

UNIVERSITÉ DE LIMOGES

ÉCOLE DOCTORALE Science - Technologie - Santé

FACULTÉ des Sciences et Techniques

Année : 2008

Thèse N°: 19-2008

**Thèse**

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE LIMOGES

Discipline / Spécialité : INFORMATIQUE

présentée et soutenue par

Guillaume GUIMBERTEAU

le Vendredi 04 Juillet 2008

*Modélisation volumique de bois*

Thèse dirigée par M. Dimitri PLÈMENOS, Professeur à l'Université de Limoges  
Co-dirigée par M. Olivier TERRAZ, Maître de Conférences à l'Université de Limoges

JURY:

Rapporteurs

M. Eric Galin                   Maître de Conférences HDR à l'Université Claude-Bernard Lyon 1  
M. Jean-Pierre Jessel       Professeur à l'Université de Toulouse

Examineurs

M. Georges Miaoulis       Professeur au TEI d'Athènes  
M. Dimitri Plèmenos       Professeur émérite à l'Université de Limoges  
M. Djamchid Ghazanfarpour   Professeur à l'Université de Limoges  
M. Olivier Terraz           Maître de Conférences à l'Université de Limoges







*On peut chercher tout ce qu'on veut,  
à la fin, ce qui compte c'est seulement  
ce qu'on trouve !*

Michel Michaud, *L'Amour atomique*



# Remerciements

Je tiens à exprimer toute ma reconnaissance à M. Dimitri PLEMENOS, professeur à l'Université de Limoges et directeur du laboratoire MSI pour m'avoir accueilli et encadré durant ces années de thèse. Je le remercie tout particulièrement pour m'avoir toujours soutenu et laissé libre de mes choix dans la conduite de cette thèse.

Je tiens également à remercier M. Olivier TERRAZ, maître de conférences à l'Université de Limoges, qui a co-dirigé mes recherches pendant ma thèse. Il a su m'apporter un soutien non négligeable dont j'avais besoin pour réaliser ce travail. Je lui exprime ici ma profonde gratitude et ma considération pour ses conseils sur mes orientations de recherches et pour sa grande disponibilité lors des relectures d'articles, même tard le soir...

Toute ma reconnaissance à M. Georges Miaoulis, professeur au TEI d'Athènes pour avoir accepté de faire parti de mes examinateurs. J'apprécie d'autant plus qu'il évalue mes travaux, dans la mesure où il a pu suivre l'évolution de mes recherches lors de ses déplacements réguliers à Limoges. J'espère bien pouvoir un jour lui soumettre un article pour la conférence 3IA en Grèce...

Je remercie M. Djamchid Ghazanfarpour, professeur à l'Université de Limoges, directeur du laboratoire SIR de faire parti de mon jury.

J'exprime tout ma gratitude à l'égard de M. Eric Galin, maître de Conférence HDR à l'Université Claude-Bernard Lyon 1, pour avoir accepté d'être l'un de mes rapporteurs. J'espère qu'il appréciera l'approche originale de mes travaux sur les phénomènes naturels.

Un grand merci à M. Jean-Pierre Jessel, professeur à l'Université de Toulouse, d'avoir bien voulu être l'un de mes rapporteurs. Peut-être, un jour, nous recroiserons-nous dans la «ville rose»...

Je n'oublie pas mes collègues de laboratoire, Benoît, Yahn, Cyril, et bien d'autres pour la bonne humeur qu'ils ont su apporter ; ni les étudiants que j'ai suivis dans le cadre de ma thèse et qui m'ont permis d'avoir une approche plus collaborative de la recherche.

Une pensée nostalgique au traditionnel ravitaillement du mercredi soir au restaurant vietnamien avec les voisins sans lesquels je n'aurais pas tenu ces années de dur labeur !

Un merci chaleureux à mon amie, Sandrine, pour son aide et sa patience, surtout pendant la rédaction de ce mémoire.

Je suis très reconnaissant envers la Communauté européenne et la région du Limousin qui ont financés partiellement ces recherches.

Enfin, je terminerais en remerciant tout simplement ma famille qui a su me motiver dans les moments délicats.

# Table des matières

<b>Remerciements</b>	<b>1</b>
<b>Table des matières</b>	<b>3</b>
<b>Introduction</b>	<b>7</b>
<b>1 Éléments de botanique</b>	<b>13</b>
1.1 Définitions d'un arbre et du bois . . . . .	13
1.1.1 Notion d'arbre . . . . .	13
1.1.2 Notion de bois . . . . .	15
1.2 Aspect externe et interne de l'arbre . . . . .	17
1.2.1 Aspect externe . . . . .	17
1.2.2 Structure interne d'un arbre . . . . .	21
1.3 Croissance d'un arbre . . . . .	22
1.3.1 Détails . . . . .	22
1.3.2 Structuration en axes de croissance . . . . .	24
<b>2 Techniques de modélisation</b>	<b>27</b>
2.1 État de l'art . . . . .	27
2.1.1 Objets de dimension 0 . . . . .	28
2.1.2 Objets de dimension 1 . . . . .	30
2.1.3 Objets de dimension 2 . . . . .	31
2.1.4 Objets de dimension 3 . . . . .	33
2.1.5 Logiciels . . . . .	41
2.2 Cartes généralisées et cartes généralisées augmentées . . . . .	45
2.2.1 Définition des 3Gcartes . . . . .	45
2.3 Mise en place du modèle . . . . .	47
2.3.1 Structuration du modèle . . . . .	47
2.3.2 Opérations topologiques de base . . . . .	52
2.3.2.1 Création d'un volume cylindrique . . . . .	52
2.3.2.2 Éclatement d'une arête . . . . .	53
2.3.2.3 Identification de deux faces . . . . .	53
2.3.2.4 Éclatement d'une face . . . . .	54
2.3.2.5 Éclatement d'un volume . . . . .	55

2.3.3	Modèles de plongements . . . . .	56
2.3.3.1	Plongement géométrique par défaut . . . . .	56
2.3.3.2	Plongement manuel . . . . .	58
2.3.3.3	Plongement géométrique hiérarchique pour les subdivisions . . . . .	58
2.4	Opérations topologiques de plus haut niveau . . . . .	59
2.4.1	Unité de croissance en longueur (UCL) . . . . .	60
2.4.2	Unité de branche (UB) . . . . .	63
2.4.3	Unité de jointure (UJ) . . . . .	65
2.4.4	Croissance en épaisseur (OCE) . . . . .	66
2.4.5	Algorithme général . . . . .	67
2.5	Premiers résultats . . . . .	69
2.5.1	Opération de coupe . . . . .	69
2.5.2	Résultats . . . . .	71
2.6	Conclusion . . . . .	72
<b>3</b>	<b>Formalisation du modèle de croissance</b>	<b>73</b>
3.1	Etat de l'art . . . . .	73
3.1.1	Présentation des L-systems . . . . .	74
3.1.1.1	Introduction par l'étude d'un cas pratique . . . . .	74
3.1.1.2	L-systems de dimension 1 . . . . .	76
3.1.1.3	Interprétation graphique . . . . .	77
3.1.1.4	Application aux arbres, les Tree L-systems . . . . .	79
3.1.1.5	Interprétation graphique, les problèmes de jonctions . . . . .	81
3.1.2	Présentation des Maps L-systems . . . . .	83
3.1.3	Présentation des Cellworks L-systems . . . . .	84
3.2	Introduction des 3Gmap L-systems . . . . .	85
3.2.1	Simplifications et notations . . . . .	86
3.2.2	Règles de base et opérations correspondantes . . . . .	88
3.2.2.1	Eclatement d'un volume . . . . .	88
3.2.2.2	Ajout d'un volume sur une face . . . . .	91
3.2.2.3	Identification de faces . . . . .	92
3.2.2.4	Autres opérations . . . . .	92
3.3	Interprétation des 3Gmaps L-systems . . . . .	92
3.3.1	Interprétation géométrique . . . . .	92
3.3.2	Introduction des attributs dans les L-systems . . . . .	93
3.4	Premiers résultats . . . . .	93
3.5	Application à la croissance du bois . . . . .	94
3.5.1	Tronc sans branche . . . . .	95
3.5.2	Tronc avec branche . . . . .	97
3.5.3	Relation avec les L-systems <i>traditionnels</i> . . . . .	99
3.6	Conclusion . . . . .	100

