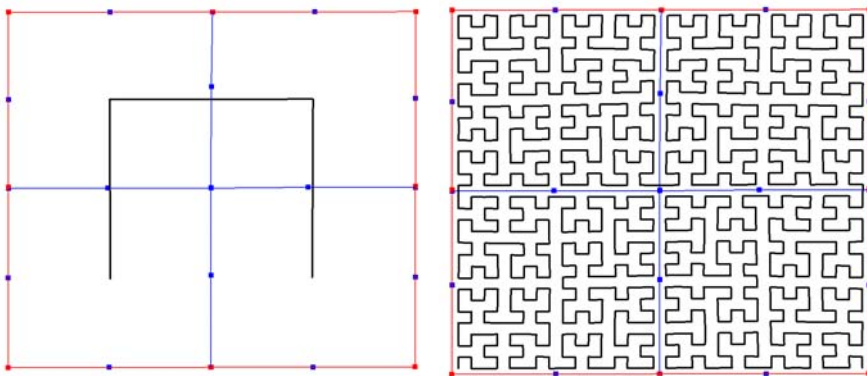
On établit enfin les équations d'incidence de  $g$ . Sur la figure 4.46, on note que le bord  $\partial_0$  de  $g$  se subdivise selon la subdivision 3 de  $f$ , et que le bord  $\partial_1$  de  $g$  se subdivise selon la subdivision 0 de  $f$  (voir figure 4.47).

$$T_0^g \Pi_0^g = \Pi_0^g T_3^f \quad T_0^g \Pi_1^g = \Pi_1^g T_0^f$$

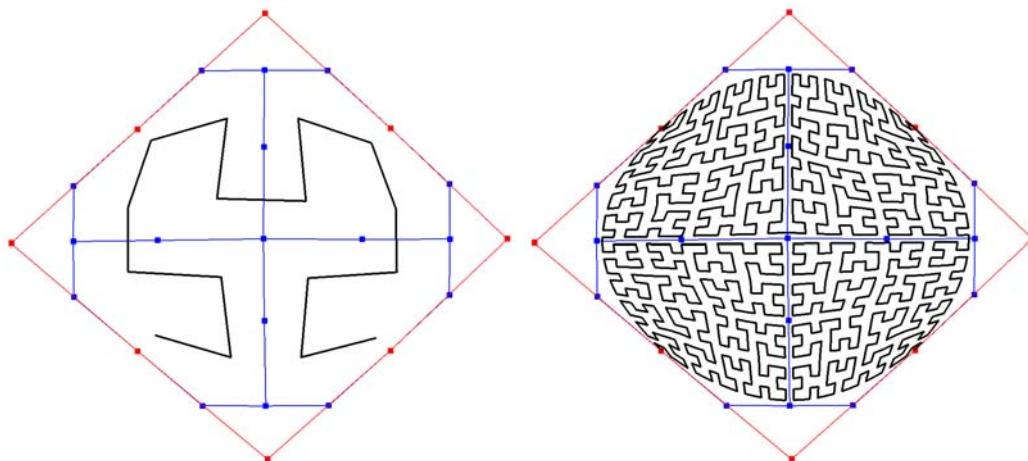


**Fig. 4.47** – *Subdivision de  $g$  avec mise en commun de ses bords  $f$*

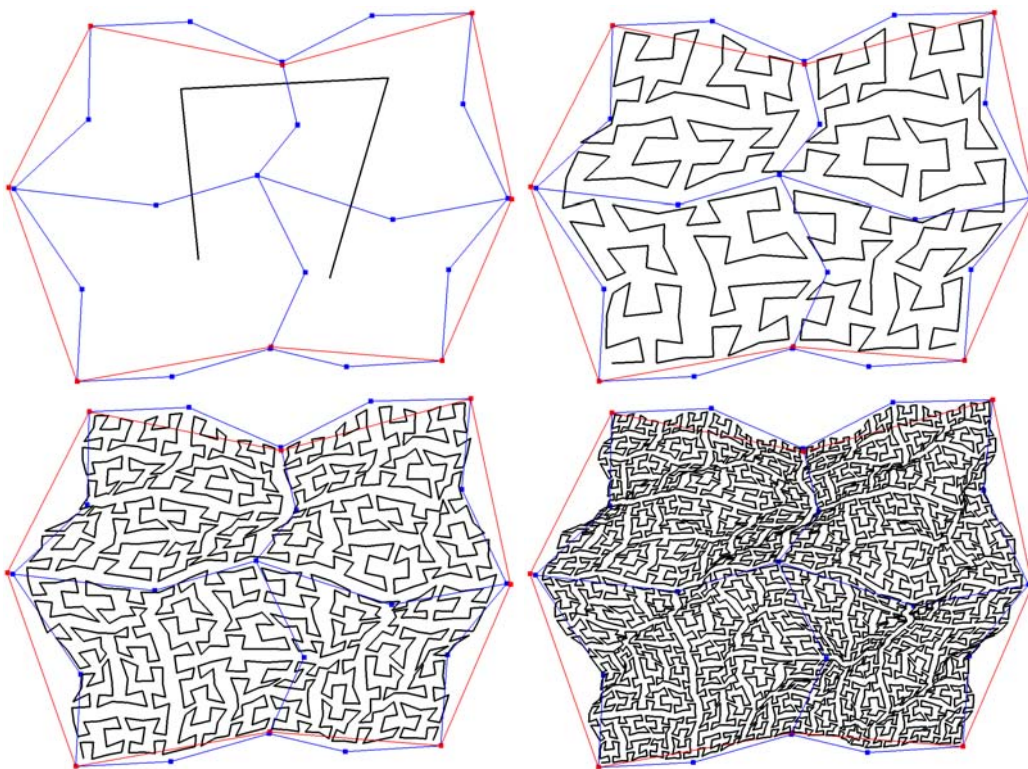
La primitive  $K^g$  est définie comme étant le segment reliant  $\Pi_0^g K^f$  et  $\Pi_1^g K^f$ , où  $K^f$  est un point défini dans  $E^f$ .



**Fig. 4.48** – *Courbe de HILBERT "standard", aux niveaux de subdivision 1 et 5*



**Fig. 4.49** – Courbe de HILBERT dont les bords a sont des courbes de BÉZIER

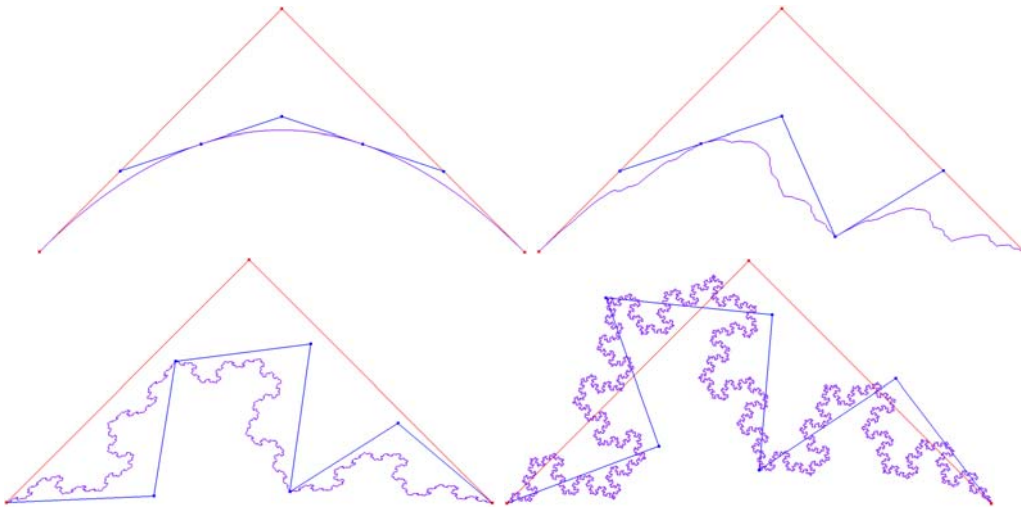


**Fig. 4.50** – Exemple d'une courbe de HILBERT ayant des paramètres de subdivision quelconques

## 4.4 Saisie interactive

Les logiciels de CAO classiques utilisent des schémas de subdivision prédéfinis pour générer des formes lisses. L'utilisateur contrôle l'aspect global de ces formes en positionnant ou en déplaçant, dans la scène, un certain nombre de points de contrôle. Dans notre formalisme, ce type de saisie revient à associer un point de contrôle pour chaque colonne de  $T_{\frac{h}{2}}^h$ , qui agit sur l'aspect global de la forme, qu'elle soit lisse ou fractale.

Nous utilisons un procédé analogue pour contrôler la matrice  $T_{\frac{x}{2}}^x$ , qui agit à chaque niveau de subdivision, ce qui permet notamment de modifier un schéma initialement lisse, pour le rendre fractal (voir figure 4.51).



**Fig. 4.51** – Saisie interactive des paramètres de subdivision d'une courbe, par manipulation des points de subdivision (en bleu). Les points rouges représentent les points de contrôle. Trois opérateurs affines  $T_0^a$ ,  $T_1^a$  et  $T_2^a$  sont utilisés sur cet exemple. Chacun de ces opérateurs associe les trois points de contrôle, à un ensemble de trois points de subdivision consécutifs. Le déplacement des points de subdivision entraîne la mise à jour des opérateurs de subdivision associés à ce point [TBSS\* 07].

### 4.4.1 Matrice globale

À chaque colonne  $q_j^x = T_{\frac{x}{2}}^x e_j$ , on associe dans la scène un point de subdivision  $p_j^x = T^h q_j^x$

Bien qu'ayant des caractéristiques communes, on fait une distinction entre les points de contrôle  $p_j^h = T^h e_j^x$  et les points de subdivision car ceux-ci nécessitent des traitements particuliers pour que leur saisie dans la scène soit possible.

Leur différence vient, d'une part, de la nature de l'espace dans lequel leur information est codée. En effet, chaque point de contrôle est une colonne de  $T_{\frac{h}{2}}^h$ , ce qui correspond directement à un point de la scène, tandis que chaque point de subdivision est une colonne de  $T_{\frac{x}{2}}^x$ , ce qui correspond à un point de  $E^x = B(\mathbb{R}^{J_x})$ . Chaque point de subdivision  $p_j$  est affiché dans  $E^h$  selon la projection par  $T_{\frac{h}{2}}^h$  de la colonne correspondante.

$$p_j = T_{\frac{h}{2}}^h q_j^x$$

#### 4.4.2 Saisie de points

Pour alléger les notations, on note  $q$  la colonne associée au point de subdivision  $p$ , et  $T$  la matrice de projection dans la scène.

$$p = Tq$$

Ce changement d'espace nécessite également un traitement particulier pour effectuer le calcul inverse,  $T$  en fonction de  $p$ , lorsque l'utilisateur déplace le point  $p$  dans la scène.  $q$  est défini sur  $E^x$ , qui est généralement de dimension supérieure à  $E^d$  dans lequel  $p$  est défini. L'équation  $p = Tq$  possède donc plusieurs solutions, lorsque  $p$  et  $T$  sont fixés, et il est alors nécessaire de faire un certain choix. Un procédé basique serait d'appliquer directement la pseudo-inverse de MOORE-PENROSE  $T^{(-1)}$ .

$$q = T^{(-1)}p$$

Ce procédé a pour effet de choisir la solution  $q$  qui minimise la norme  $\|q\| = \sum_i (q_i)^2$ . Cela a pour inconvénient de limiter arbitrairement les valeurs que peuvent prendre  $q$ . Par ailleurs, le déplacement infinitésimal d'un point, initialement en  $p_0$  vers une nouvelle position  $p$  entraîne une discontinuité entre les colonnes associées  $q_0$  et  $q$ , lorsque  $q_0$  ne satisfait pas lui-même le critère de moindres carrés, c'est-à-dire  $q_0 \neq T^{(-1)}Tq_0$ . Nous proposons donc de choisir une solution de l'équation  $p = Tq$ , en minimisant sa distance avec son ancienne valeur  $\|q - q_0\|$ .

$$q = q_0 + T^{(-1)}(p - p_0)$$

#### 4.4.3 Masques

Un autre aspect qui différencie les points de subdivision, par rapport aux points de contrôle, est dû aux contraintes dans certaines colonnes, qui sont notamment la présence de valeurs constantes. Le calcul vu précédemment ne doit donc s'appliquer qu'aux coefficients variables de  $q$ . On construit alors deux matrices de plongement  $\Pi_c$  et  $\Pi_v$  permettant d'extraire de  $q$  la sous-colonne  $q_c$  contenant les constantes, et  $q_v$  contenant les variables.

$$q = \Pi_c q_c + \Pi_v q_v$$

L'équation  $Tq = p$  se ramène à une équation  $T'q' = p'$  portant uniquement sur les coefficients variables de  $q$ .

$$\begin{aligned} Tq &= T(\Pi_c q_c + \Pi_v q_v) = p \\ T\Pi_v q_v &= p - \Pi_c q_c \end{aligned}$$

soit

$$T'q' = p'$$

en posant

$$\begin{aligned} T' &= T\Pi_v \\ q' &= q_v \\ p' &= p - \Pi_c q_c \end{aligned}$$

La résolution de l'équation  $T'q' = p'$  se fait en minimisant  $\|q' - q'_0\|$ .

$$q' = q'_0 + T'^{(-1)}(p' - p'_0)$$

On obtient une expression de  $q_v$ , le vecteur des paramètres variables de  $q$ , en fonction du vecteur initial  $q_{v0}$ , du point initial  $p_0$  et du point de destination  $p$ .

$$q_v = q_{v0} + (T'\Pi_v)^{(-1)}(p - p_0)$$

#### 4.4.4 Paramètres indépendants

Un dernier type de contraintes pouvant apparaître dans une colonne  $q$  est la présence de variables identifiées entre elles. On calcule alors  $\Pi_v$  de manière à ce que  $q_v$  soit réduite à l'ensemble des classes d'équivalence des coefficients variables de  $q$ .