<b>4</b>	<b>Exploitation du modèle</b>	<b>103</b>
4.1	Rendu primaire . . . . .	103
4.1.1	Texturage du modèle . . . . .	103
4.1.2	Essence de bois . . . . .	106
4.2	Génération de texture . . . . .	108
4.2.1	Positionnement du modèle . . . . .	108
4.2.2	Modèle de génération de texture . . . . .	113
4.2.2.1	Introduction . . . . .	113
4.2.2.2	Blender . . . . .	114
4.2.2.3	Modèle développé . . . . .	114
4.2.3	Résultats . . . . .	114
4.3	Défauts du bois . . . . .	116
4.3.1	Duramen . . . . .	116
4.3.2	Bois de réaction . . . . .	117
4.3.3	Rayons médullaires . . . . .	117
4.3.4	Noeuds . . . . .	117
4.4	Vieillissement . . . . .	118
4.4.1	Simulation de fissures et de fractures . . . . .	118
4.4.1.1	Considérations botaniques . . . . .	119
4.4.1.2	Modèle proposé . . . . .	120
4.4.1.3	Opération de décollement progressif . . . . .	122
4.4.1.4	Illustrations . . . . .	122
4.4.2	Altérations de structure et d'aspect . . . . .	124
4.5	Conclusion . . . . .	125
	<b>Conclusion</b>	<b>127</b>
	<b>Annexe : le modeleur GGToPLSystem</b>	<b>129</b>
	<b>Bibliographie</b>	<b>138</b>
	<b>Table des figures</b>	<b>139</b>



# Introduction

## Généralités

Aujourd'hui, la forêt française couvre 29% de la superficie de notre territoire, ce qui représente 16 millions d'hectares. Elle est en constante augmentation : elle s'accroît chaque année d'environ 40.000 hectares, en surface plantée et en volume d'arbres sur pied (forêts de résineux et de feuillus). Avec une telle superficie, la France est la première puissance européenne en matière de volume de bois sur pied, cependant cette ressource est nettement sous-exploitée. En effet, la France est pratiquement la dernière en terme de consommation de bois par habitant. Pourtant, c'est bien le matériau du *XXI<sup>ème</sup>* siècle : le bois subit un effet de mode. Etant un matériau agréable et chaleureux, il est très apprécié en décoration (salons de jardin en bois, vieux meubles rénovés, . . .). La filière bois est très vaste, et les débouchés sont multiples. Nous pouvons différencier deux grandes catégories suivant la variété et la qualité du bois : le bois d'industrie et le bois d'œuvre.

Le bois d'industrie, de moins bonne qualité, est le bois destiné à l'industrie de trituration, à la fabrication des emballages (cageots, palettes, cartons) et au chauffage. L'industrie de trituration permet de déchiqeter le bois sous forme de copeaux qui servent de matière première pour la fabrication des panneaux agglomérés et de la pâte à papier.

Le bois d'œuvre, de qualité supérieure, est le bois destiné à une utilisation *noble*, c'est-à-dire pour la fabrication de meuble, de charpente, . . . La première étape de transformation la plus commune pour une grume de bois est de passer par une scierie. La grume est alors tronçonnée puis débitée pour en tirer des produits intermédiaires, le plus souvent des planches ou des chevrons à faces parallèles. Ces derniers subissent ensuite une deuxième transformation pour devenir un produit fini de forte valeur ajoutée (meubles, menuiserie, charpente, . . .). Le sciage donne beaucoup de produits connexes car une partie de la grume n'est pas utilisable pour les débits de bois. Ce sont les pertes en déchets dues à l'écorce, aux défauts, aux chutes et à la sciure. Ces déchets, qui sont de l'ordre de 40% du volume de la grume brute, ont une très faible valeur ajoutée.

Le bois d'œuvre est recherché pour ses propriétés physiques et mécaniques. En construction, le bois est utilisé pour fabriquer les charpentes, les escaliers, les portes et les fenêtres (dans une moindre mesure actuellement), et même parfois pour toute l'ossature de la maison. En effet, le bois est un matériau solide et léger. Par exemple, une

maison en ossature-bois (épicéa) pèse environ trois fois moins qu'une maison construite en matériaux durs. De plus, c'est un très bon isolant thermique et, de part ses propriétés élastiques, le bois amortit les chocs. Bien entretenu, il a une grande espérance de vie. Certaines essences ont une résistance naturelle élevée à la corrosion et l'humidité (teck). Cependant certaines irrégularités du bois peuvent altérer ses propriétés mécaniques et physiques (nœuds, bois de compression).

Le bois d'œuvre est également apprécié dans l'ameublement. Ici, ce qui fait le charme du bois, ce sont ses irrégularités. Par exemple, les nœuds, considérés comme un défaut du bois pour la construction, sont au contraire recherchés pour leur aspect esthétique, mais ces derniers doivent être répartis harmonieusement sur la pièce.

Chaque domaine d'utilisation recherche donc des caractéristiques de bois bien spécifiques en fonction de leurs besoins.

## La problématique

Actuellement, le débitage d'une grume engendre beaucoup de déchets qui sont retraités, mais avec une très faible valeur ajoutée. La valeur ajoutée de la grume pourrait être augmentée en diminuant les déchets produits et/ou en augmentant la qualité des planches (gestion des défauts du bois).

Pour optimiser la découpe de bois d'un point de vue quantitatif, il existe déjà des technologies qui se basent sur une reconstruction 3D de l'extérieur de la grume de bois. Cette dernière est effectuée à partir d'un scanner 3D. Puis un logiciel, basé sur un algorithme d'optimisation de coupe, fournit la meilleure approche à utiliser. Cependant, même si le rendement augmente, il n'en est pas forcément autant de la qualité souhaitée.

Pour pouvoir optimiser la découpe de bois d'un point de vue qualitatif, il faut être capable de reconstruire la structure interne de la grume. Pour l'instant, c'est l'opérateur chargé de la découpe qui estime visuellement la localisation des défauts sur la grume (nœuds, fissures et autres imperfections...).

La technique actuelle utilisée par l'INRA (*figure 1*) consiste à passer la grume de bois dans un tomographe à rayons X, puis de post-traiter une première fois chaque image générée par le scanner (la localisation, la forme et le type de défaut sont saisis manuellement). Puis, à partir de ces données, un logiciel reconstruit une grume de bois en 3D (mais seulement les défauts). Des recherches sont en cours pour automatiser ces traitements, les premiers résultats semblent intéressants, mais ils restent fortement liés à une espèce et un échantillon bien particulier (Longuetaud, 2005). C'est une technologie en pleine expansion, mais pour l'instant trop restrictive et trop coûteuse pour être exploitée industriellement.

Une autre approche consisterait à réaliser une prise de connaissance des défauts externes (y compris les nœuds) de la grume à l'aide de simples images (photos). Et, en réintroduisant ces données dans un simulateur de structure interne d'arbre (spécifique pour chaque espèce), il serait donc possible d'obtenir une représentation plausible de

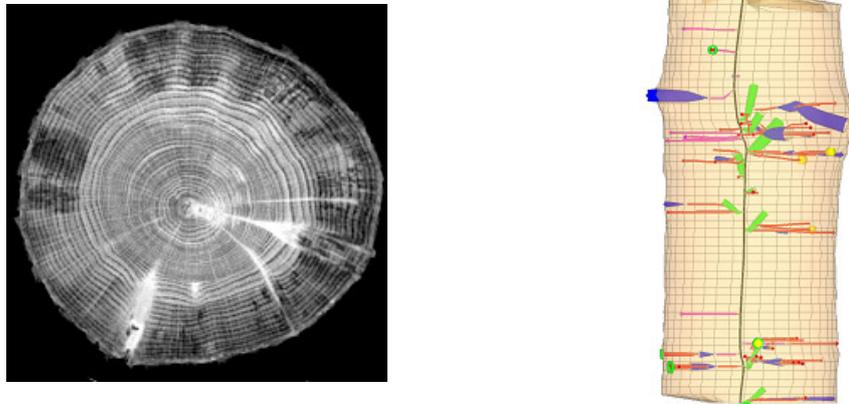


FIG. 1 – Etude de la structure interne des gourmands de chêne : image issue du tomographe à rayon X (à gauche), et reconstruction 3D à droite.

la structure interne de la grume avec une localisation cohérente des défauts. A partir de ce modèle, des simulations de coupes basées sur des aspects quantitatifs et surtout qualitatifs pourraient être effectuées, ce qui permettrait de mieux valoriser la grume.

C'est dans ce contexte, que nous proposons un modèle de génération de structure interne de bois par simulation de la croissance de l'arbre.

Notre objectif est de pouvoir générer de manière automatique des structures internes de bois plausibles d'un point de vue botanique. Pour cela, nous nous appuyons sur un modèle topologique augmenté (ce modèle est basé sur (Terraz, 1994)) pour représenter la structure interne du bois en simulant sa croissance. Notre méthode de croissance est basée sur les deux principes suivants : le respect des règles botaniques afin de réaliser un modèle proche du bois réel, et la division de la structure volumique du bois aux dimensions inférieures (tronc, branches, cernes, axes de croissance, . . .) afin de contrôler plus précisément le processus d'évolution.

## Apports des travaux

Notre modèle propose :

- une simplification double du processus de croissance :
  - grâce à l'utilisation de la théorie des cartes qui permet de décrire des métamorphoses de subdivisions de volumes (à chaque étape, les volumes représentant le cerne généré à l'étape précédente sont éclatés afin de générer le nouveau cerne) et des outils de contrôle de l'évolution à différents niveaux (volumes, cernes, axes de croissance) ;
  - grâce à l'introduction d'un formalisme basé sur les L-systems, les 3Gmaps L-systems, qui permet de simplifier l'expression de la croissance et de pouvoir réutiliser des L-systems existants ;
- l'insertion appropriée d'une nouvelle branche avec une répartition spatiale correcte des cernes autour des nœuds (botaniquement un nœud est l'origine d'une branche).

Les connexions entre les éléments du tronc et ceux des branches sont exactement manipulés, il en résulte une distribution spatiale plausible des cernes autour des nœuds ;

- la modélisation de phénomènes particuliers, comme le bois de cœur (duramen) et le bois de réaction : ces phénomènes sont contrôlés par des attributs géométriques spécifiques aux volumes comme la hauteur, l'épaisseur... ;
- la génération de textures de bois plus réalistes (gestion spatiale cohérente des nœuds et des phénomènes particuliers décrits précédemment) ;
- la simulation de certains phénomènes liés au vieillissement de cette structure de bois appliquée sur des objets.

Même si notre but principal est de produire un bois botaniquement correct en infographie, beaucoup de domaines d'applications peuvent être concernés par notre méthode :

- la botanique, avec l'introduction dans notre modèle, de paramètres de croissance spécifiques (atmosphère, environnement) et la sélection de bons spécimens, comme dans (Szafran *et al.*, Octobre 2003) ;
- la science des matériaux, avec l'introduction de modèles de vieillissement ou de propriétés mécaniques dans notre structure ;
- l'industrie du bois, avec une optimisation de la découpe de bois d'un point de vue qualitatif (localisation des nœuds et autres défauts) et un outil de reconstruction de la grume de bois.

Tous ces champs d'application exigent un modèle 3D interne plausible de bois.

## Organisation du mémoire

Dans ce mémoire, nous allons donc présenter les modèles que nous avons développés afin de proposer une approche originale pour générer des structures internes de bois ainsi que des textures.

Tout d'abord, nous commencerons par introduire les notions de botanique nécessaires pour une meilleure compréhension de la croissance de l'arbre. Nous définirons donc les notions d'arbre et de bois, et les différentes parties constituant celui-ci. Puis nous détaillerons les différentes étapes de la croissance.

Ensuite, dans un second chapitre, nous présenterons la théorie et le premier modèle que nous avons développé. Nous commencerons par nous positionner par rapport aux travaux déjà existants dans ce domaine. Et, après avoir présenté quelques définitions sur la théorie des cartes généralisées de dimension 3 (ou 3Gcartes), nous détaillerons les opérations topologiques de bas et haut niveau qui sont nécessaires pour mettre en place ce modèle. Nous finirons par quelques exemples de construction de structure de bois.

Ce premier modèle étant complexe à mettre en œuvre, nous proposons donc dans le chapitre 3, d'introduire un nouveau formalisme basé sur les L-systems : les 3Gmaps L-systems qui simplifie l'expression des opérations de croissance, et permet de réutiliser les L-systems déjà existants. Nous finirons par illustrer cette théorie sur quelques exemples.

Le dernier chapitre traitera de l'exploitation du modèle développé dans les chapitres

précédents. Après avoir introduit le mode d'application de texture utilisé sur tous les volumes du modèle, nous présenterons un outil qui permet d'en déduire une texture volumique de bois. Nous l'illustrerons par des exemples qui permettent de mettre en avant la cohérence visuelle du modèle. Nous montrerons ensuite que le modèle est capable de générer assez facilement des défauts de bois comme le bois de réaction. Enfin, nous finirons par une application au vieillissement du bois (simulation de fissures, altération des couleurs).



# Chapitre 1

## Éléments de botanique

Afin d'accroître le réalisme de notre méthode, nous avons choisi de simuler la croissance du bois pour générer nos structures internes de bois. Naturellement, nous avons choisi de développer notre modèle en accord avec les règles de croissances botaniques.

Dans ce chapitre, nous définirons donc les notions de botanique qui nous serviront à modéliser la croissance du bois (quelques autres éléments de botanique seront introduits dans le chapitre 4).

Après avoir défini les notions d'arbre et de bois, nous présenterons de manière plus approfondie les différentes parties qui constituent l'arbre (partie externe) et le bois (partie interne). Puis, nous détaillerons les différents phénomènes qui peuvent intervenir lors de la croissance de l'arbre. Enfin, nous montrerons comment les botanistes décomposent la croissance de l'arbre en axes de croissances.

### 1.1 Définitions d'un arbre et du bois

Dans un premier temps, nous définirons la notion d'arbre et les deux grandes familles qui les constituent, puis nous ferons de même pour le bois.

#### 1.1.1 Notion d'arbre

Pour caractériser formellement la notion d'arbre, nous allons nous appuyer sur quelques définitions trouvées dans des dictionnaires.

**ARBRE** *n.m. (lat. arbor) :*

Grande plante ligneuse vivace dont la tige, fixée au sol par des racines, n'est chargée de branches et de feuilles qu'à partir d'une certaine hauteur.

*Petit Larousse illustré 1996*

**ARBRE** *n. m.* :

Végétal ligneux de grande taille (6 ou 7 m au minimum), dont la tige (tronc), simple à la base, ne se ramifie qu'à partir d'une certaine hauteur.

HACHETTE : le dictionnaire couleur 1991

Avant de poursuivre, il convient donc de définir une plante ligneuse. Une plante ligneuse est une plante qui fabrique de la lignine, molécule qui donne à la plante sa solidité.

Parmi les végétaux ligneux, nous pouvons distinguer trois grandes catégories (*figure 1.1*) :

- **les arbres** : un arbre se distingue par l'existence d'un tronc, tige unique qui se ramifie en branches maîtresses au-delà d'une certaine hauteur (chêne)<sup>1</sup> ;
- **les arbustes** : un arbuste est un arbre de plus petites dimensions (entre 4 et 7 mètres) mais dont le tronc est bien différencié (néffier) ;
- **les arbrisseaux** : un arbrisseau est un arbuste de moins de 4 mètres, ramifié dès le niveau du sol (forcicia, laurier).



FIG. 1.1 – De gauche à droite : arbre (chêne), arbuste (néffier), arbrisseau (laurier)

L'arbre peut donc être défini de manière formelle comme une plante ligneuse de grande taille (plus de 7 mètres), constituée d'une tige unique (le tronc) qui se ramifie en branches au delà d'une certaine hauteur. La grande taille de cette plante est due en partie à la grande vascularisation de ses organes aériens et souterrains (cellules de grandes dimensions).

L'arbre est un organisme végétal. Les premiers arbres (ancêtres des conifères que nous connaissons aujourd'hui) sont apparus sur Terre il y a environ 300 millions d'années (Dévonien). Au cours de leur évolution, deux grands groupes d'arbres se sont distingués :

- **les conifères** : ce sont les plus anciens. Pour la plupart, ils gardent leur feuillage en hiver. Ce feuillage est constitué d'aiguilles (feuilles repliées) ou d'écaillés. Commu-

<sup>1</sup>Le palmier est une exception car il ne possède pas de branches (il est constitué d'un tronc surmonté de feuilles).

nément ces arbres sont appelés *résineux*. Cette appellation provient de la constitution plus épaisse de leur sève qui colle aux doigts : la résine. Ils appartiennent au groupe des gymnospermes (ovule à nu, graine non enfermée dans un fruit ou plante à graine nue). Leur fruit est en forme de cône (ex : la pomme de pin). C'est un bois qui est surtout utilisé dans la fabrication de charpente et de pâte à papier ;

- **les feuillus** : ils sont apparus plus récemment. Ils sont habillés de feuilles larges et plates qui naissent au printemps et tombent à l'automne. Ils appartiennent au groupe des angiospermes (ovule enfermé, reproduction par pollinisation des fleurs grâce aux insectes, graine protégée par une coque). Le fruit peut être une baie, une drupe, un akène... Leur famille se décompose en deux classes :
  - *les monocotylédones* (plantule à un cotylédon<sup>2</sup>) : ce sont les arbres dont l'unique cotylédon évolue en donnant une préfeuille (palmier). La tige ne connaît pas de formation de bois secondaire (cette notion sera définie dans la *section 1.3.1*), et les feuilles présentent des nervures principales souvent parallèles. Il en résulte une structure interne très fibreuse et peu résistante. Ces derniers sont donc très peu utilisés dans l'industrie, leur emploi restant limité essentiellement pour la confection de sièges et la décoration ;
  - *les dicotylédones* (à deux cotylédons) ou végétaux dits *supérieurs* rassemblent les arbres à feuilles larges et à nervures ramifiées. La tige connaît une formation de bois secondaire. Cette sous-famille compte plusieurs milliers d'espèces dont les bois sont très variés (chêne, if, ...). De part leurs propriétés physiques remarquables, ils sont très utilisés dans l'industrie (ameublement, pièces d'ouvrages, ...).

Dans les exemples que nous illustrerons par la suite, nous ne nous intéresserons pas aux feuillus monocotylédones. Aussi, et par abus de langage, nous nommerons simplement *feuillus*, les *feuillus dicotylédones*.

Après avoir défini rigoureusement la notion d'arbre, nous allons présenter la structure interne de l'arbre : *le bois*.