Exemple, pour

$$q = \begin{pmatrix} a \\ b \\ c \\ b \\ a \end{pmatrix}$$

on pose

$$\Pi_v = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \quad q_v = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$



## Chapitre 5

# Contraintes géométriques

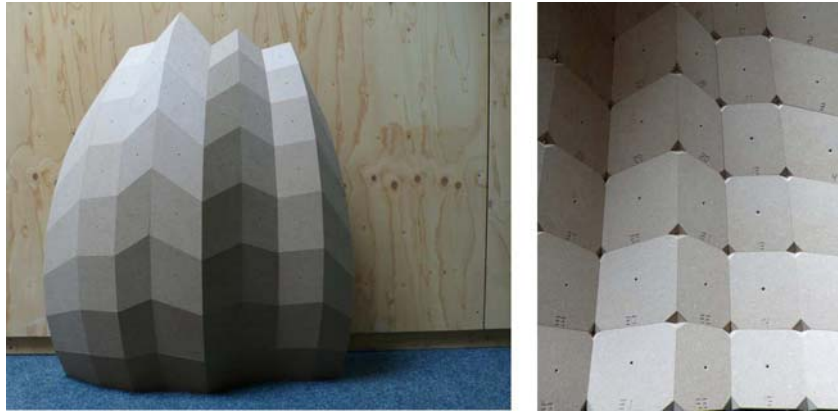


Fig. 5.1 – Maquette réalisée par assemblage de panneaux quadrangulaires plans

### 5.1 Introduction

#### 5.1.1 Contexte

Dans le cadre de la construction en bois de forme complexes, nous nous intéressons à la modélisation géométrique basée sur les modèles itératifs, et plus particulièrement, sur le modèle IFS. Ce modèle permet de représenter une grande diversité de formes, avec relativement peu de paramètres. Ce modèle itératif peut générer des formes lisses ou fractales, et généralise les schémas de subdivision classiques [TBSG\*06] [SGW08].

#### 5.1.2 Contraintes liées à la construction

Le fait que les formes modélisées soient appliquées à la construction entraîne généralement un certain nombre de contraintes géométriques. Nous souhaitons notamment réaliser des structures surfaciques par assemblage de panneaux de bois [Wei08]. Cela nécessite de modéliser des maillages composés de faces planes.

Nous nous intéressons ici à la modélisation de maillages quadrangulaires. La triangulation de maillages pour obtenir des faces planes est une solution que nous écartons. Ceci provient des contraintes liées à la construction, et plus particulièrement de la complexité à réaliser des assemblages autour de sommets de valence élevée. Les maillages triangulés, par

exemple, présentent une valence moyenne de 6 tandis que les maillages à base de carreaux ont des sommets de valence 4. L'élaboration et la réalisation des solutions constructives sont relativement coûteuses en fonction de la valence du nœud. La valence des nœuds d'un maillage a une influence directe sur la géométrie des faces : plus la valence d'un sommet est élevée, plus les angles des faces avoisinant ces sommets seront aigus. Les faces à angles aigus ne sont pas souhaitables pour une application constructive parce que les éléments constructifs ont des coins fragiles et leur manipulation lors du processus de mise en œuvre devient une opération délicate.

### 5.1.3 Travaux précédents

POTTMANN étudie l'utilisation des surfaces discrètes en architecture. Ainsi dans l'article [PBCW07], les auteurs étudient plusieurs exemples d'utilisation de maillages discrets (triangulaires, quadrangulaires et hexagonaux). En ce qui concerne les études de maillages quadrangulaires, l'équipe de POTTMANN utilise souvent la géométrie différentielle discrète. Les maillages à faces planes étudiés en [LPW\*06], [PW07] et [WWL06] sont des cas spécifiques qui représentent l'image discrète du réseau des lignes de courbure principales sur des surfaces lisses. En [PW07] et [WWL06] ces maillages sont appelés "circulaires" lorsque les quatre sommets de chaque face sont éléments d'un même cercle, et sont appelés "coniques" lorsque les normales aux faces autour d'un sommet sont sur un même cône circulaire. Les maillages coniques présentent d'ailleurs des propriétés remarquables, comme celle permettant de générer des maillages "parallèles", tels que chaque face est à la même distance de la face correspondante dans le maillage d'origine. Les propriétés des maillages circulaires à sommets coniques sont discutées en détail dans les papiers cités ci-dessus. LIU et al. [LPW\*06] présentent des résultats très intéressants obtenus par modélisation géométrique utilisant des maillages coniques et des surfaces développables. Un intérêt particulier est porté dans l'emploi de modélisation itérative par subdivision en [LPW\*06].

GLYMPH et al. [GSC\*04], définissent des modèles de surfaces discrètes par des opérateurs de balayage entre courbes discrètes. Les maillages obtenus sont constitués de trapèzes.

Bien que la méthode que nous proposons et la méthode proposée par GLYMPH [GSC\*04] produisent des maillages quadrangulaires à faces planes, elles sont en fait très différentes des maillages étudiés par POTTMANN, LIU et WANG. Les maillages que nous modélisons respectent uniquement la contrainte géométrique de la coplanarité des quatre sommets de chaque face, sans vérifier les propriétés géométriques qui sont spécifiques aux maillages coniques.

### 5.1.4 Méthode

Nous proposons une méthode basée sur un modèle itératif qui génère directement des maillages quadrangulaire à faces planes.

Nous partons d'une somme de MINKOWSKI de deux courbes. Nous étendons les possibilités de création de formes en travaillant dans un espace  $\mathbb{R}^4$  de coordonnées homogènes, et en projetant les figures obtenues dans l'espace de modélisation  $\mathbb{R}^3$ . L'utilisation de coordonnées homogènes permet d'étendre la méthode par des paramètres supplémentaires comme le poids de certains points.

Pour définir les courbes, nous utilisons le modèle IFS. Nous obtenons ainsi un modèle itératif de surface, qui répond aux contraintes tout en offrant une grande richesse dans la création de formes.

Nous avons précédemment développé un outil qui permet de générer et manipuler des formes fractales [TBSS\*07] dans une interface graphique. L'utilisateur peut agir d'une part sur un ensemble de points de contrôle, à la manière des formes à pôles classiques, et d'autre part sur un ensemble de points de subdivision, permettant de moduler l'aspect lisse ou



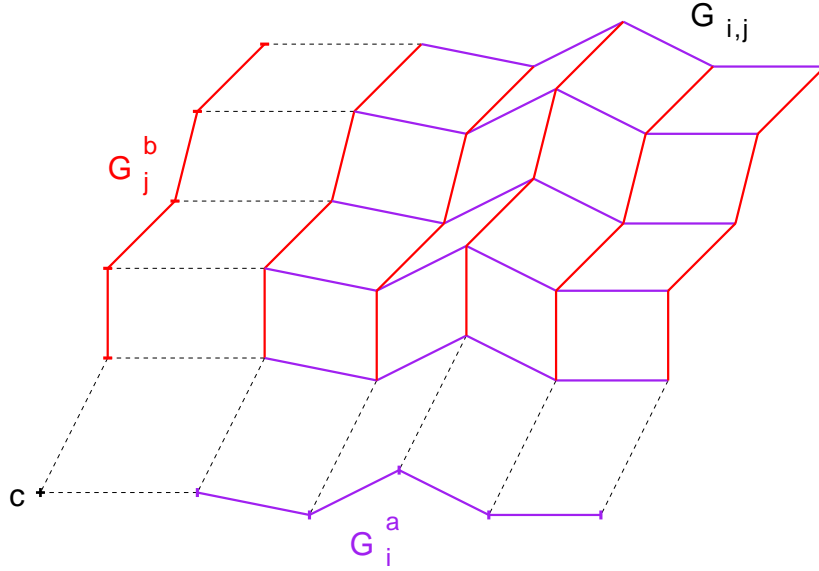


Fig. 5.2 – Construction d'un maillage par somme de lignes brisées

fractal. En nous basant sur ceci, nous mettons en œuvre un procédé permettant de saisir automatiquement des maillages quadrangulaires à faces planes.

## 5.2 Définition de surfaces par somme de courbes

Nous définissons un procédé permettant de construire une surface à partir de deux courbes quelconques. Nous décrivons dans un premier temps cet opérateur, indépendamment de la manière dont les courbes sont générées. Dans la partie 5.3, nous appliquerons cet opérateur à des courbes générées par IFS.

L'opérateur utilisé est basé sur la somme de MINKOWSKI. Nous définissons cet opérateur, non pas sur un espace affine, mais sur un espace vectoriel "homogénéisé" de dimension plus élevée. Nous verrons par la suite que cela permet d'obtenir une gamme de formes plus générale que celles obtenues par somme de MINKOWSKI dans l'espace affine initial.

### 5.2.1 Homogénéisation d'un espace affine

On notera  $(E, \vec{E})$  un espace affine à coordonnées  $(x, y, z)^T$ . On lui associe son homogénéisation  $\hat{E}$  à coordonnées  $(w, x, y, z)^T$  [Gal00]. On identifie  $E$  à l'hyperplan affine  $w = 1$  et  $\vec{E}$  à l'hyperplan linéaire  $w = 0$ .

### 5.2.2 Somme centrée

Soit  $c$  un élément de  $\hat{E}$ , on définit un opérateur binaire  $\star$  sur  $\hat{E}$  par la combinaison affine suivante :

$$\begin{aligned} \star : \hat{E} \times \hat{E} &\rightarrow \hat{E} \\ G_1 \star G_2 &= G_1 + G_2 - c \end{aligned}$$

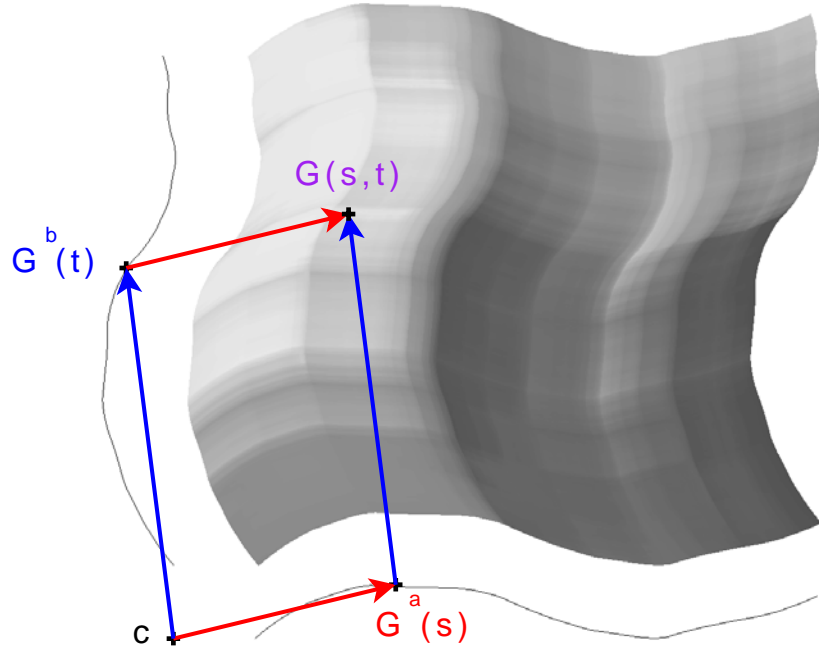


Fig. 5.3 – Construction d'une surface par somme de courbes

### 5.2.3 Interprétation en géométrie affine

On impose dans un premier temps comme restrictions aux paramètres  $G_1$ ,  $G_2$  et  $c$  d'appartenir à  $E$ , soit l'hyperplan  $w = 1$  de  $\widehat{E}$ . L'opérateur  $\star$  reste alors dans  $E$ , et peut s'interpréter comme une somme dans l'espace affine centré en  $c$ .

Cet opérateur permet de définir une surface comme la somme de MINKOWSKI de deux courbes. Dans le cas où les courbes d'entrée sont des lignes brisées, les surfaces obtenues sont des maillages.

#### Maillages (cas discret)

On considère deux listes de points  $(G_i^a)$  et  $(G_j^b)$  définis dans  $E$  ainsi que  $c \in E$ . En composant ces deux listes par l'opérateur  $\star$ , on obtient une grille de points  $(G_{i,j})$  :

$$\begin{aligned} G_{i,j} &= G_i^a \star G_j^b \\ &= G_i^a + G_j^b - c \end{aligned}$$

La notation  $\overrightarrow{AB}$  des vecteurs de  $\widehat{E}$ , avec  $A$  et  $B$  des points de  $E$ , est équivalente à la soustraction dans  $\widehat{E}$  de  $B - A$ . La définition de l'opérateur  $\star$  peut s'écrire en utilisant les opérations basiques entre points et vecteurs de l'espace affine  $(E, \widehat{E})$ .

$$\begin{aligned} G_{i,j} &= G_i^a + G_j^b - c \\ &= G_i^a + \overrightarrow{c G_j^b} \\ &= G_j^b + \overrightarrow{c G_i^a} \\ &= c + \overrightarrow{c G_i^a} + \overrightarrow{c G_j^b} \end{aligned}$$

Les listes  $(G_i^a)$  et  $(G_j^b)$  représentent des lignes brisées, dont les arêtes sont les couples

$(G_i^a, G_{i+1}^a)$  (resp.  $(G_j^b, G_{j+1}^b)$ ). Ces arêtes sont combinées pour former les faces quadrangulaires d'un maillage (voir figure 5.2).

Chacune de ces faces est un parallélogramme. En effet, les bords hauts et bas de chaque face sont tous les deux images par translation d'une arête de  $G^a$ , de même que les bords droits et gauches sont images par translation d'une arête de  $G^b$ .

En effet, de manière générale, on a :

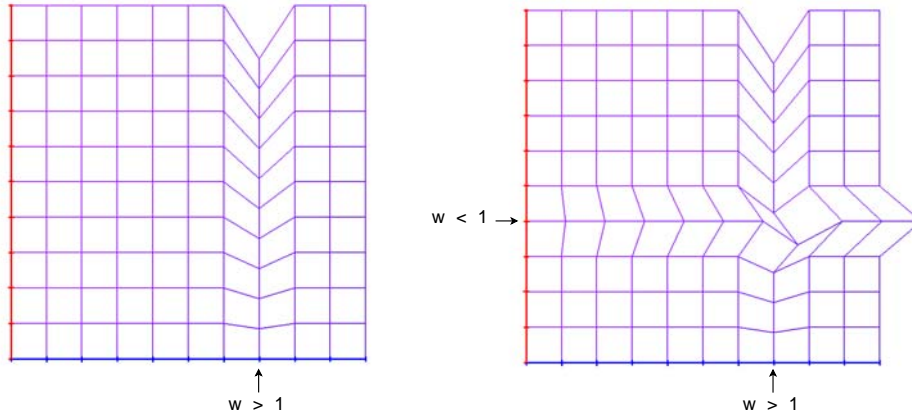
$$\begin{aligned} G_{i_2, j_2} &= G_{i_2}^a + G_{j_2}^b - c \\ &= (G_{i_1}^a + G_{j_2}^b - c) + (G_{i_2}^a + G_{j_1}^b - c) - (G_{i_1}^a + G_{j_1}^b - c) \\ &= G_{i_1, j_2} + G_{i_2, j_1} - G_{i_1, j_1} \\ &= G_{i_1, j_1} + \overrightarrow{G_{i_1, j_1} G_{i_2, j_1}} + \overrightarrow{G_{i_1, j_1} G_{i_1, j_2}} \end{aligned}$$

En particulier, cette propriété est vérifiée pour chaque face  $(G_{i, j}, G_{i+1, j}, G_{i, j+1}, G_{i+1, j+1})$  :

$$\begin{aligned} G_{i+1, j+1} &= G_{i+1, j} + G_{i, j+1} - G_{i, j} \\ &= G_{i, j} + \overrightarrow{G_{i, j} G_{i+1, j}} + \overrightarrow{G_{i, j} G_{i, j+1}} \end{aligned}$$

Les points du bord haut  $G_{i, n-1}$  sont images par translation des points du bord bas  $G_{i, 0}$ .

$$G_{i, n-1} = G_{i, 0} + \overrightarrow{G_{0, 0} G_{0, n-1}}$$



**Fig. 5.4** – Maillages surfaciques obtenus par somme de lignes brisées définies en coordonnées homogènes. Effets dus à l'augmentation ou la diminution d'un poids sur un bord

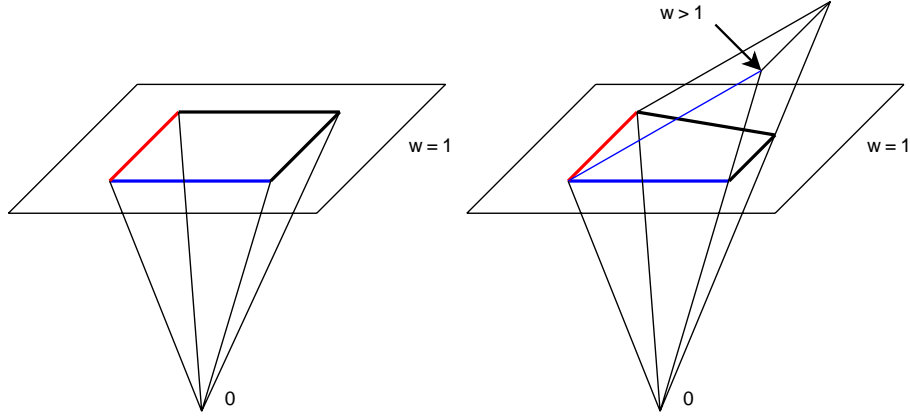
### Carreaux (cas continu)

On prend deux courbes paramétrées dans  $E$  :

$$\begin{aligned} G^a &: [0, 1] \rightarrow E \\ G^b &: [0, 1] \rightarrow E \end{aligned}$$

L'opérateur  $\star$  appliqué à  $G^a$  et  $G^b$  génère alors une surface paramétrée (voir figure 5.3) :

$$\begin{aligned} G(s, t) &= G^a(s) \star G^b(t) \\ &= G^a(s) + G^b(t) - c \end{aligned}$$



**Fig. 5.5** – Visualisation dans l'espace homogène de l'augmentation du poids d'un point

Notons que la surface ainsi générée correspond alors au balayage d'une courbe par une autre. En particulier, les bords hauts et bas de la surface sont tous les deux images par translation de la courbe  $G^a$ , de même que les bords droits et gauches sont images par translation de la courbe  $G^b$  :

$$\begin{aligned} G(s, 0) &= G^a(s) + \overrightarrow{c G^b(0)} \\ G(s, 1) &= G^a(s) + \overrightarrow{G^b(0) G^b(1)} \\ G(0, t) &= G^b(s) + \overrightarrow{c G^a(0)} \\ G(1, t) &= G^b(s) + \overrightarrow{G^a(0) G^a(1)} \end{aligned}$$

De manière analogue au cas discret, on a :

$$G(s_2, t_2) = G(s_1, t_2) + G(s_2, t_1) - G(s_1, t_1)$$

## 5.2.4 Interprétation en géométrie projective

Le procédé décrit précédemment permet de définir un ensemble de surfaces comme la somme de deux courbes. Les maillages générés ont la propriété d'être uniquement composés de parallélogrammes. Ceci constitue une limitation importante sur les maillages que l'on peut obtenir avec ce modèle.

Notre but est d'obtenir des maillages à faces planes, mais il n'est pas nécessaire que ces faces soient des parallélogrammes. L'utilisation des coordonnées homogènes présente l'intérêt suivant : l'image d'un parallélogramme par une transformation projective, est un quadrilatère plan, qui n'est pas forcément un parallélogramme.

Nous généralisons le modèle précédent, de manière à générer des maillages dont les faces ne sont plus des parallélogrammes dans l'espace affine  $(E, \vec{E})$  mais des parallélogrammes dans  $\hat{E}$ . L'utilisation des coordonnées homogènes  $(w, x, y, z)^T$  permet de généraliser la combinaison de courbes.

### Projection homographique

On définit une homographie  $\Pi$ , qui à chaque quadruplet  $(w, x, y, z)^T \in \hat{E}$  associe sa projection centrée en 0 sur l'hyperplan  $w = 1$  :

$$\Pi(w, x, y, z)^T = \left(1, \frac{x}{w}, \frac{y}{w}, \frac{z}{w}\right)^T$$

L'opérateur  $\star$  défini précédemment peut s'appliquer à deux courbes définies dans  $\widehat{E}$  et effectue leur somme dans ce même espace. La surface modélisée est obtenue par projection centrale de cette surface sur l'hyperplan  $w = 1$ .

Notons que la projection homographique  $\Pi$  ne conserve pas  $\star$ . Le procédé que nous avons défini n'est donc pas équivalent à la somme des projections centrales des deux courbes.

### Projection de la somme centrée

En posant  $q^a = \Pi(G^a)$ ,  $q^b = \Pi(G^b)$  et  $q^c = \Pi(c)$ , ainsi que  $w^a$ ,  $w^b$  et  $w^c$  les composantes  $w$  des points  $G^a$ ,  $G^b$  et  $c$ , la projection de la somme centrée  $q = \Pi(G^a \star G^b)$  vérifie :

$$\begin{aligned} q &= \Pi(G^a \star G^b) \\ &= \frac{1}{w^a + w^b - w^c} (G^a + G^b - c) \\ &= \frac{w^a}{w^a + w^b - w^c} q^a + \frac{w^b}{w^a + w^b - w^c} q^b + \frac{w^c}{w^a + w^b - w^c} q^c \end{aligned}$$

En prenant  $w^c = 1$ , et en posant  $w^a = 1 + r^a$ ,  $w^b = 1 + r^b$ , on a  $w^a + w^b - w^c = 1 + r^a + r^b$  et :

$$\begin{aligned} q &= \frac{1 + r^a}{1 + r^a + r^b} q^a + \frac{1 + r^b}{1 + r^a + r^b} q^b - \frac{1}{1 + r^a + r^b} q^c \\ &= q^a - \frac{r^b}{1 + r^a + r^b} q^a + q^b - \frac{r^a}{1 + r^a + r^b} q^b - q^c + \frac{r^a + r^b}{1 + r^a + r^b} q^c \\ &= (q^a \star q^b) - \frac{r^b}{1 + r^a + r^b} \overrightarrow{q^c q^a} - \frac{r^a}{1 + r^a + r^b} \overrightarrow{q^c q^b} \end{aligned}$$

Pour  $w^a = 1$  et  $w^b = 1$ , on a  $r^a = r^b = 0$  et il y a conservation de la somme centrée :

$$\Pi(G^a \star G^b) = q^a \star q^b$$

Pour  $w^a \neq 1$  et  $w^b = 1$ , on a  $r^a \neq 0$  et  $r^b = 0$  et la somme centrée subit une translation supplémentaire dans la direction  $\overrightarrow{q^c q^b}$  :

$$\Pi(G^a \star G^b) = (q^a \star q^b) - \frac{r^a}{1 + r^a} \overrightarrow{q^c q^b}$$

Pour  $w^a \neq 1$  et  $w^b \neq 1$ , on a  $r^a \neq 0$  et  $r^b \neq 0$  et la somme centrée subit deux translations supplémentaires dans les directions  $\overrightarrow{q^c q^a}$  et  $\overrightarrow{q^c q^b}$  :

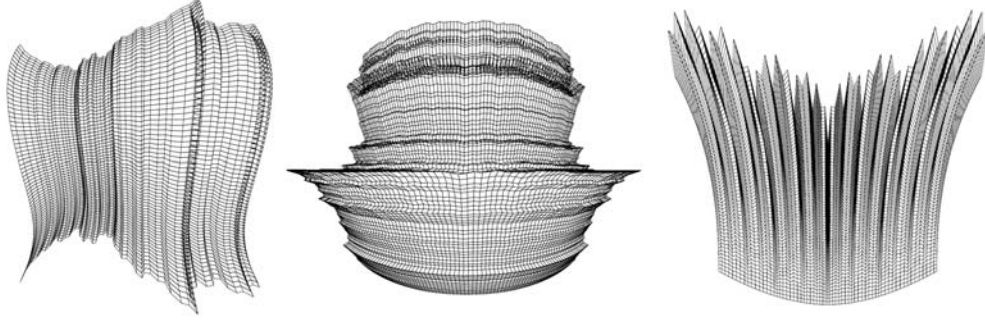
$$\Pi(G^a \star G^b) = (q^a \star q^b) - \frac{r^b}{1 + r^a + r^b} \overrightarrow{q^c q^a} - \frac{r^a}{1 + r^a + r^b} \overrightarrow{q^c q^b}$$

### Projection de parallélogrammes

Le parallélogramme  $(G_{i_1, j_1}, G_{i_2, j_1}, G_{i_1, j_2}, G_{i_2, j_2})$  est défini par quatre points de  $\widehat{E}$  tels que :

$$G_{i_2, j_2} = G_{i_2, j_1} + G_{i_1, j_2} - G_{i_1, j_1}$$

Sa projection dans  $\mathbb{R}^3$  est un quadrilatère de sommets  $(q_{i_1, j_1}, q_{i_2, j_1}, q_{i_1, j_2}, q_{i_2, j_2})$  avec  $q_{i, j} = \Pi(G_{i, j})$ .