### 1.1.2 Notion de bois

Comme précédemment, nous allons nous appuyer sur quelques définitions trouvées dans des dictionnaires pour introduire la notion de bois.

**BOIS** *n. m.* (*bas lat. boscus, germ. bosk*) :

Matière compacte et ligneuse, plus ou moins dure, composant le tronc, les racines et les branches des arbres, recouverte de l'écorce et qui transporte la sève.

*Petit Larousse illustré 1996*

<sup>2</sup>Les cotylédons sont les feuilles primordiales constitutives de la graine.

**BOIS** *n.m.* :

Substance solide et fibreuse qui compose les racines, la tige, et les branches des arbres.

HACHETTE LE DICTIONNAIRE COULEUR 1991

Intuitivement, le bois peut être défini comme « *l'intérieur de l'arbre* », c'est à dire la matière enfermée par l'écorce.

D'une manière plus rigoureuse, la norme **NF B 50-003** définit le bois comme « *un ensemble de tissus résistants secondaires (de soutien, de conduction, et de mise en réserve) qui forment les troncs, branches et racines des plantes ligneuses. Issu du fonctionnement du cambium périphérique, il est situé entre celui-ci et la moelle* ».

C'est un matériau très apprécié pour ses propriétés mécaniques et pour son pouvoir calorifique. C'est également une matière première importante pour l'industrie chimique. Nous reviendrons plus en détails sur les propriétés mécaniques dans la *section 4.4*.

Le bois (ou xylène) est composé de différents types de tissus végétaux (*figure 1.2*)<sup>(3)</sup> :

- **les fibres** : ce sont des faisceaux de cellules résistantes qui sont disposés dans le sens axial et qui assurent la rigidité et la résistance mécanique du bois. Il s'agit d'un bio-composite constitué de cellulose, d'hémicellulose et de lignine ;
- **le tissu vasculaire** : il est formé de vaisseaux, cellules creuses qui servent à acheminer la sève brute des racines jusqu'aux feuilles. Ces vaisseaux sont souvent associés à des cellules parenchymateuses (aussi appelés parenchyme) qui contribuent au transport des nutriments. Ces parenchymes, associés aux vaisseaux, donnent des motifs particuliers à chaque essence ;
- **les rayons ligneux** : ce sont les cellules de réserve (parenchymateuses) à parois épaissies et lignifiées, qui accompagnent le tissu vasculaire. Ces cellules participent en outre à la fonction de soutien. Leur orientation est transversale et rayonnante en partant de l'axe longitudinal de l'arbre. Ils sont invisibles à l'œil nu chez les résineux, mais parfois visibles chez les feuillus comme le chêne.

La disposition des tissus, la forme, la taille et la couleur des cellules est appelée plan ou rayon ligneux. Celui-ci est caractéristique de chaque essence. Les cellules et les fibres du bois sont orientées généralement dans le sens axial de la croissance de l'arbre, définissant ainsi le « fil du bois ».

Nous verrons également dans la *section 4.4* que cette disposition peut aussi être altérée par les conditions de vie de l'arbre.

Le bois peut donc prendre plusieurs aspects qui dépendent de l'espèce et des conditions de croissance de l'arbre. Ainsi le pin a une couleur plus claire que le noyer (*figure 1.3*) .

Le modèle que nous proposons, nous permet de construire la structure interne de

<sup>3</sup>Images extraite du site <http://www.snv.jussieu.fr/vie/dossiers/chene/chene1.htm>



FIG. 1.2 – Coupe macroscopique du chêne (de gauche à droite) : chêne sec, chêne mouillé, et coupe zoomée



FIG. 1.3 – Divers essences de bois : noyer, merisier, chêne, pin

l'arbre : le bois.

Bien que nous détaillerons complètement l'aspect externe d'un arbre, nous ne nous attarderons que sur la partie dite exploitable du bois (c'est à dire la grume ou le tronc). Cependant, nous pourrions remarquer que notre modèle permet de générer aisément les parties (bouts de branches, racines) que nous avons omises volontairement.

Avant de détailler les processus qui interviennent lors de la croissance de l'arbre, nous allons détailler les différentes parties qui le constituent.

## 1.2 Aspect externe et interne de l'arbre

Dans un premier temps, nous commencerons par détailler l'aspect externe de l'arbre, puis nous nous pencherons sur la structure interne de celui-ci.

### 1.2.1 Aspect externe

#### Les différentes parties de l'arbre

Traditionnellement, un arbre est décomposé en plusieurs parties (*figure 1.4*) <sup>(4)</sup> :

- **le houppier ou cime** : c'est la partie supérieure de l'arbre (au dessus du tronc). Elle est constituée d'un enchevêtrement de branches dites maîtresses, lorsqu'elles constituent l'ossature de la cime de l'arbre (lorsqu'elles partent du tronc), ou

<sup>4</sup>Image extraite du site <http://alain.granier2.free.fr>

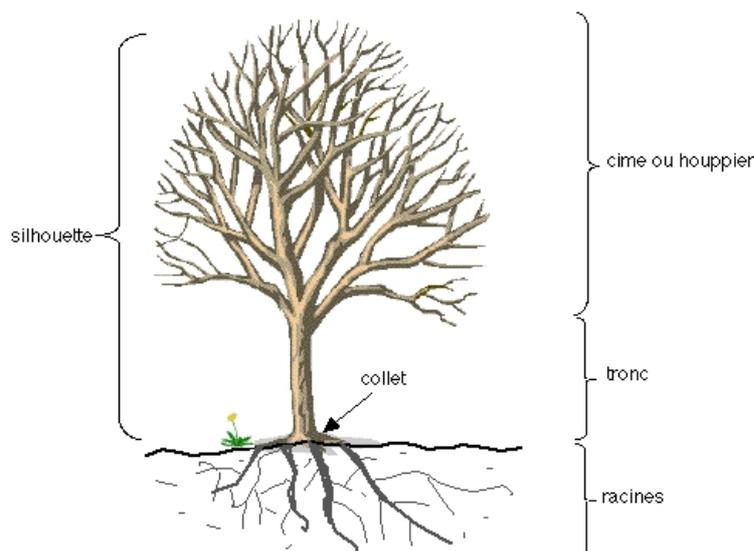


FIG. 1.4 – Les différentes parties d'un arbre

secondaires, si elles supportent les rameaux. Sur les rameaux se trouvent suivant la saison des bourgeons, des fleurs, des feuilles (ou des aiguilles) et des fruits ;

- **le tronc, ou tige** : c'est la partie cylindrique située entre le sol et la naissance des branches principales. Chaque branche est constituée comme un petit tronc. La tige se compose d'une suite de nœuds et d'entre-nœuds. Elle possède généralement une forme cylindro-conique qui lui assure une symétrie plus ou moins parfaite par rapport à son axe. Le sommet de la tige est occupé par un bourgeon, dit bourgeon terminal. Le tronc est constitué de deux parties : le bois, tissu ligneux au centre, et l'écorce en périphérie. La partie exploitable du bois est le tronc<sup>5</sup> ;
- **le système racinaire (les racines)** montre une certaine ressemblance par sa disposition avec les branches. En effet, les racines principales qui partent du tronc, se ramifient en s'amincissant jusqu'à devenir des radicelles (racines très fines également appelées chevelus) garnies de poils permettant de capter l'eau et les sels minéraux ;
- **le collet** constitue le point où la partie aérienne rejoint la partie souterraine de l'arbre (on l'appelle souche quand l'arbre est abattu).

Pour être complet dans notre modèle, il nous faut maintenant définir la notion de *bourgeon* (nous reviendrons sur son rôle dans la *sous-section 1.3.1*).

En botanique, un bourgeon désigne une excroissance apparaissant sur certaines parties des végétaux et donnant naissance aux branches (bourgeon à bois), aux feuilles (bourgeon à feuilles), aux fleurs (bourgeon à fleurs). Dans notre cas, nous ne nous in-

<sup>5</sup>Le tronc abattu et non équarri est appelé aussi grume.

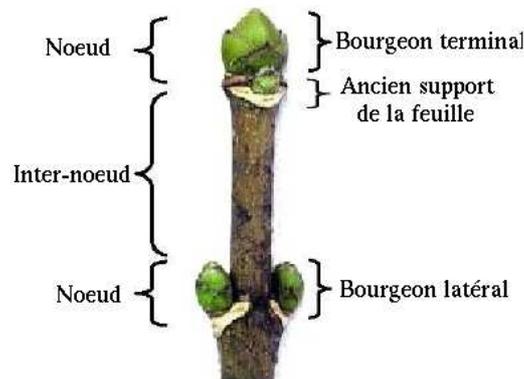


FIG. 1.5 – Nœud, internœud et bourgeons

téresserons qu'aux bourgeons à bois (*figure 1.5*) . Un bourgeon (à bois) est constitué d'une agglomération de jeunes feuilles pressées les unes contre les autres. Il assure la croissance et la ramification des tiges. On distingue, suivant leur position sur la tige, les bourgeons terminaux (situés à l'extrémité de la tige), les bourgeons latéraux ou axillaires (qui naissent à l'aisselle des feuilles) et les bourgeons adventifs (qui naissent sur un organe ou à une position inhabituels, comme par exemple entre deux nœuds ou des racines). L'éclosion du bourgeon au printemps est nommée *débourrement*. Le bourgeon se forme pendant l'été, principalement à l'endroit où la feuille est attachée au rameau ou à la branche. Ce bourgeon reste en place lorsque la feuille tombe à l'automne et, le printemps suivant, touché par la chaleur du soleil, il éclot.