**Fig. 5.6** – Exemples de maillages obtenus par la méthode que nous proposons

Les points projetés sont coplanaires :

$$\begin{aligned}
 q_{i_2, j_2} &= \frac{G_{i_2, j_2}}{w_{i_2, j_2}}, \\
 &= \frac{1}{w_{i_2, j_2}} (G_{i_2, j_1} + G_{i_1, j_2} - G_{i_1, j_1}), \\
 &= \frac{1}{w_{i_2, j_2}} (w_{i_2, j_1} q_{i_2, j_1} + w_{j_1, j_2} q_{i_1, j_2} - w_{i_1, j_1} q_{i_1, j_1}), \\
 &= \frac{w_{i_2, j_1}}{w_{i_2, j_2}} q_{i_2, j_1} + \frac{w_{i_1, j_2}}{w_{i_2, j_2}} q_{i_1, j_2} - \frac{w_{i_1, j_1}}{w_{i_2, j_2}} q_{i_1, j_1}
 \end{aligned}$$

$$\text{avec } \frac{w_{i_2, j_1}}{w_{i_2, j_2}} + \frac{w_{i_1, j_2}}{w_{i_2, j_2}} - \frac{w_{i_1, j_1}}{w_{i_2, j_2}} = 1 \text{ car } w_{i_2, j_2} = w_{i_2, j_1} + w_{i_1, j_2} - w_{i_1, j_1}.$$

### Projection de maillages

Tout maillage de  $\widehat{E}$  projeté  $q_{ij} = \Pi(G_{ij})$  avec  $G_{i,j} = G_i^a \star G_j^b$  est composé de facettes quadrangulaires planes  $(q_{i,j}, q_{i+1,j}, q_{i,j+1}, q_{i+1,j+1})$  car :

$$q_{i+1, j+1} = \frac{w_{i+1, j}}{w_{i+1, i+1}} q_{i+1, j} + \frac{w_{i, j+1}}{w_{i+1, j+1}} q_{i, j+1} - \frac{w_{i, j}}{w_{i+1, j+1}} q_{i, j}$$

$$\text{avec } \frac{w_{i+1, j}}{w_{i+1, i+1}} + \frac{w_{i, j+1}}{w_{i+1, j+1}} - \frac{w_{i, j}}{w_{i+1, j+1}} = 1.$$

### Influence des poids

Sur les nœuds  $(i, j)$  où  $w_i^a = 1$ ,  $w_j^b = 1$ , il y a conservation de la somme centrée :

$$q_{i, j} = q_i^a \star q_j^b$$

Si  $w_k^a = 1 + r_k^a$ , la colonne  $k$  du maillage est modifiée :

$$q_{k, j} = (q_k^a \star q_j^b) - \frac{r_k^a}{1 + r_k^a} (q_j^b - q^c)$$

Si  $w_k^a = 1 + r_k^a$  et  $w_l^b = 1 + r_l^b$ , la colonne  $k$  et la ligne  $l$  sont modifiées (Voir figures 5.4

et 5.5) :

$$\begin{aligned}
q_{k,j} &= (q_k^a \star q_j^b) - \frac{r_k^a}{1+r_k^a}(q_j^b - q^c) \\
q_{i,l} &= (q_i^a \star q_l^b) - \frac{r_l^a}{1+r_l^a}(q_i^a - q^c) \\
q_{k,l} &= (q_k^a \star q_l^b) - \frac{r_l^b}{1+r_k^a+r_l^b}(q_k^a - q^c) - \frac{r_k^a}{1+r_k^a+r_l^b}(q_l^b - q^c)
\end{aligned}$$

### Projection de carreaux

Le carreau paramétré dans  $E$  est défini par la projection de la somme centrée des arcs paramétrés :

$$q(s, t) = \Pi(G^a(s) \star G^b(t))$$

Il vérifie la propriété de planéité locale suivante. Pour tout quadruplets de nombres  $0 \leq s_1 < s_2 \leq 1$  et  $0 \leq t_1 < t_2 \leq 1$ , le quadrilatère formé par les points  $(q(s_1, t_1), q(s_2, t_1), q(s_1, t_2), q(s_2, t_2))$  est l'image par  $\Pi$  du parallélogramme  $(G(s_1, t_1), G(s_2, t_1), G(s_1, t_2), G(s_2, t_2))$

## 5.3 Modélisation itérative

La méthode de construction de surfaces par somme de courbes, vue précédemment, permet de générer automatiquement des maillages quadrangulaires à faces planes, quelles que soient les deux courbes passées en entrée. Dans cette partie, nous appliquons cette méthode, en utilisant comme paramètres des courbes définies par IFS.

### 5.3.1 Modèle itératif des courbes et des surfaces

Nous définissons les arcs paramétrés  $G^a(s)$  et  $G^b(t)$  à partir des  $\varphi^a$  et  $\varphi^b$  fonctions d'adressage associées à des IFS :

$$\begin{aligned}
G^a(s) &= T_a^{\sharp} \Phi^a(\sigma) \\
G^b(t) &= T_b^{\sharp} \Phi^b(\tau)
\end{aligned}$$

avec :

- $\Sigma_a = \{0; \dots; N_a - 1\}$ ,  $\Sigma_b = \{0; \dots; N_b - 1\}$ ;
- $(T_i^a)_{i \in \Sigma_a}$ ,  $(T_i^b)_{i \in \Sigma_b}$ , deux IFS vérifiant certaines conditions de raccord pour générer des courbes (voir 5.3.3);
- $\sigma$  développement en base  $N_a$  de  $s \in [0, 1]$ ;
- $\tau$  développement en base  $N_b$  de  $t \in [0, 1]$ ;
- $J^a = \{0; \dots; m_a - 1\}$  et  $J^b = \{0; \dots; m_b - 1\}$  indices des points de contrôle.

Nous définissons les carreaux paramétrés comme somme de ces deux arcs à pôles :

$$G(s, t) = T_a^{\sharp} \Phi^a(\sigma) \star T_b^{\sharp} \Phi^b(\tau)$$

### 5.3.2 Calcul itératif des maillages

Les lignes brisées qui approximent les arcs sont données par :

$$\begin{aligned}
p_i^{a(n)} &= G_n^a(s_i) \\
&= T_a^{\sharp} T_{\sigma_1}^a \dots T_{\sigma_n}^a \kappa_0^a
\end{aligned}$$

avec :

- $\sigma_1 \dots \sigma_n 0^\omega$  développement de  $s_i = \frac{i}{N_a^n}$  en base  $N_a$ ,
- $\kappa_0^a \in \mathbb{R}^{J^a}$ .

De même :

$$\begin{aligned} p_j^{b(n)} &= G_n^b(t_j) \\ &= T_b^{\natural} T_{\tau_1}^b \dots T_{\tau_n}^b \kappa_0^b \end{aligned}$$

avec :

- $\tau_1 \dots \tau_n 0^\omega$  développement de  $t_j = \frac{j}{N_b^n}$  en base  $N_b$ ,
- $\kappa_0^b \in \mathbb{R}^{J^b}$ .

Les maillages qui approximent les carreaux sont donnés par :

$$\begin{aligned} p_{i,j}^{(n)} &= G_n(s_i, t_j) \\ &= G_n^a(s_i) \star G_n^b(t_j) \\ &= p_i^{a(n)} \star p_j^{b(n)} \end{aligned}$$

### 5.3.3 Interprétation en géométrie affine

On prend les points de contrôle  $(p_i^a)$  et  $(p_j^b)$  dans  $E$ , c'est-à-dire tels que leurs composantes  $w_i^a$  et  $w_j^b$  soient égales à 1.

#### Somme d'arcs

Le modèle itératif affine peut s'écrire comme un carreau à pôles dont les points de contrôle sont les  $p_i^a, p_j^b$  et  $c$  :

$$\begin{aligned} G(s, t) &= T_a^{\natural} \Phi^a(\sigma) \star T_b^{\natural} \Phi^b(\tau) \\ &= \sum_i \varphi_i^a(\sigma) p_i^a + \sum_j \varphi_j^b(\tau) p_j^b - c \end{aligned}$$

avec  $\sum_i \varphi_i^a(\sigma) + \sum_j \varphi_j^b(\tau) - 1 = 1$ .

#### Produit d'arcs

On définit un opérateur  $\bullet$  permettant d'adresser un élément de la surface en fonction des adresses associées dans chacune des deux courbes :

$$\begin{aligned} k \bullet l &= m_a k + l \\ \sigma \bullet \tau &= \sigma_1 \bullet \tau_1 \dots \sigma_n \bullet \tau_n \dots \end{aligned}$$

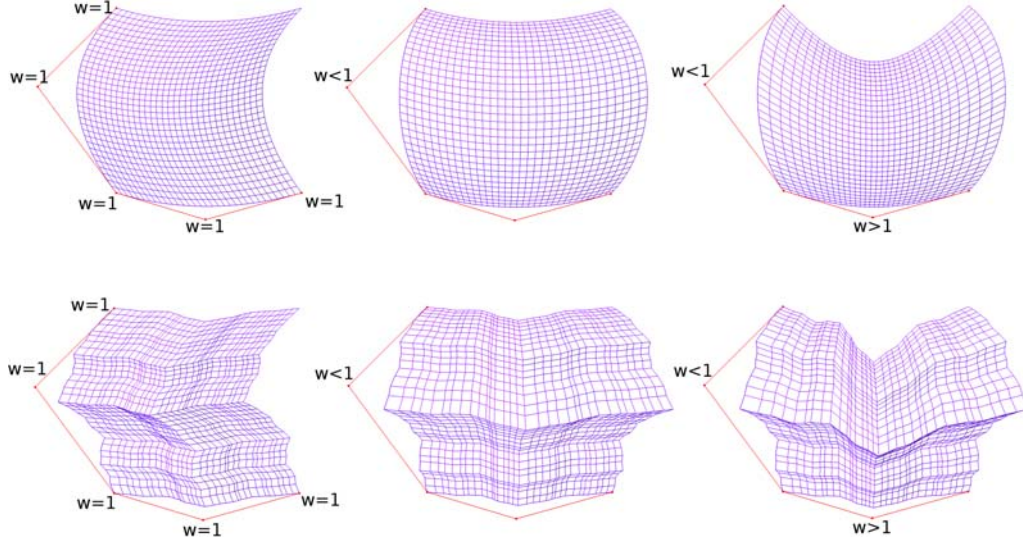
Le modèle itératif affine peut également s'écrire comme un carreau à pôles défini par un IFS :

$$\begin{aligned} G(s, t) &= (T_a^{\natural} \star T_b^{\natural}) \varphi(\sigma \bullet \tau) \\ &= \sum_{i,j} p_{i,j} \Phi_{i,j}(\sigma \bullet \tau) \end{aligned}$$

C'est un carreau à pôles avec :



- la grille de contrôle  $T_a^a \star T_b^b = (p_{i,j})_{(i,j) \in J^a \times J^b}$  obtenue par somme des polygones de contrôle des arcs :  $p_{i,j} = p_i^a \star p_j^b$  ;
- les fonctions de mélange sont obtenues par produit  $\Phi_{i,j}(\sigma \bullet \tau) = \varphi_i^a(\sigma) \varphi_j^b(\tau)$ .



**Fig. 5.7** – Maillages surfaciques obtenus par déformation homographique d'une somme de courbes générées par des IFS affines. La modification du poids des points de contrôle entraîne une variation globale de la forme en conservant ses aspects locaux. En haut, on utilise une IFS qui génère une surface fractale. En bas, les matrices de subdivision ont été modifiées pour obtenir une surface fractale.

En effet, comme  $\sum_i \varphi_i^a(\sigma) = 1$  et  $\sum_j \varphi_j^b(\tau) = 1$  nous avons la propriété suivante :

$$\begin{aligned}
\sum_{i,j} (p_i^a \star p_j^b) \varphi_i^a(\sigma) \varphi_j^b(\tau) &= \sum_{i,j} (p_i^a + p_j^b - c) \varphi_i^a(\sigma) \varphi_j^b(\tau) \\
&= \sum_{i,j} \varphi_i^a(\sigma) \varphi_j^b(\tau) p_i^a + \sum_{i,j} \varphi_i^a(\sigma) \varphi_j^b(\tau) p_j^b - \sum_{i,j} \varphi_i^a(\sigma) \varphi_j^b(\tau) c \\
&= \left( \sum_i \varphi_i^a(\sigma) p_i^a \right) \left( \sum_j \varphi_j^b(\tau) \right) \\
&\quad + \left( \sum_i \varphi_i^a(\sigma) \right) \left( \sum_j \varphi_j^b(\tau) p_j^b \right) \\
&\quad - \left( \sum_i \varphi_i^a(\sigma) \right) \left( \sum_j \varphi_j^b(\tau) \right) c \\
&= \sum_i \varphi_i^a(\sigma) p_i^a + \sum_j \varphi_j^b(\tau) p_j^b - c
\end{aligned}$$

De plus, le produit tensoriel  $\varphi(\sigma \bullet \tau) = \varphi^a(\sigma) \otimes \varphi^b(\tau)$  est une fonction d'adressage engendrée par le produit tensoriel d'IFS  $(T_i^a \otimes T_j^b)_{i,j \in \Sigma_a \bullet \Sigma_b}$  [ZT96].

### Saisie interactive des paramètres dans $E$

Les paramètres de cette famille de carreaux sont les points de contrôle des arcs de courbe :  $T_a^{\natural} = (p_j^a)$  et  $T_b^{\natural} = (p_j^b)$  ainsi que les matrices de subdivision  $T_i^a$  et  $T_i^b$  vérifiant un ensemble d'équations de raccord. Nous avons précédemment développé une méthode traitant de manière formelle ce type de contraintes [TBSS\*07]. Les paramètres indépendants du système sont saisis de manière interactive dans une interface graphique. La saisie s'effectue en déplaçant un ensemble de points de contrôle et de points de subdivision dans  $E$  (voir figure 4.51). Le déplacement de ces points entraîne la mise à jour de coefficients dans les matrices, tout en conservant les contraintes de raccord.