### Silhouette d'un arbre

Le port d'une plante (ou allure) est surtout influencé par la manière dont la tige principale et les tiges secondaires ou branches se comportent les unes par rapport aux autres. Lorsque la tige principale est beaucoup plus forte que les tiges secondaires, on a la forme ordinaire de la plupart des arbres dont le tronc est couronné d'un grand nombre de branches très ramifiées. Si, au contraire, la tige principale ne s'accroît pas plus que ses ramifications, la plante prend l'aspect de buisson caractéristique des arbustes ou des arbrisseaux. Certaines tiges ne se ramifient pas du tout, c'est le cas pour les Palmiers dont le tronc en colonne est surmonté d'un énorme bouquet de feuilles. Notre étude se portera plus particulièrement sur les variétés d'arbres dites courantes (les feuillus, voir définition *sous-section 1.1.1* ).

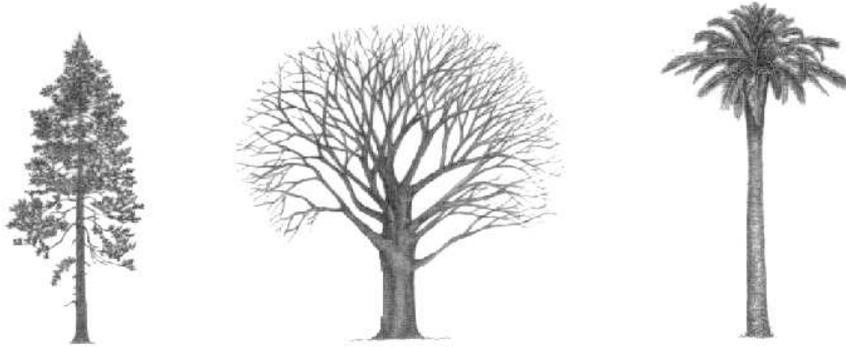


FIG. 1.6 – Les différents aspects du tronc pour des arbres résineux, feuillus et monocotylédones résineux

La silhouette des feuillus (arbres portant des feuilles) correspond souvent à une boule ou à un œuf, alors que celle des résineux est généralement en forme de cône. Il faut noter cependant que ces formes ne sont visibles que pour des arbres isolés, non taillés et qui se développent sans contrainte (*figure 1.6*) .

La silhouette d'un arbre (forme générale) est donc souvent caractéristique et constante. Elle dépend de son espèce, mais également de ses conditions de vie. Plus ses conditions de vie seront altérées plus l'arbre s'éloignera de sa forme générale (*figure 1.7*) .



FIG. 1.7 – Pin parasol déformé par le vent

Nous traiterons plus en profondeur des défauts de silhouette et de structure (interne) qui peuvent découler des altérations du biotope dans la *section 4.3* .

### 1.2.2 Structure interne d'un arbre

Après avoir vu l'aspect externe d'un arbre, nous allons maintenant détailler son aspect interne.

Si l'on effectue une section transversale d'un arbre à mi-hauteur du tronc, nous pouvons retrouver, en partant de l'extérieur vers l'intérieur, les sections suivantes (*figure 1.8*) :

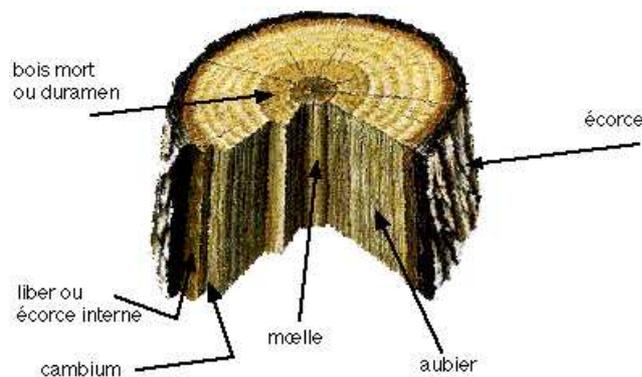


FIG. 1.8 – Coupe transversale

- **l'écorce** : c'est le revêtement extérieur du tronc, des branches et des racines de l'arbre. Il est souvent possible de reconnaître un arbre par son écorce. L'écorce protège l'arbre des intempéries, des insectes... Elle est plus ou moins épaisse selon les espèces. En botanique, l'écorce comprend également l'ensemble des tissus produits vers l'extérieur de la tige par le fonctionnement du cambium. C'est une couche de cellules génératrices, qui produit vers l'intérieur le xylène, ou bois, et vers l'extérieur le phloème ou liber ;
- **l'écorce interne ou liber** : le liber (écorce interne) sert à transporter la sève élaborée. La plupart des cellules qui le composent sont mortes. La partie la plus interne du liber possède des canaux qui amènent la sève élaborée des feuilles aux cellules du cambium ;
- **le cambium** : c'est une partie très peu visible à l'œil nu, mais elle assure la croissance de l'arbre en épaisseur. Cette couche, plus facilement visible au microscope, est constituée de cellules vivantes. Ce méristème secondaire produit le bois d'aubier vers l'intérieur et le liber vers l'extérieur de tronc ;
- **le bois vivant ou aubier** : il est composé de cellules plus ou moins vivantes en voie de duraminisation. Ces cellules transportent la sève brute des racines aux feuilles ;
- **le bois du cœur (duramen)** : il est composé de cellules mortes qui ont fini d'évoluer. Il sert de charpente pour l'aubier qui l'entoure. Les cernes marquent les couches successives d'aubier superposées au cours des années. C'est le support de l'arbre que l'on nomme aussi le bois parfait ;

- **les cernes** : ils correspondent aux couches de croissance annuelles. Leur comptage précis permet de connaître l'âge de l'arbre au moment de sa coupe ;
- **la moelle** : c'est le tissu végétal un peu mou au centre du tronc. Il se forme au début de la croissance de l'arbre et ne se modifie pas.

## 1.3 Croissance d'un arbre

### 1.3.1 Détails

Au cours de son existence, l'arbre croît en hauteur et en diamètre jusqu'à sa maturité. Puis, sa croissance se poursuit essentiellement en diamètre. La hauteur, la circonférence et la longévité des arbres sont très variables (*figure 1.9*). Ainsi, le hêtre commun a une espérance de vie entre 150 et 400 ans et atteint des hauteurs d'environ 45 mètres.

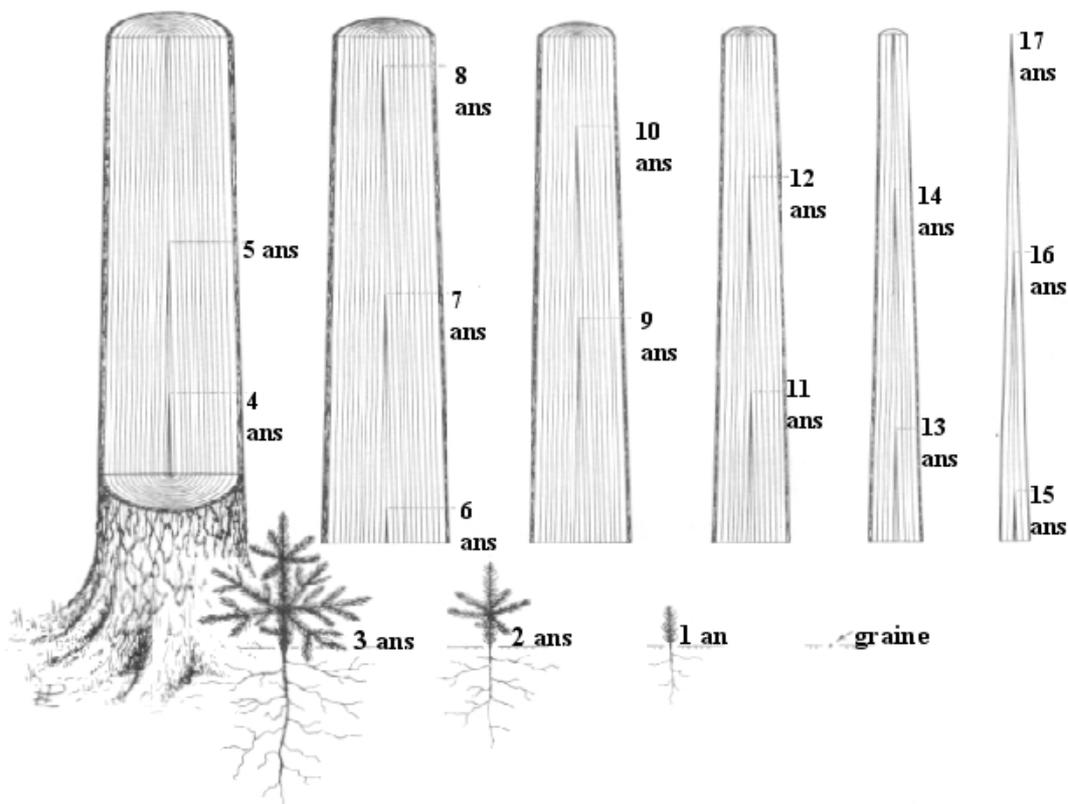


FIG. 1.9 – Croissance de la tige d'un arbre

Chaque année l'arbre se développe donc en longueur (ou hauteur) et en diamètre (ou épaisseur) principalement au cours du printemps :