Chaque point de contrôle  $p_k^a$  (resp  $p_k^b$ ) correspond à la  $k$ -ième colonne de la matrice  $T_a^{\natural}$  (resp  $T_b^{\natural}$ ) :

$$p_k^a = T_a^{\natural} e_k$$

Chaque point de subdivision  $p_{i,k}^a$  (resp  $p_{i,k}^b$ ) correspond à la  $k$ -ième colonne de la matrice  $T_a^{\natural} T_i^a$  (resp  $T_b^{\natural} T_i^b$ ) :

$$p_{i,k}^a = T_a^{\natural} T_i^a e_k$$

En raison des équations de raccord, à chacun des points de subdivision peuvent être associées plusieurs colonnes dans les matrices de subdivision. En effet :

$$\begin{aligned} T_i^a e_k &= T_j^a e_l \\ \Rightarrow T_a^{\natural} T_i^a e_k &= T_a^{\natural} T_j^a e_l \end{aligned}$$

### 5.3.4 Interprétation en géométrie projective

On prend les points de contrôle  $(p_i^a)$  et  $(p_j^b)$  dans  $\widehat{E}$ , c'est-à-dire tels que leurs composantes  $w_i^a$  et  $w_j^b$  puissent être différentes de 1.

#### Projection de somme

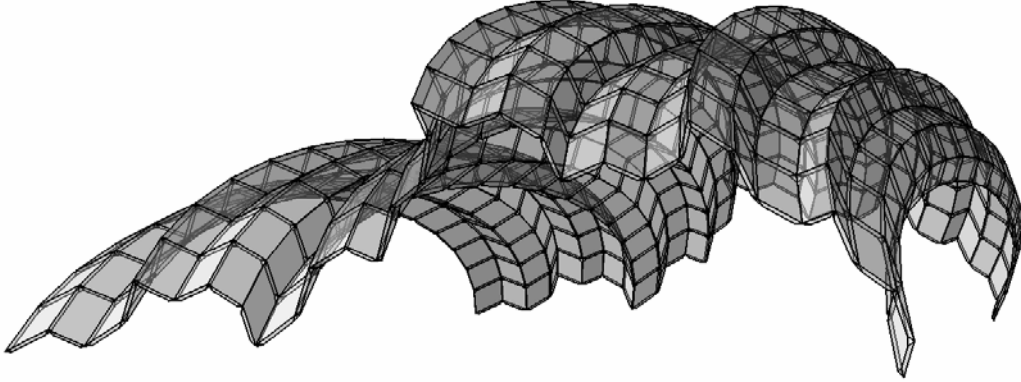
Le modèle itératif projeté de carreau s'écrit comme une combinaison affine dans  $E$  :

$$\begin{aligned} q(s, t) &= \Pi \left( T_a^{\natural} \varphi^a(\sigma) \star T_b^{\natural} \varphi^b(\tau) \right) \\ &= \Pi \left( \sum_i \varphi_i^a(\sigma) p_i^a + \sum_j \varphi_j^b(\tau) p_j^b - c \right) \\ &= \sum_i \tilde{\varphi}_i^a(\sigma, \tau) q_i^a + \sum_j \tilde{\varphi}_j^b(\sigma, \tau) q_j^b - \tilde{\varphi}^c(\sigma, \tau) p^c \end{aligned}$$

C'est un carreau à pôles dont les points de contrôle sont  $q_i^a = \Pi(p_i^a)$ ,  $q_j^b = \Pi(p_j^b)$ , et  $q^c = \Pi(c)$  et les fonctions de mélange sont les fractions :

$$\begin{aligned} \tilde{\varphi}_i^a(\sigma, \tau) &= \frac{\varphi_i^a(\sigma) w_i^a}{w(\sigma, \tau)} \\ \tilde{\varphi}_j^b(\sigma, \tau) &= \frac{\varphi_j^b(\tau) w_j^b}{w(\sigma, \tau)} \\ \tilde{\varphi}^c(\sigma, \tau) &= \frac{1}{w(\sigma, \tau)} \end{aligned}$$

avec  $w(\sigma, \tau) = \sum_i \varphi_i^a(\sigma) w_i^a + \sum_j \varphi_j^b(\tau) w_j^b - 1$ .



**Fig. 5.8** – Structure d'un maillage volumique en vue d'une réalisation en panneaux de bois. Chaque face de la figure géométrique initiale est épaissie et chanfreinée pour former un ensemble d'éléments de construction cohérent.

### Projection de produit

Le modèle itératif projeté peut également s'écrire comme un carreau à pôles défini par un IFS obtenu par produit tensoriel :

$$\begin{aligned}
 q(s, t) &= \Pi \left( (T_a^{\natural} \star T_b^{\natural}) \varphi(\sigma \bullet \tau) \right) \\
 &= \Pi \left( \sum_{i,j} p_{i,j} \Phi_{i,j}(\sigma \bullet \tau) \right) \\
 &= \Pi(T_a^{\natural} \star T_b^{\natural}) \tilde{\varphi}(\sigma \bullet \tau)
 \end{aligned}$$

C'est un carreau à pôles dont les points de contrôle sont  $q_{i,j} = \Pi(p_i^a \star p_j^b)$  et les fonctions de mélange sont les fractions :

$$\tilde{\varphi}_{i,j}(\sigma \bullet \tau) = \frac{\Phi_{i,j}(\sigma \bullet \tau) w_{i,j}(\sigma \bullet \tau)}{w(\sigma \bullet \tau)}$$

avec  $w(\sigma \bullet \tau) = \sum_{i,j} w_{i,j}(\sigma \bullet \tau)$ .

Étant une projection de somme, la grille de contrôle  $\Pi(T_a^{\natural} \star T_b^{\natural})$  est composée de faces planes.

### Saisie interactive des paramètres dans $\hat{E}$

L'interface de saisie interactive permet de déplacer les points de contrôle et les points de subdivision, et en plus de moduler leur poids.

Les points de contrôle  $p_i^a$  et  $p_j^b$  dans  $\hat{E}$  sont définis par la donnée de points de contrôle  $q_i^a$  et  $q_j^b$  dans  $E$  et celle de poids  $w_i^a$  et  $w_j^b$  car on a :

$$\begin{aligned}
 p_i^a &= w_i^a q_i^a \\
 p_j^b &= w_j^b q_j^b
 \end{aligned}$$

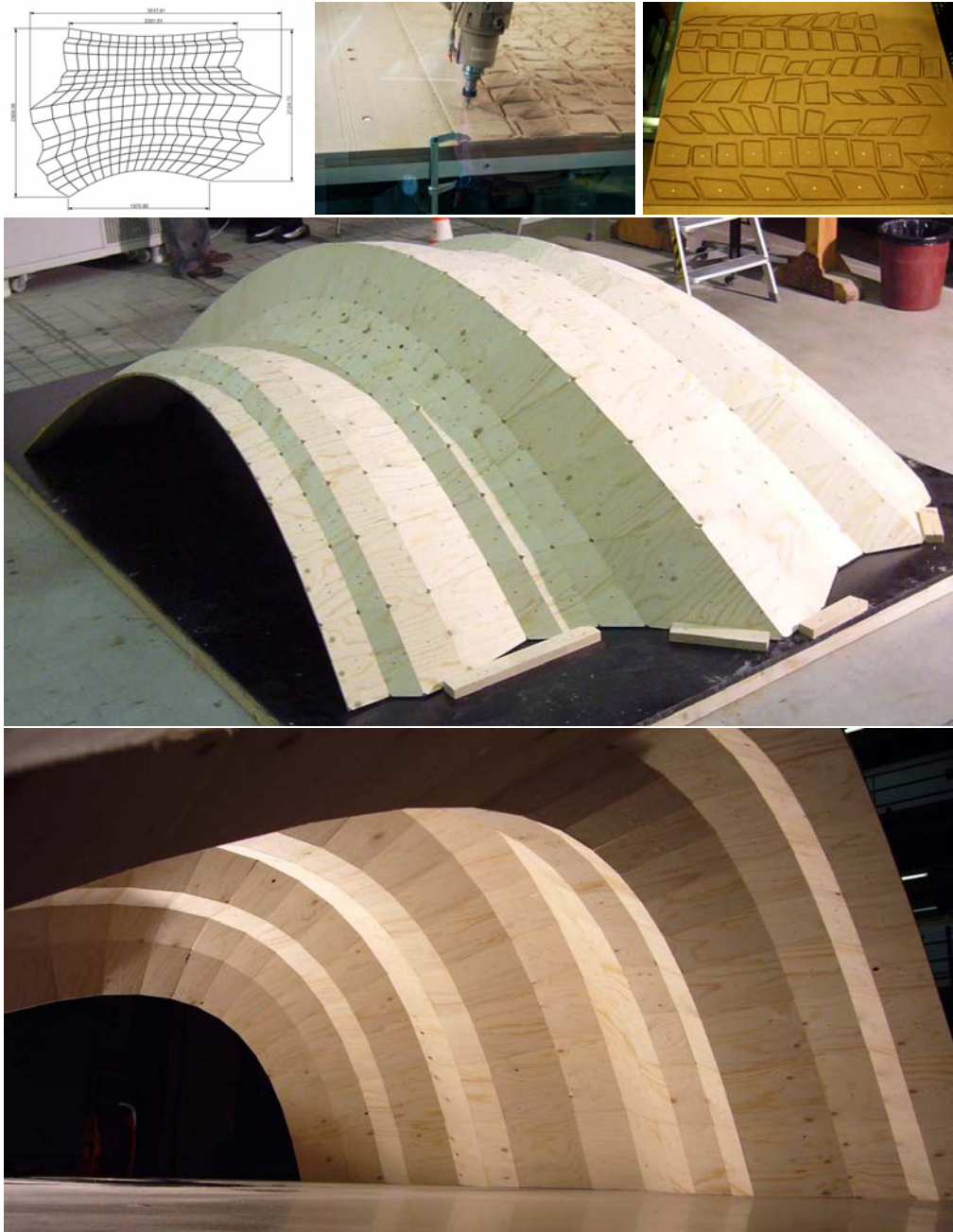
La modification du poids d'un point revient à multiplier ses coordonnées homogènes par un certain facteur. Ceci permet d'effectuer des déformations homographiques (voir figure 5.7).

## 5.4 Application à la construction

La méthode de surface proposée permet la génération d'une gamme de figures nouvelles, tout en répondant aux critères de planéité. Ainsi, elle offre aux concepteurs architectes un outil de modélisation de formes libres, permettant la création de structures a priori constructibles. L'implémentation du modèle sous forme d'un logiciel CAO permet à l'architecte la manipulation simple et intuitive des figures géométriques, à l'aide d'une interface graphique d'utilisateur. La figure 5.6 montre quelques exemples de figures que l'on peut obtenir par la méthode proposée.

Dans le cadre plus large de cette recherche interdisciplinaire, l'architecte peut alors aisément se servir de ces figures géométriques, épaissir ses faces et traiter les détails d'assemblage le long des arêtes pour obtenir des éléments de construction : des panneaux quadrilatéraux chanfreinés (voir figure 5.8). Dans le cas général, ces éléments constructifs sont tous du même type mais géométriquement différents. Cette propriété permet d'automatiser la génération de plans d'exécution, qui sont dans le cas présent des fichiers machine pour la fabrication numérique. La fonction d'adressage, introduite en 5.3.1, peut servir à l'organisation des éléments de construction au niveau logistique. L'adressage nécessaire pour optimiser les coûts de production et pour réaliser le montage de ce genre de constructions est directement proposé par le modèle CIFS.

La figure 5.9 montre la réalisation d'un prototype se basant sur la méthode de surface présentée.



**Fig. 5.9** – *Étude d'application à la construction réalisée par Ivo STOTZ : plan de fabrication des pièces, et structure en bois assemblée*



# Chapitre 6

## Conclusion

En étudiant ses propriétés mathématiques, nous avons développé une modélisation géométrique basée sur des généralisations des IFS. Nous avons montré qu'il est possible d'étendre aux formes fractales les propriétés des formes utilisées en modélisation classique.

Nous avons élaboré et étudié un nouveau formalisme, que nous avons nommé BCIFS (Boundary Controlled Iterated Function System), et qui a pu servir de base au développement du modèleur [TBSG\*06].

Ce formalisme permet d'unifier le modèle IFS et celui des schémas de subdivision ainsi que, plus généralement, une grande part des modèles itératifs déterministes tels que les L-Systèmes.

### 6.1 Topologie

Le modèle BCIFS que nous avons défini s'appuie sur le modèle CIFS qui décrit une famille de figures de manière mutuellement récursive. L'automate d'un CIFS décrit la manière dont les figures se subdivisent en d'autres figures. L'automate d'un BCIFS complète ceci en décrivant également les relations d'incidence entre les différentes figures. Chaque figure correspond alors à une certaine cellule topologique, à la manière des B-Rep.

Un BCIFS est muni d'une relation d'équivalence établissant la mise en commun entre certaines sous-cellules incidentes et subdivisées. Cette relation d'équivalence se décompose en un ensemble d'équations d'adjacence et d'incidence. Les équations d'adjacence décrivent la manière dont ces sous-cellules sont raccordées entre elles (deux arêtes partagent un même sommet, deux faces partagent la même arête, ...). Les équations d'incidence mettent en correspondance les cellules incidentes, entre un niveau de subdivision et le niveau suivant.

Chaque calcul itératif produit une suite de maillages dont les faces-arêtes-sommets correspondent aux subdivisions d'une décomposition cellulaire initiale.

Nous avons montré que ce modèle permet de décrire un ensemble très large de structures topologiques. Il permet tout aussi bien de représenter des topologies classiques que fractales. Dans le cas d'une topologie classique, la structure topologique est conservée à chaque étape de subdivision et la structure topologique initiale définit celle de la figure. La figure initiale décomposée en carreaux-arcs-singletons, voit sa géométrie raffinée par subdivision de chaque cellule. Dans le cas d'une topologie fractale, la structure topologique est modifiée à chaque étape, mais une certaine structure perdure (exemple : triangle de SIERPINSKI).

Nous avons introduit une méthode traitant des problèmes d'orientation des arêtes, et permettant d'établir une propriété de symétrie, sur un exemple simple. Cette problématique peut donner lieu à une étude plus approfondie. Nous n'avons pas abordé les problèmes

d'orientation de faces ou de sommets. Par ailleurs, dans les cas où les cellules ont une dimension interne élevée, le choix de l'opérateur de permutation peut poser problème.

## 6.2 Géométrie

Chaque relation de subdivision ou d'incidence d'un BCIFS correspond à un opérateur géométrique. Le contrôle de la géométrie passe par le contrôle des paramètres de ces opérateurs.

Notre objectif était de permettre une saisie de ces paramètres la plus intuitive possible. En choisissant d'associer un espace barycentrique à chaque état de l'automate (hormis l'état initial), nous obtenons un modèle similaire au formalisme des formes à pôles couramment utilisés en CAO, permettant une manipulation des formes par l'intermédiaire de points de contrôle. Cependant, dans les formalismes classiques, les formes sont entièrement déterminées par les points de contrôles. Dans le modèle BCIFS, les formes sont déterminées par leurs opérateurs de subdivision associés, qui ne sont pas prédéfinis. Les formes à pôles classiques sont des cas particuliers des formes générées par BCIFS. Ce sont généralement des courbes ou des surfaces polynomiales ou rationnelles par morceaux. Les courbes ou les surfaces rationnelles peuvent être obtenues par BCIFS en prenant  $E^i = P(\mathbb{R}^4)$ .

Nous avons détaillé une méthode permettant d'automatiser le traitement d'une description BCIFS pour en faire un objet manipulable graphiquement. Cette méthode prend en paramètre, en plus de la description BCIFS, la dimension interne associée à chaque état. Cette méthode permet de construire, conformément aux contraintes décrites dans le BCIFS, un ensemble de matrices de plongement associées aux relations d'incidence, ainsi qu'un ensemble de matrices formelles associées aux relations de subdivision. Ces matrices formelles représentent la forme générale que peuvent prendre les opérateurs de subdivision. Leurs coefficients sont représentés par des constantes ou des variables, dont certaines peuvent être identifiées entre elles.

Le formalisme BCIFS a été implémenté dans un logiciel dont le format d'entrée est expliqué en annexe. Dans le cadre du projet de recherche, ce logiciel a été testé et utilisé par l'architecte doctorant Ivo STOTZ pour réaliser des maquettes en bois. Il a été également testé par des élèves architectes.

La méthode de saisie interactive que nous avons développée permet de manipuler les paramètres d'un BCIFS en les représentant sous forme de points dans l'espace. La modification de ces points entraîne la mise à jour des valeurs numériques attribuées aux variables.

Les formulations BCIFS et schémas de subdivision peuvent être vues comme complémentaires dans la description et la saisie des figures : l'un s'exprime avec des matrices de subdivision et des automates, l'autre avec des grilles de points de contrôle et des masques. On distingue deux types de paramètres :

- La déformation globale obtenue par les points de contrôle,
- Suivant le choix des paramètres de subdivision, coefficients des masques (ou des matrices), la forme obtenue peut être lisse ou rugueuse.

Cependant, la méthode de saisie des points de subdivision peut être améliorée. Ces points sont définis dans un espace barycentrique  $B(\mathbb{R}^n)$  de dimension généralement supérieure à l'espace 3D dans lequel ils sont saisis, ce qui ne permet pas un contrôle total des paramètres. Lorsque l'utilisateur déplace un point de subdivision, ce point se déplace uniquement dans un sous-espace de dimension 3 de son espace barycentrique. Une perspective possible est d'élaborer un procédé permettant de contrôler la totalité des coordonnées barycentriques disponibles pour chaque point de subdivision.



## 6.3 Contraintes de fabrication

Nous avons développé une nouvelle méthode de génération de maillages à faces quadrangulaires planes pour la construction en bois [GTSW08] [SWG08] [GTSW09].

Le modèle itératif que nous proposons ne se limite pas à des surfaces développables, ni même différentiables, localement assimilables à un plan. Si les maillages obtenus possèdent bien des faces planes, les arêtes peuvent être vives.

Le concepteur peut définir des familles de maillages vérifiant la contrainte de planéité des faces, puis en moduler la forme et l'aspect avec une saisie interactive des paramètres.

Le nombre de ces paramètres est relativement limité, la surface étant entièrement déterminée par la donnée de deux courbes à pôles dans  $\widehat{E} = \mathbb{R}^4$ . Une amélioration possible de ce modèle permettrait notamment de contrôler directement les quatre courbes de bords de cette surface, ce qui permettrait une plus grande souplesse dans le contrôle de la surface.

Enfin, ce modèle permet uniquement de générer des maillages quadrangulaires. Nous avons comme perspective de mettre en œuvre un procédé plus général, permettant d'établir des contraintes de coplanarité pour des IFS surfaciques dont les maillages seraient de topologie quelconque. Ce procédé n'utiliserait donc pas d'opérateur de combinaison. Il s'agit d'intégrer les contraintes géométriques dans le modèle BCIFS, et d'établir ces contraintes sur les paramètres de subdivision d'une surface.



# Annexe

## Un langage de description de modèles itératifs

Ce chapitre présente le langage de description utilisé par le modèleur pour représenter les modèles itératifs. On utilisera tout au long de ce document l'exemple d'une courbe continue de type Bézier à 3 points de contrôle.

### Représentation d'une courbe de Bézier dans notre langage de description

```
1  /*          #
2
3          |
4          | T
5          |
6          | V
7
8    B0      B1
9    s <----- a -----> s
10
11   |         / \         |
12  Ts |       T0 /   \ T1 | Ts
13   |       / \   /   \ |
14   |      V   /   \ V   |
15   |     <----- a -----> s <----- a -----> s
16   |           B0    B1           B0    B1           */
17
18 // Etats
19
20 a.I = 1;
21 s.I = 1;
22
23 // Relations de bordure
24
25 a B0 -> s;
26 a B1 -> s;
27
28 // Relations de subdivision
29
30 # T -> a;
31
32 a T0 -> a;
```

```

33 a T1 -> a;
34
35 s Ts -> s;
36
37 // Relations d'adjacence
38
39 a [ T0 B0 = B0 Ts ];
40 a [ T0 B1 = T1 B0 ];
41 a [ T1 B1 = B1 Ts ];
42
43 // Primitives d'affichage
44
45 a [ Prim = E(B0 , B1) ];
46
47 // Grille de controle
48
49 a [ Grid = E(B0 , I , B1) ];
50
51 // Ordonnancement des espaces
52
53 a [ B0 , I , B1 ];
54
55 // Matrices de points de controle
56
57 # T = [
58 1      , 1      , 1      ,
59 -1     , 0      , 1      ,
60 0      , 1      , 0      ,
61 0      , 0      , 0      ] ;
62
63 // Matrices de subdivision
64
65 a T0 = [
66 1      , 0.5    , 0.25   ,
67 0      , 0.5    , 0.5    ,
68 0      , 0      , 0.25   ] ;
69
70 a T1 = [
71 0.25   , 0      , 0      ,
72 0.5    , 0.5    , 0      ,
73 0.25   , 0.5    , 1      ] ;

```

## Remarques préliminaires

Avant d'entrer dans les détails, voici d'abord quelques règles générales.

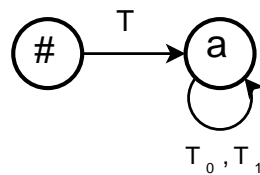
Le texte se décompose en 2 parties : le code (en noir) et les commentaires (en bleu). Les commentaires ne sont là que pour aider le lecteur à lire le code. Lorsque le programme charge un fichier de description, il ne lit que le code, et ignore les commentaires. Tout le texte compris entre les délimiteurs `\*` et `\` (retours à la ligne compris) sera pris comme du commentaire, de même que tout ce qui est précédé par `\\` (ceci jusqu'au prochain retour à la ligne).

Le code est constitué d'une suite d'instructions. Le caractère ; sert à délimiter les instructions entre elles. Un simple retour à la ligne ne suffit pas à délimiter une instruction de la suivante. Les caractères d'espace, de tabulation, et de retour à la ligne servent uniquement à délimiter les mots entre eux. Ils ont rigoureusement la même valeur dans le code. On choisit l'un ou l'autre uniquement pour des raisons de lisibilité du code.

### Description de l'automate

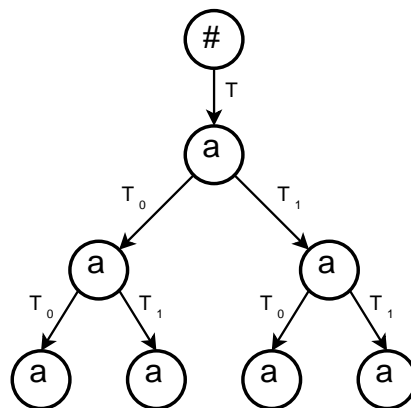
La partie principale des spécifications consiste à définir l'automate, c'est à dire les règles de subdivision entre les différents composants. L'automate permet également d'établir des contraintes topologiques. Le fichier de description pris comme exemple permet de construire une courbe de Bézier, en prenant en compte les contraintes topologiques permettant de garantir que l'on obtienne bien une courbe. Ces contraintes seront utiles par la suite, car en déplaçant les points de contrôle et les points de subdivision, on modifiera le système itératif de sorte de ne plus avoir de courbe de Bézier, mais une forme fractale qui restera quand même une courbe continue.

Rappelons d'abord la méthode permettant de définir l'automate, avec les contraintes topologiques. On définit d'abord l'automate du système sans tenir compte de la topologie (fig. 6.1).



**Fig. 6.1** – Automate sans propriétés topologiques

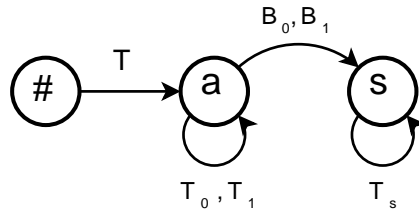
Avec cet automate, l'arbre de subdivision associé (l'arbre représentant tous les chemins possibles dans l'automate) aurait une structure totalement déconnectée, si l'on regarde niveau par niveau (fig. 6.2).



**Fig. 6.2** – Arbre de subdivision sans contraintes de raccord

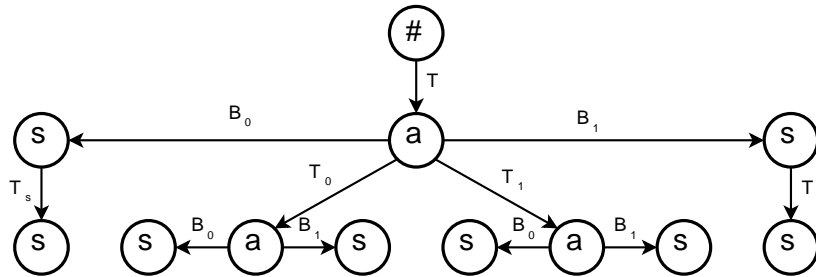
Pour établir des contraintes de raccord, on définit un nouvel état  $s$  représentant un

sommet. On définit également deux opérateurs  $B_0$  et  $B_1$ , qui ne sont pas des opérateurs de subdivision mais des relations de bordure, permettant de dire que l'arête de type  $a$  est bordée par deux sommets de type  $s$ . On définit également  $T_s$  un opérateur de subdivision associé à  $s$ . Ces états et ces transformations sont représentés dans un nouvel automate (fig. 6.3).



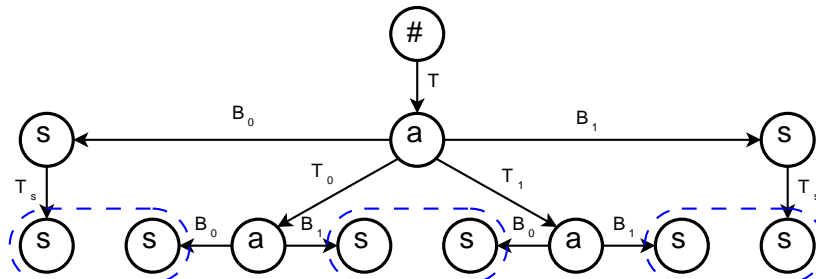
**Fig. 6.3** – Automate avec relations de bordure

Avec ce nouvel automate, l'arbre de subdivision est toujours déconnecté, mais contient plus d'informations (fig. 6.4).



**Fig. 6.4** – Arbre de subdivision avec relations de bordure, mais sans contraintes de raccord

Les informations supplémentaires disponibles sur ce nouvel arbre de subdivision vont nous permettre d'établir les contraintes topologiques. Pour obtenir une courbe bien raccordée, on établit la mise en commun de certains sommets (fig. 6.5).