- **la croissance en longueur** : la croissance en longueur des tiges de l'arbre (tronc est branche) est assurée par le développement des bourgeons terminaux situés à chaque extrémité de chaque tige. Ces bourgeons, après éclosion, peuvent faire s'allonger les branches de 10 à 15 cm en quelques jours. Dans le bourgeon terminal, il y a la tige qui est formée en miniature (en partie ou en totalité) avec ses feuilles et ses entre-nœuds. Tous ces organes, en se gorgeant d'eau, se développent pendant une période plus ou moins longue. A la pointe de la tige se trouve le méristème, zone à partir de laquelle sont fabriqués tous les organes constituant la future portion de tige ;
- **l'apparition de nouvelles branches** : en dehors de la possibilité de s'allonger, les végétaux ont aussi la faculté de se diviser. Pour ce faire, la plupart des plantes mettent en place des points de croissance au niveau de chaque feuille. A partir de ces points, des rameaux vont pouvoir se développer. Selon les essences, c'est telle ou telle feuille qui va présenter un rameau à son aisselle. Ainsi la ramification peut s'exprimer en donnant des étages de branches très groupées comme chez les pins ou plus lâches comme chez le merisier, ou bien être répartie tout le long de la tige comme chez l'aulne. Les différentes façons de disposer les rameaux sur la tige sont une des principales composantes de la forme des plantes ;
- **la croissance en largeur** : grâce à sa faculté de s'allonger et de se ramifier, la plante est capable de grandir dans plusieurs directions. Mais pour tenir debout et rester rigide, elle a besoin d'une autre possibilité : la croissance en diamètre. La croissance en épaisseur est le fruit du fonctionnement d'une fine pellicule annulaire située juste sous l'écorce : le cambium ou assise cambiale. Cette zone cellulaire produit les éléments du bois (alimentation en eau, stockage et rigidité mécanique des tiges) et les tissus (le liber) véhiculant la sève élaborée. Dès que cette couche est approvisionnée en sève elle se met à fabriquer du bois (aubier et liber). Durant le printemps, le bois fabriqué est très aéré avec de gros vaisseaux pour faire circuler les importantes quantités de sève brute (le bois). Mais au fil des jours les quantités d'eau présentes dans le sol s'amenuisent et diminuent le trafic de sève brute, ainsi le bois fabriqué devient plus compact et sombre. Ce mode d'expansion explique les cercles concentriques de couleurs claires puis foncées, qui se renouvellent chaque année, offrant ainsi la possibilité de chiffrer l'âge d'un arbre coupé en comptant les cercles clairs ou foncés. Le liber se renouvelle en un ou deux ans, poussé vers l'extérieur par l'expansion de l'aubier, il se dessèche, parfois se fissure, et renouvelle l'écorce.

Suivant l'épaisseur du cerne de croissance, on apprend si les conditions de développement étaient bonnes (cercle large) ou mauvaise (cercle fin). Les conditions sont favorables lorsque l'arbre n'est pas malade, qu'il a assez de place (voisinage), d'eau pour ses racines et de soleil pour ses feuilles.

### 1.3.2 Structuration en axes de croissance

La croissance des arbres est décrite par les botanistes à l'aide de la notion d'unité de croissance et par extension d'axe de croissance. Comme nous l'avons vu précédemment, la croissance d'un bourgeon peut être détaillée en deux étapes distinctes :

- généralement en un temps très court, un bourgeon éclot et se développe en produisant une série d'entre-nœuds (les nœuds sont les supports des feuilles et des nouveaux bourgeons) ;
- puis, sur un temps plus long, la taille des entre-nœuds augmente.

Cette portion de tige qui se développe depuis le bourgeon de l'année en court jusqu'au prochain bourgeon apical (qui apparaît à la fin de l'hiver de l'année en court) est appelée *unité de croissance*. Une unité de croissance est composée de nœuds. Un nœud peut :

- donner naissance à une feuille, une fleur, ou mourir ;
- donner naissance à un axe d'ordre supérieur, on parle alors de ramification. L'axe ainsi formé se comporte alors exactement comme un arbre (on boucle ainsi le processus de croissance). Il peut donc lui aussi donner naissance à de nouveaux bourgeons ou non. La ramification peut se faire de manière continue (tout bourgeon latéral donne naissance à un nouvel axe aussitôt), ou non. Les bourgeons peuvent attendre avant de se développer, on parle alors de bourgeons latents.

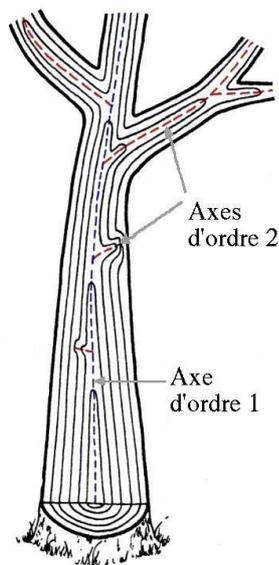


FIG. 1.10 – Coupe radiale et transversale d'un arbre

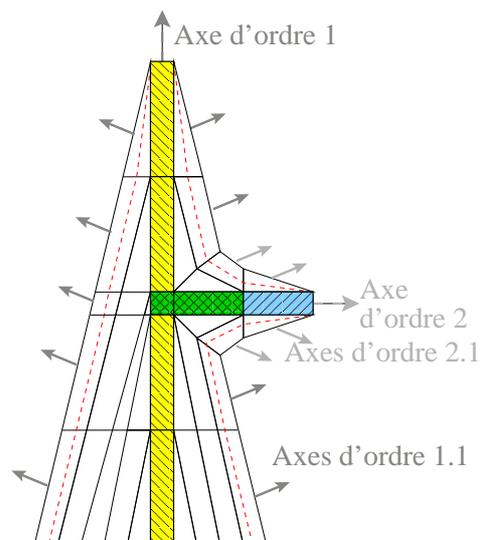


FIG. 1.11 – Structuration principale

Un axe de croissance peut être défini intuitivement comme un ensemble d'unités de croissances contiguës (i.e. reliées entre elles par le développement d'un bourgeon apical). La distinction des différents axes de croissance s'introduit en fonction du type de bourgeon qui se développe. L'apparition d'un nouvel axe de croissance est due au développement des bourgeons axillaires (nous n'approfondirons pas plus ce sujet par souci de clarté, mais nous pourrions remarquer que certaines espèces peuvent avoir plusieurs bourgeons terminaux qui vont donner naissance à plusieurs "troncs" qui vont se développer en parallèle. Il suffit alors de considérer que le tronc ne se développe plus mais que deux nouvelles branches apparaissent). Le développement d'un axe de croissance se fait tout simplement à partir de ses bourgeons terminaux.

La décomposition en axe de croissance s'opère de la façon suivante : le tronc est l'axe principal (axe d'ordre 1) ; sur l'axe principal sont insérées les branches principales (axe d'ordre 2). D'une manière plus générale sur l'axe d'ordre  $i$  ( $i > 0$ ) sont insérées les branches d'ordre  $i+1$  (axe d'ordre  $i+1$ ). Un axe est donc constitué d'un ensemble d'inter-nœuds créés les uns après les autres.

A l'aide de ces notions, la croissance des arbres peut maintenant se définir ainsi (voir figure 1.10) :

- **la croissance primaire** : c'est elle qui assure la croissance en longueur de l'arbre et l'apparition de nouvelles branches. La croissance de tout axe résulte de l'activité du bourgeon situé à son extrémité (le bourgeon terminal). L'apparition des branches principales et secondaires est due à la croissance de bourgeons terminaux et latéraux sur les axes supérieurs (voir figures 1.10 et 1.11) ;
- **la croissance secondaire** : c'est elle qui assure la croissance en épaisseur.



## Chapitre 2

# Techniques de modélisation

Dans une première partie, nous détaillerons les principaux travaux déjà existants dans nos domaines de recherches. Nous nous attarderons plus particulièrement sur trois articles majeurs qui ont inspiré notre travail. Dans un second temps, nous justifierons notre modèle. Pour cela, nous présenterons la structuration particulière des objets que nous manipulons. Nous décrirons la structuration des subdivisions d'objets volumiques en utilisant la simple notion d'orbite. Ensuite, nous montrerons quelques opérations qui clarifient et simplifient le contrôle du processus d'évolution. Nous distinguerons les opérations qui font évoluer la structure de l'objet (opérations topologiques) de celles qui modifient sa forme (opérations de plongement). Cette distinction, qui apporte une simplification des algorithmes, est un avantage bien connu des modèles topologiques (Bertrand *et al.*, 1993b). Pour finir, nous illustrerons notre modèle en proposant quelques exemples de modélisation.