**Fig. 6.5** – Mises en commun nécessaires (en bleu) pour établir les contraintes de raccord d'une courbe

Ces mises en commun se font en établissant des relations d'équivalences entre différents chemins du graphe. Par exemple, en partant du  $a$  initial, si l'on prend la subdivision  $T_0$  puis

le bord  $B_1$  on veut que le sommet obtenu soit le même que si, toujours en partant du même  $a$ , on prend d'abord la subdivision  $T_1$  puis le bord  $B_0$ . Ceci se formalise en établissant la propriété  $T_0 B_1 = T_1 B_0$ . Les autres contraintes de raccord nécessaires sont  $B_0 T_s = T_0 B_0$  et  $B_1 T_s = T_1 B_1$ . Ces trois contraintes établies, on obtient alors un arbre de subdivision qui vérifie bien les propriétés de raccord d'une courbe (fig. 6.6).

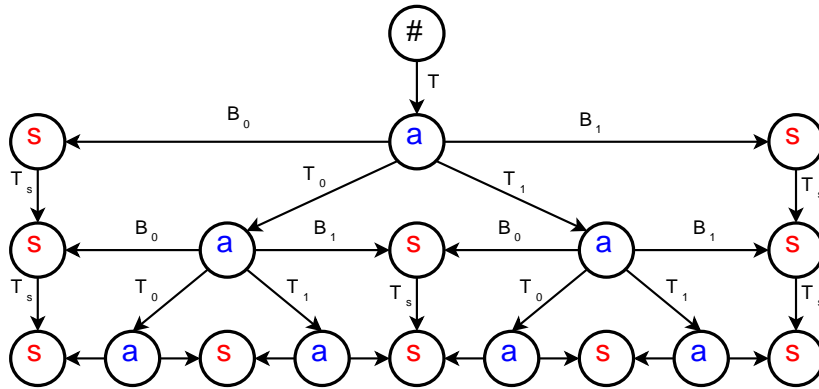


Fig. 6.6 – Arbre de subdivision avec contraintes de raccord

### Etats (lignes 20-21)

La première chose à faire est de déclarer les états présents dans l'automate. L'état # représente l'état initial du système. C'est un état qui est défini par défaut, et il ne faut pas le déclarer.

En même temps que la déclaration, on affecte une certaine valeur, qui correspond à un certain nombre de points de contrôle que l'on attribue à chaque état. Il faut faire attention au fait que l'on ne doit pas compter ici le nombre de points de contrôle faisant partie des bords. On souhaite attribuer au total trois points de contrôle à  $a$ , mais si  $a$  est bordé par deux  $s$  ayant chacun un point de contrôle, ces points de contrôle sont implicitement attribués à  $a$ , et il n'y a alors qu'un point supplémentaire à ajouter (fig. 6.7).

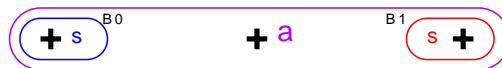


Fig. 6.7 – Points de contrôle associés à  $a$  et  $s$

### Relations de bordure (lignes 25-26)

On définit ici les relations de bordure entre les éléments. L'instruction  $a B_0 - > s$  signifie que  $a$  est bordé par  $s$  et on nomme  $B_0$  cette relation de bordure.

Les relations de bordures commencent obligatoirement par  $B$  et sont suivies d'un identificateur qui peut être une suite (éventuellement vide) de chiffres ou de lettres minuscules.

Lorsque l'on définit une relation de bordure d'un état  $x$  vers l'état  $y$ , les points de contrôle de  $y$  deviennent implicitement les points de contrôle de  $x$ . C'est pourquoi le nombre de points attribués à  $a$  n'était que de 1 car  $a$  étant bordé deux fois par  $s$ , les points de contrôle de  $s$  s'ajoutent implicitement à ceux de  $a$  (fig. 6.7).

### Relations de subdivision (lignes 30-35)

On définit les relations de subdivision. L'instruction  $a T_0 \rightarrow a$  signifie que lors du processus itératif,  $a$  génère un nouveau  $a$  par la transformation  $T_0$ .

Les relations de subdivision commencent obligatoirement par  $T$  et sont suivies d'un identificateur de la même façon que les bordures.

### Relations d'adjacence (lignes 39-41)

On définit les relations d'adjacence, c'est-à-dire les contraintes de raccord. L'instruction  $a [ T_0 B_0 = B_0 T_s ]$  signifie que si l'on part de  $a$ , le chemin  $T_0 B_0$  est équivalent au chemin  $B_0 T_s$ .

Étant donné que l'on a défini le nombre de points de contrôle associé à chaque état, on peut représenter les points de contrôle et leur mise en commun dans l'arbre de subdivision (fig. 6.8).

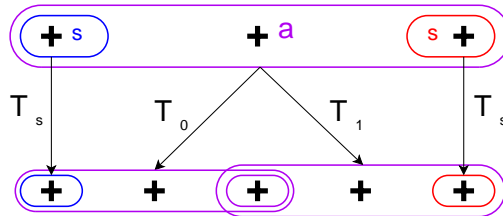


Fig. 6.8 – Arbre de subdivision avec points de contrôle

### Paramètres d'affichage

Nous avons jusqu'ici établi la description de l'automate, ainsi que les contraintes de raccord. Nous devons encore spécifier la primitive d'affichage que nous souhaitons utiliser. La primitive d'affichage définit la forme des éléments de construction que nous souhaitons modéliser. Nous pouvons également définir une grille de contrôle. Ceci n'est pas obligatoire et ne modifie en rien l'objet modélisé, mais apporte seulement une aide visuelle à l'utilisateur pour la manipulation des points de contrôle et de subdivision.

La syntaxe utilisée pour définir les primitives d'affichages est la même que pour les grilles de contrôle, et sera décrite plus en détails à la fin de ce chapitre.



### Primitive d'affichage (ligne 45)

L'instruction  $a [ Prim = E(B0, B1) ]$  signifie que la primitive d'affichage associée à  $a$  est un segment reliant les points issus des relations  $B0$  et  $B1$ .

### Grille de contrôle (ligne 49)

L'instruction  $a [ Grid = E(B0, I, B1) ]$  signifie que la primitive d'affichage associée à  $a$  est une polyligne reliant les points issus des relations  $B0$ ,  $I$  et  $B1$ .  $I$  est un opérateur défini implicitement, et qui permet de désigner le point de contrôle intérieur à  $a$ .

### Syntaxe générale

Les primitives d'affichage et les grilles de contrôle sont des polygones ou des surfaces polygonales (en utilisant  $F$  au lieu de  $E$ ) que l'on attribue à certains états de l'automate.  $E$  et  $F$  prennent un nombre quelconque de paramètres, séparés par une virgule, et qui est l'ensemble des points par lesquels passe la figure.

Chacun des sommets est désigné par le chemin permettant d'y accéder, et peut être composé d'un nombre quelconque de transformations (subdivisions, bordures, et  $I$ ), dans la mesure où le chemin est valide dans l'automate. Exemple :  $a [ Prim = F(B0, T0 T0 B1 Ts, T1 T0 T1 I, B1) ]$

Il est possible d'attribuer plusieurs figures, avec l'opérateur  $+$ . Exemple :  $a [ Prim = E(B0, I) + F(I, T1 I, B1) ]$

### Initialisation des matrices

Jusqu'ici, nous avons défini la structure du système itératif. Il reste encore à définir à quoi correspondent géométriquement les opérateurs de subdivision que l'on a déclarés formellement. Chaque opérateur peut être une combinaison de plusieurs transformations géométriques de base comme les rotations, les translations, les homothéties, ou plus généralement, les transformations affines. Chaque opérateur de subdivision est représenté par une matrice.

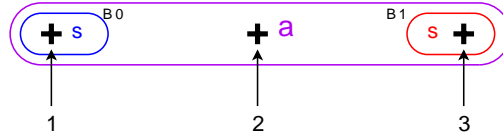
La phase d'initialisation des matrices n'est pas obligatoire. À l'exécution du programme, les matrices non définies seront initialisées avec des valeurs aléatoires.

### Ordonnement des espaces (ligne 53)

À chaque état de l'automate, on associe un ensemble de points de contrôle. À chaque transformation  $T$  allant de  $x$  vers  $y$  (déclarée par l'instruction  $x T -> y$ ) on associe une matrice à  $n_x$  lignes et  $n_y$  colonnes, où  $n_x$  est le nombre de points de contrôle de  $x$ , et  $n_y$  le nombre de points de contrôle de  $y$ . Si on souhaite affecter des valeurs à la matrice  $T$ , on doit savoir à quel point de  $x$  correspond chaque ligne de  $T$ , et à quel point de  $y$  correspond chaque colonne. On définit donc un certain ordre sur les points de contrôle de  $x$  et de  $y$ , correspondant à l'ordre des lignes et des colonnes de  $T$ . L'ordre que l'on choisit peut être complètement arbitraire, il n'a aucune conséquence sur le système itératif, mais uniquement

sur l'ordre dans lequel les lignes et les colonnes des matrices sont écrites.

L'instruction  $a [ B0 , I , B1 ]$  signifie que l'on choisit comme premier point de  $a$  le point issu de  $B0$ , puis le point intérieur  $I$ , et enfin le point issu de  $B1$  (fig. 6.9).



**Fig. 6.9** – *Ordre des points de contrôle de  $a$*

Cet ordre définit l'ordre des colonnes des matrices de transformation dont  $a$  est l'état d'arrivée (par exemple  $\# T \rightarrow a$ ), et l'ordre des lignes des matrices de transformation dont  $a$  est l'état de départ (par exemple  $a T0 \rightarrow a$ ).

### Matrice de points de contrôle (lignes 57-61)

Les matrices de points de contrôle sont les opérateurs dont  $\#$  est l'état de départ. L'état  $\#$  correspond à l'espace à quatre dimension  $(w, x, y, z)$  et les matrices de points de contrôle sont toujours des matrices à quatre lignes. Le nombre de colonnes correspond au nombre de points de contrôle dans l'état d'arrivée. Par exemple la matrice  $T$  correspondant à l'opérateur  $\# T \rightarrow a$  est une matrice à quatre lignes et trois colonnes, car  $a$  possède trois points de contrôle.

La coordonnée  $w$  correspond au poids des points de contrôle. On considère, pour simplifier, que tous les points ont un poids de 1. Chaque point de contrôle, de coordonnées  $(1, x, y, z)$  est écrit en colonne dans la matrice, et l'ordre des colonnes correspond à l'ordre des points définis précédemment.

$$\# T = \begin{bmatrix} 1 & , & 1 & , & 1 & , \\ -1 & , & 0 & , & 1 & , \\ 0 & , & 1 & , & 0 & , \\ 0 & , & 0 & , & 0 & , \end{bmatrix}$$

L'instruction précédente signifie que le premier point de contrôle de  $a$  a ses coordonnées cartésiennes  $(x, y, z)$  égales à  $(-1, 0, 0)$ , le second  $(0, 1, 0)$ , et le troisième  $(1, 0, 0)$ .

### Matrices de subdivision (lignes 65-73)

L'initialisation des matrices de subdivisions se fait de la même manière que pour les matrices de points de contrôle, à la seule différence que les colonnes ne sont pas entrées en coordonnées cartésiennes, mais sous forme de combinaisons barycentriques des points de contrôles de l'état de départ.

$$a T0 = \begin{bmatrix} 1 & , & 0.5 & , & 0.25 & , \\ 0 & , & 0.5 & , & 0.5 & , \\ 0 & , & 0 & , & 0.25 & , \end{bmatrix}$$

# Bibliographie

- [Bar88] BARNESLEY M. : ,*Fractal Everywhere*. ,Academic press, INC, 1988.
- [BD85] BARNESLEY M., DEMKO S. : ,Iterated function systems and the global construction of fractals. ,*Proceeding of the Royal Society of London Ser A399* (1985), 243–275.
- [BGN07] BENSOUANE H., GENTIL C., NEVEU M. : ,Tangentes à une courbe fractale. ,*REFIG (Revue Electronique Francophone d’Informatique Graphique*, Num. 1 (2007).
- [BGN08] BENSOUANE H., GENTIL C., NEVEU M. : ,Fractionnal half-tangent of a curve described by iterated function system. ,*Journal Of Applied Functional Analysis* (2008).
- [BJM\*88] BARNESLEY M. F., JACQUIN A., MALASSENET F., REUTER L., SLOAN A. D. : ,Harnessing chaos for image synthesis. ,*Computer graphics. Vol. 22*, Num. 4 (august 1988), 131–140.
- [Bri88] BRISSON E. : ,*Representing geometric structures in d dimensions : Topology and order*. ,Tech. Rep. 88-11-07, Dept. of Computer Science, Univ. of Washington, 1988.
- [CD92] CULIK II K., DUBE S. : ,L-systems and mutually recursive function systems. ,To appear, 1992.
- [CD93] CULIK K., DUBE S. : ,Balancing order and chaos in image generation. ,*Computer & Graphics. Vol. 17*, Num. 4 (1993), 465–486.
- [DHN85] DEMKO S., HODGES L., NAYLOR B. : ,Construction of fractal objects with iterated function systems. ,In *SIGGRAPH’85* (1985).
- [DL92] DAUBECHIES I., LAGARIAS J. C. : ,Two-scale difference equations II. local regularity, infinite products of matrices and fractals. ,*SIAM. J. Math. Anal.*. Vol. 23, Num. 4 (1992), 1031–1079.
- [Gal00] GALLIER J. : ,Embedding an affine space in a vector space. ,In *Geometric methods and applications : for computer science and engineering*. Springer Verlag, 2000, pp. 70–86.
- [Gen92] GENTIL C. : ,*Les fractales en synthèse d’image, le modèle IFS*. ,PhD thesis, Université Claude Bernard Lyon 1, mars 1992.
- [GSC\*04] GLYMPH J., SHELDER D., CECCATO C., MUSSEL J., SCHOBER H. : ,A parametric strategy for free form glass structures using quadrilateral planar facets. ,*Automation in Construction* (2004).
- [GTN06] GENTIL C., TOSAN E., NEVEU M. : ,Formes fractales mixtes. ,In *GTMG 2006, Journées du Groupe de Travail en Modélisation Géométrique, Cachan* (22-23 Mars 2006), pp. 185–210.
- [GTSW08] GOUATY G., TOSAN E., STOTZ I., WEINAND Y. : ,Un modèle itératif de surface pour la construction en bois. ,In *GTMG08* (2008).
- [GTSW09] GOUATY G., TOSAN E., STOTZ I., WEINAND Y. : ,Un modèle itératif de surfaces pour la construction en bois. ,*REFIG (Revue Electronique Francophone d’Informatique Graphique)*. Vol. 3, Num. 1 (2009).