### 2.1 État de l'art

Cette partie est construite à partir de (Terraz, 1994) et (Françon *et al.*, 1994).

Durant ces dernières années, de nombreux travaux sur la modélisation d'objets naturels ont vu le jour tels que (Reffye *et al.*, 1988), (Chen *et al.*, 1992), (Mech *et al.*, 1996), (Deussen *et al.*, 1998) et (Prusinkiewicz *et al.*, 2001). La difficulté commune à toutes ces approches provient des structures, parfois très complexes, des objets naturels à modéliser. Cependant, cette complexité résulte souvent de règles d'évolution biologique relativement simples. Le fait de mettre en application ces règles dans un *modèle d'évolution* est une idée-clé permettant de générer des familles complètes d'objets similaires en utilisant des paramètres faciles à contrôler (épaisseur du cerne, longueur de croissance, ...). La plupart des méthodes concernant la croissance des arbres ne s'intéressent qu'à une croissance externe (silhouette de l'arbre). Elles manipulent le plus souvent des subdivisions d'objets topologiques de dimension 0 ou 1 même si le résultat final est un volume ((Françon *et al.*, 1994) et (Jones *et al.*, 2000), pour avoir un panorama plus complet, le lecteur pourra se référer à (Deussen *et al.*, 2005)). Ces articles se basent sur des modèles géométriques ou morphologiques souvent très simplifiés. A l'op-

posé de ces modèles simplifiés, il existe des travaux récents en botanique proposant des modèles plus complets, comme par exemple : (Kurth, 1994), (Perttunen *et al.*, 1996), (Reffye *et al.*, 1997a), (Reffye *et al.*, 1997b), et (Kang *et al.*, 2002). Dans ces travaux, les auteurs proposent des modèles de croissance d'arbres intégrant de multiples facteurs tels que la photosynthèse (en fonction de la surface foliaire), la circulation de la sève dans l'arbre, la stabilité mécanique . . . Ces modèles permettent notamment de simuler la croissance secondaire (création des cernes de bois) et donc de calculer la production de bois dans l'arbre. L'application visée dans cette thèse étant en premier lieu la visualisation et le rendu pour la synthèse d'images, nous n'utilisons pas de tels modèles dans la suite, mais de simples éléments descriptifs comme les tailles moyennes de cernes ou les angles d'insertion des branches qui caractérisent les différentes espèces d'arbres ((Vigue, 2002) et (Barry-Lenger *et al.*, 1999)).

L'un des principaux axes d'études pour la simulation de la croissance d'arbre est l'utilisation de grammaires (L-system, . . .). Ces grammaires sont basées sur des règles de productions locales qui dépendent ou non du contexte. Nous y reviendrons plus en détail dans la *section 3*. D'autres approches ont bien sûr été développées telles que les systèmes de particules, la méthode AMAP (botANique et bio-inforMatique de l'Architecture des Plantes), . . .

Dans cette partie, nous nous intéresserons plus particulièrement aux méthodes permettant de simuler des évolutions d'objets, par opposition à celles ne permettant que la modélisation d'objets statiques. Notre étude se fera dimension par dimension en commençant par la dimension 0.

### 2.1.1 Objets de dimension 0

La méthode introduite par (Reeves, 1983), qui utilise des systèmes de particules, consiste à modéliser les objets non pas par des surfaces élémentaires mais par des flots de particules (objets de dimension 0) évoluant dans  $R^3$ .

Ainsi, une particule est identifiée comme un simple point dans l'espace. Chaque particule possède ses propres caractéristiques : une position de base, une durée de vie, une vitesse, . . . Ces grandeurs peuvent être déterminées aléatoirement ou non. Elle peut donc se déplacer, changer de couleur, apparaître ou disparaître, . . .

Les systèmes de particules sont surtout utilisés pour simuler des effets de masse, comme la simulation de déplacements de troupeaux (vol d'oies (Reynolds, 1987)), la simulation d'objets naturels et leurs déplacements. Ces objets peuvent être composés de nombreux sous-objets. Dans ce cas, ils permettent de simuler de la poussière, des étincelles, du feu, des chutes de neige ou de la pluie, . . .

Les systèmes de particules peuvent être structurés, comme pour la simulation de la croissance d'arbres botaniques ((Aono M., 1984) et (Reeves *et al.*, 1985)). Les particules ne sont donc plus indépendantes. Les branches de chaque arbre sont représentées comme un ensemble d'arêtes. Un algorithme récursif génère les branches à partir de paramètres et de relations permettant de définir le type de branchement, la distance entre deux sous-branches consécutives, l'angle de la branche par rapport au tronc, sa longueur et son épaisseur. Chaque branche hérite d'une grande partie des paramètres de ses parents

(éventuellement remis à l'échelle). La boucle s'arrête quand l'épaisseur de la branche atteint un seuil ou quand le niveau de profondeur voulu est atteint (*figure 2.1*).



FIG. 2.1 – Image issue de l'article de Reev85

(Y. *et al.*, 2003) propose de générer des arbres en simulant le transport de l'énergie des feuilles (et des racines) vers le tronc de l'arbre. Pour initialiser le processus, des particules sont réparties aléatoirement sur l'enveloppe de l'arbre que l'on cherche à modéliser (au niveau des feuilles ou des racines). Ensuite les particules se rapprochent entre elles, tout en convergeant vers le centre de l'arbre (un point cible). Lorsque deux particules sont trop proches l'une de l'autre, elles fusionnent et leurs énergies se cumulent (*figure 2.2*). Plus l'énergie est importante, plus la section modélisée est grosse. Bien que basés sur des constatations botaniques (plus la section modélisée transporte de sève, plus elle est grosse) et que les résultats soient plutôt réalistes (*figure 2.3*), le mode de génération ne suit pas un processus de croissance naturelle.

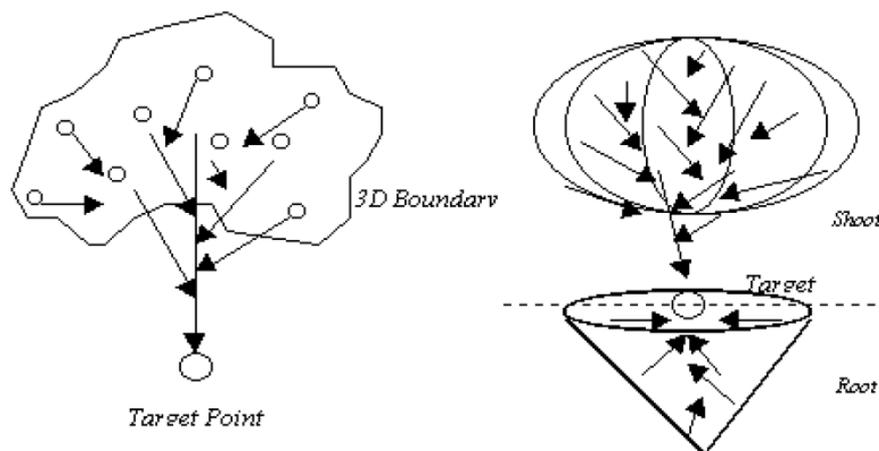


FIG. 2.2 – Diagramme du comportement des particules (à gauche) et modélisation de l'arbre (à droite)

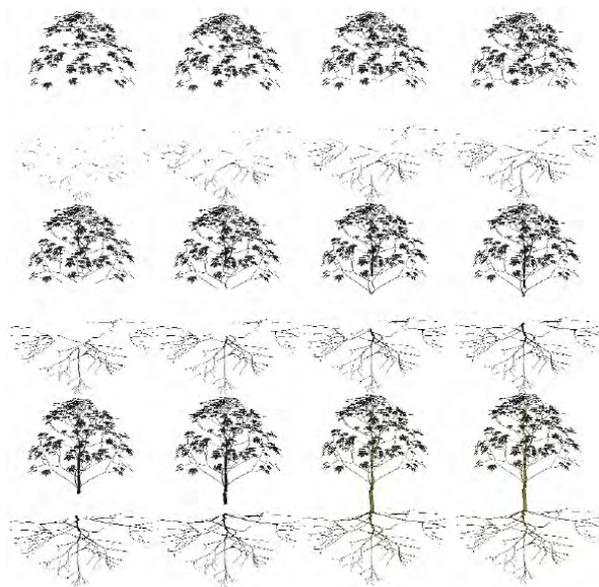


FIG. 2.3 – Développement de l'arbre

### 2.1.2 Objets de dimension 1

Afin de modéliser et simuler principalement la croissance d'arbres, plusieurs approches ont été proposées (L-systems, Amap, ...). Ces méthodes associent des formes aux complexes cellulaires, qui sont ici de dimension 1, c'est à dire à chaque sommet est associé un point de  $R^3$  et à chaque arête est associée une courbe dans  $R^3$ . Par exemple, pour modéliser un arbre naturel, nous pouvons associer à chaque arête un cylindre qui représentera une branche ou le tronc. Chaque sommet représentant ainsi un nœud de l'arbre naturel. Ici, le complexe cellulaire correspond donc au squelette de l'arbre.