- [GTV91] GENTIL C., TOSAN E., VANDORPE D. : ,Les fractales, un outil pour la synthèse d'images. ,*MICAD* (février 1991), 306–320.
- [Gué02] GUÉRIN E. : ,*Approximation fractale de courbes et de surfaces*. ,Thèse de doctorat, Université Claude Bernard Lyon 1, 18 décembre 2002.
- [GV90] GENTIL C., VANDORPE D. : ,Modélisation de fractales par ifs : visualisation et approximation. ,In *Journées GROPLAN Saint-Etienne* (1990).
- [HPW07] HELMUT POTTMANN S. B.-C., WALLNER J. : ,*Curves and Surfaces Design : Avignon 2006*. ,Nashboro Press, 2007, ch. Discrete Surfaces for Architectural Design methods, pp. 213–234.
- [Hut81] HUTCHINSON J. : ,Fractals and self-similarity. ,*Indiana University Journal of Mathematics*. Vol. 30 (1981), 713–747.
- [Kal83] KALAY Y. E. : ,Modelling polyhedral solids bounded by multi-curved parametric surfaces. ,*Computer-Aided Design*. Vol. 15, Num. 3 (May 1983), 141–146.
- [Lie91] LIENHARDT P. : ,Topological models for boundary representation : A comparison with n-dimensional generalized maps. ,*CAD*. Vol. 23, Num. 1 (January 1991), 59–82.
- [LM90] LE MÉHAUTÉ A. : ,*Les Géométries Fractales*, vol. Traité des Nouvelles Technologies de *série-Images*. ,Hermès, Paris, 1990.
- [LPW\*06] LIU Y., POTTMANN H., WALLNER J., YANG Y.-L., WANG W. : ,Geometric modeling with conical meshes and developable surfaces. ,*ACM Trans. Graphics*. Vol. 25, Num. 3 (2006), 681–689. ,Proc. SIGGRAPH.
- [LV88] LEVY-VEHEL J. : ,*Analyse et synthèse d'objets bi-dimensionnels par des méthodes stochastiques*. ,PhD thesis, Université Paris-sud centre d'Orsay, INRIA, 1988.
- [Man75] MANDELBROT B. : ,*Les objets fractals*. ,Flammarion, 1975.
- [Man83] MANDELBROT B. : ,*The Fractal Geometry of Nature*, 3<sup>e</sup> ed. ,Freeman, 1983.
- [Mor96] MORCRETTE M. : ,Sur l'équivalence de descriptions de figures itérées. ,*Theoretical Computer Science* (1996).
- [MP87] MICCHELLI C. A., PRAUTZSCH H. : ,Computing surfaces invariant under subdivision. ,*Computer Aided Geometric Design*, Num. 4 (1987), 321–328.
- [MP89] MICCHELLI C. A., PRAUTZSCH H. : ,Union Refinement of Curves. ,*Linear Algebra and its Applications*, Num. 114-115 (1989), 841–870.
- [PBCW07] POTTMANN H., BRELL-COKCAN S., WALLNER J. : ,Discrete surfaces for architectural design. ,In *Curves and Surface Design : Avignon 2006*, Chenin P., Lyche T., Schumaker L. L., (Eds.). Nashboro Press, 2007, pp. 213–234.
- [PH90] PRUSINKIEWICZ P., HANAN J. : ,Visualization of Botanical Structures and Processes using Parametric L-systems. ,*Scientific Visualisation and Graphic Simulation* (1990), 183–201.
- [PH91] PRUSINKIEWICZ P., HAMMEL M. : ,Automata, languages and iterated function systems. ,In *lecture notes for the SIGGRAPH'91 course : "Fractal modeling in 3D computer graphics and imagery"* (1991).
- [PH94] PRUSINKIEWICZ P., HAMMEL M. : ,Language-Restricted Iterated Function Systems, Koch Constructions, and L-systems. ,In *New Directions for Fractal Modeling in Computer Graphics, SIGGRAPH'94 Course Notes* (1994), ACM Press.
- [PLH88] PRUSINKIEWICZ P., LINDENMAYER A., HANAN J. : ,Developmental Models of Herbaceous Plants for computer imagery purposes. ,*Computer Graphics*. Vol. 22, Num. 4 (1988), 141–150.
- [PM87] PRAUTZSCH H., MICCHELLI C. A. : ,Computing curves invariant under halving. ,*Computer Aided Geometric Design*, Num. 4 (1987), 133–140.

- [PR86] PEITGEN H., RICHTER P. : ,*The Beauty of Fractals*. ,Springer-Verlag, New York Berlin Heidelberg London Paris Tokyo, 1986.
- [Pra89] PRAUTZSCH H. : ,A Round Trip to B-Splines via de Casteljau. ,*ACM Transactions on Graphics*. Vol. 8, Num. 3 (1989), 243–254.
- [Pra98] PRAUTZSCH H. : ,Smoothness of subdivision surfaces at extraordinary points. ,*Advances in Computational Mathematics*, Num. 9 (1998), 377–389.
- [Pru86] PRUSINKIEWICZ P. : ,Graphical applications of L-systems, mai 1986.
- [PW07] POTTMANN H., WALLNER J. : ,The focal geometry of circular and conical meshes. ,*Adv. Comp. Math* (2007).
- [SGW08] STOTZ I., GOUATY G., WEINAND Y. : ,IFS-Modeling for Feasible Freeform Timber Constructions. ,In *10th WCTE* (Miyazaki, Japan, 2008).
- [SLG04] SCHAEFER S., LEVIN D., GOLDMAN R. : ,Subdivision Schemes and Attractors. ,In *Eurographics Symposium on Geometry Processing* (2004), Desbrun M., Pottman H., (Eds.).
- [Smi84] SMITH A. R. : ,Plants, Fractals, and Formal Languages. ,*Computer Graphics*. Vol. 18, Num. 3 (juillet 1984).
- [SWG08] STOTZ I., WEINAND Y., GOUATY G. : ,Iterative surface design for constructions based on timber panels. ,In *SAH Statusseminar* (2008), EMPA.
- [SYC06] SOO S. C., YU K. M., CHIU W. K. : ,Modeling and fabrication of artistic products based on IFS fractal representation. ,*Computer-Aided Design*. Vol. 38, Num. 7 (2006), 755–769.
- [TBSG\*06] TOSAN E., BAILLY-SALINS I., GOUATY G., STOTZ I., BUSER P., WEINAND Y. : ,Une modélisation géométrique itérative basée sur les automates. ,In *GTMG 2006, Journées du Groupe de Travail en Modélisation Géométrique, Cachan* (22-23 Mars 2006), pp. 155–169.
- [TBSS\*07] TOSAN E., BAILLY-SALINS I., STOTZ I., GOUATY G., WEINAND Y. : ,Modélisation itérative de courbes et surfaces : aspect multirésolution. ,In *GTMG 2007, Journées du Groupe de Travail en Modélisation Géométrique, Valenciennes* (21-22 Mars 2007), pp. 53–69.
- [TGB02] TOSAN E., GUÉRIN E., BASKURT A. : ,Design and reconstruction of fractal surfaces. ,In *6th International Conference on Information Visualisation IV 2002, London, UK* (10-12 July 2002), Society I. C., (Ed.), pp. 311–316.
- [Tho96] THOLLOT J. : ,*Extension du modèle IFS pour une géométrie fractale constructive*. ,Thèse de doctorat, Université Claude Bernard Lyon 1, 9 septembre 1996.
- [Tos96a] TOSAN E. : ,*Une Algèbre de Formes fractales utilisant les IFS*. ,Master’s thesis, Université Claude Bernard Lyon 1, LIRIS, 1996.
- [Tos96b] TOSAN E. : ,*Une Algèbre de Formes fractales utilisant les IFS*. ,Thèse de doctorat, Université Claude Bernard Lyon 1, 9 septembre 1996.
- [Tos99a] TOSAN E. : ,Surfaces fractales définies par leurs bords. ,In *Journées “Courbes, surfaces et algorithmes”, Grenoble* (15-17 Septembre 1999), Briard L., Szafran N., B.Lacolle, (Eds.).
- [Tos99b] TOSAN E. : ,Wire frame Fractal Topology and IFS Morphisms. ,In *4th Conference Fractals in Engineering, Delft, The Netherlands* (June 1999), INRIA T., (Ed.).
- [TRM99] TOSAN E., RONDET-MIGNOTTE M. : ,Géométrie fractale, Informatique et Art plastique. ,In *“Créativité & Informatique”, Rencontres Musicales Pluridisciplinaires 1999* (Mars 1999), Orlarey Y., (Ed.), pp. 79–86.
- [TZTV97] THOLLOT J., ZAIR C. E., TOSAN E., VANDORPE D. : ,Modeling fractal shapes using generalizations of IFS techniques. ,In *Fractals in Engineering*, Vêhel J. L., Lutton E., Tricot C., (Eds.). Springer, 1997, pp. 65–80.

- [Wei04] WEINAND Y. : ,*New Modeling : Projeter ensemble.* ,PPUR, 2004.
- [Wei06] WEINAND Y. : ,*Géométries complexes entre l'ingénieur et l'architecte.* ,*Matières. Vol. 8* (2006), 13.
- [Wei08] WEINAND Y. : ,*Innovative Timber Construction.* ,In *10th WCTE* (2008).
- [Wei09] WEINAND Y. : ,*Innovative timber constructions.* ,*Journal of the international association for shell and spatial structures. Vol. 50, Num. 2* (2009), 111–120.
- [WWL06] WANG W., WALLNER J., LIU Y. : ,*An angle criterion for conical mesh vertices.* ,Tech. Rep. 157, Geometry Preprint Series, Vienna Univ. of Technology, April 2006.
- [Zai98] ZAIR C. E. : ,*Formes fractales à pôles basées sur une généralisation des IFS.* ,Thèse de doctorat, Université Claude Bernard Lyon 1, 11 juin 1998.
- [ZT96] ZAIR C. E., TOSAN E. : ,*Fractal modeling using free form techniques.* ,*Computer Graphics Forum. Vol. 15, Num. 3* (August 1996), 269–278. ,EUROGRAPHICS'96 Conference issue.
- [ZTA08] ZAMMOURI I., TOSAN E., AYEB B. : ,*A theoretical framework mapping Grammar based systems and Fractal description.* ,*Fractals. Vol. 16, Num. 4* (décembre 2008), 389–401.

## Curriculum vitæ

Gilles Gouaty  
Ingénieur informaticien  
Né le 28 mai 1979  
Français

### Formation

**2003 - 2004** Diplôme d'Études Approfondies DEA DISIC à l'EDIIS, Lyon

**1997 - 2003** Diplôme d'ingénieur en Informatique à l'INSA de Rennes

**1994 - 1997** Baccalauréat série Scientifique (S), mention bien

### Expérience

**Octobre 2005 - octobre 2009** Assistant doctorant sur poste d'ingénieur informatique au laboratoire de construction en bois IBOIS (EPFL), encadreurs : Yves Weinand, Eric Tosan. Sujet : Modélisation basée sur la géométrie fractale, appliquée à l'architecture et la construction en bois, prise en compte des contraintes liées à la constructibilité, géométrie algorithmique.

**Mars - juin 2004** Stage de DEA en Synthèse d'images fractales au LIRIS, axe 3, encadreur : Eric Tosan. Sujet : Visualisation rapide en 3D d'images fractales sur le modèle IFS projetés, prise en compte de la multi-résolution, subdivision dynamique, élagage d'arbres de calcul.

**Février - juillet 2003** Stage en Optimisation de code à l'IRISA, projet CAPS, encadreur : François Bodin. Sujet : Génération de code optimisé de noyaux de calculs vectoriels pour l'architecture PowerPC, compilation, optimisation, ordonnancement, allocation de registres, architecture RISC, instructions SIMD.

**Juillet - août 2002** Stage en Synthèse d'images à l'IRISA, projet SIAMES, encadreur : Kadi Bouatouch. Sujet : Modélisation 3D à partir d'images réelles, filtrage d'images afin de supprimer les ombres portées et les réflexions spéculaires tout en préservant les textures dans le but d'une reconstruction 3D.