Nous allons présenter plus particulièrement une des deux méthodes qui ont, en partie, inspiré notre travail sur la simulation de la croissance des arbres : l'Amap. Les L-systems seront présentés de manière plus approfondie dans la *section 3*. Il existe bien sûr d'autres méthodes. Une synthèse de celles-ci est reprise dans (Jones *et al.*, 2000).

#### Méthode AMAP

La méthode développée par l'Amap<sup>1</sup> (Reffye *et al.*, 1988) est une approche plus particulièrement spécialisée dans la modélisation de plantes. Elle a pour objectif de reproduire un modèle de croissance très proche de la réalité pour des besoins botaniques. Celle-ci a été mise au point grâce à des observations et des prises de mesures sur de véritables plantes. Les objets modélisés sont des complexes cellulaires de dimension 1 sans cycle (arbre combinatoire plongé). Dans l'exemple plus particulier des arbres, la structure développée correspond à celle d'un arbre naturel (*figure 2.4*) : un sommet du

<sup>1</sup>[www.cirad.fr](http://www.cirad.fr)

complexe cellulaire de dimension 1 correspond à un nœud de l'arbre naturel (la partie de l'arbre où les feuilles et les bourgeons sont insérés), une arête correspond à un entre-nœud (c'est la partie de l'arbre située entre deux nœuds). Les sommets et les arêtes sont structurés en unités de croissance, ce qui correspond au fait que la croissance des arbres se fait en deux étapes distinctes. En premier lieu, une série d'entre-nœuds est produite dans un bourgeon (généralement en un temps très court), définissant ainsi une unité de croissance. Ensuite, la taille des entre-nœuds augmente (généralement dans un temps plus long). Comme le montre la (figure 2.4), les unités de croissance sont structurées en axes. Le tronc est l'axe principal (axe d'ordre 1). Sur l'axe principal sont insérées les branches principales (axe d'ordre 2). Le système se généralise ainsi : sur l'axe d'ordre  $i$  sont insérées les branches d'ordre  $i+1$  (axe d'ordre  $i+1$ ), sur l'axe d'ordre  $i+1$ , les branches d'ordres  $i+2 \dots$ . Un exemple d'arbre est donné par la (figure 2.5)<sup>2</sup>.

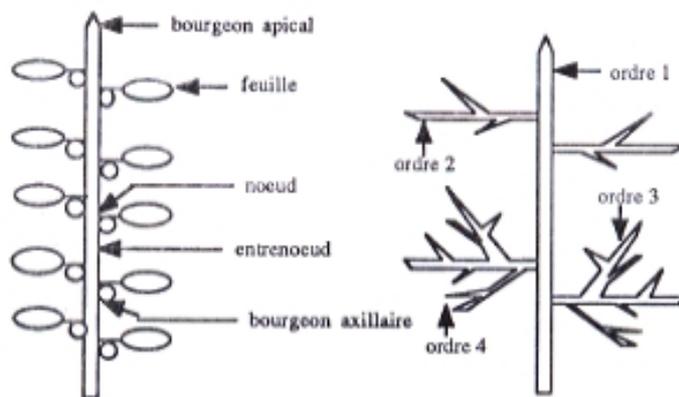


FIG. 2.4 – Structure d'un arbre naturel



FIG. 2.5 – Exemple d'arbre obtenu à partir du logiciel NatFX

### 2.1.3 Objets de dimension 2

Ici les complexes cellulaires sont de dimension 2, c'est à dire composés de sommets, d'arêtes et de surfaces. Ils sont surtout utilisés pour modéliser l'évolution de phénomènes naturels tels que la croissance de fleurs, de feuilles ou encore la métamorphose de papillons. Ils sont aussi utilisés dans les simulations de développement cellulaires (morphogénèse).

<sup>2</sup>Image extraite de [http://www.3dvf.com/modules/publish/Logiciels\\_1389\\_1.html](http://www.3dvf.com/modules/publish/Logiciels_1389_1.html).

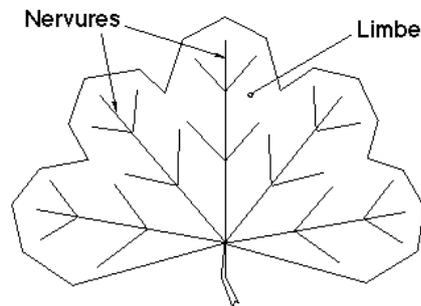


FIG. 2.6 – Structure d'une feuille

### Maps L-systems

(Prusinkiewicz *et al.*, 1990b) propose dans son livre de simuler les divisions cellulaires en introduisant une extension des L-systems en dimension 2 : les Maps L-systems (ou L-systems de carte). Nous ne détaillerons pas plus cette approche dans cette partie car elle sera reprise dans la *section 3*.

### Modélisation de feuilles

Contrairement aux systèmes classiques de génération de feuilles qui manipulent des objets de dimensions 0 ou 1 ((Y. *et al.*, 2003), (Prusinkiewicz *et al.*, 2001)), certains travaux ont été effectués en manipulant directement des objets de dimension 2. Ils s'appuient sur un modèle topologique qui permet de représenter la topologie de toute subdivision de dimension 2, les 2Gmaps (Lienhardt, 1994).

(Lienhardt *et al.*, 1987), (Lienhardt, 1995) aborde la modélisation de feuilles végétales par une approche combinatoire. Les auteurs se basent sur l'utilisation de connaissances botaniques appliquées à un modèle où la topologie prime sur la géométrie pour simuler l'évolution d'une feuille au cours du temps. Le modèle est basé sur la constatation suivante : le limbe d'une feuille forme une surface s'appuyant sur des nervures constituant elles-mêmes un arbre enraciné (*figure 2.6*). Le développement de l'architecture des nervures au cours du temps est assez proche de celui des axes d'un arbre. Les notions propres aux axes des arbres (ramification, mortalité, rythme,...) peuvent donc être étendues aux nervures d'une feuille. La métamorphose de la surface d'une feuille est obtenue en simulant simplement la métamorphose de l'arbre qui la constitue (les nervures). Puis, pour obtenir un aspect plus naturel, une simple texture est plaquée sur la surface.

Ce modèle permet de générer des feuilles d'une grande diversité de formes. Comme avec la génération d'arbres (Terraz *et al.*, 1995), le temps de calcul d'une feuille peut varier de quelques secondes à plusieurs minutes, pour obtenir des graphes de quelques centaines à plusieurs milliers de facettes triangulaires.



FIG. 2.7 – Dégradation des feuilles d'une branche de rosier (de la gauche vers la droite, les feuilles sont affectées par une altération des couleurs, des attaques d'insectes, et des flétrissements)

(Peyrat *et al.*, 2008) propose d'introduire, un formalisme basé sur les Maps L-systems, les 2Gmaps L-systems paramétriques. Ce modèle est basé sur l'approche décrite dans la *section 3.2* mais elle est restreinte à la dimension 2. Ici, seulement trois types d'opérations sont utilisés pour décrire le développement des feuilles : l'ajout, l'identification et la subdivision. Puis ce modèle est complété pour introduire les effets liés au vieillissement (flétrissement et décoloration des feuilles, ...) (*figure 2.7*) .

### 2.1.4 Objets de dimension 3

Ici, les objets manipulés sont des complexes cellulaires de dimension 3, c'est à dire composés de sommets, d'arêtes, de surfaces et de volumes. A notre connaissance, il n'existe que quatre types d'approches qui utilisent directement des objets topologiques de dimension 3 :

#### Cellworks L-systems

Des expériences ont été entreprises pour étendre les Maps L-systems aux objets de dimension 3 (Prusinkiewicz *et al.*, 1990b). On les appelle les Cellworks L-systems. A chaque symbole on associe ici un morceau de surface qui représente la membrane, ou le bord de la cellule. Mais selon ses auteurs, cette méthode a été abandonnée car elle n'est pas concluante et difficilement extensible à d'autres objets que les cellules. Nous y reviendrons plus en détails dans la *section 3* .

## Simulation de la croissance du bois dans un espace de voxels

(Buchanan, 1998), propose de générer des textures volumiques de bois en simulant la croissance du bois. Cette croissance est basée sur les règles botaniques classiques de croissance. Dans ce modèle, l'arbre croît dans un espace de voxels (une matrice). L'étude est restreinte à la simulation de la croissance de bois dit tendre (un seul type de cellule est géré pour la croissance alors que le bois dur en compte au moins 4). L'utilisateur doit imposer dès le départ de la simulation l'emplacement des nœuds de l'arbre dans la matrice représentant l'espace de croissance. Durant le processus de croissance, chaque voxel du bord de l'arbre peut se dupliquer ou mourir selon une fonction qui dépend du type de croissance (radial ou axial), de son voisinage et de sa saison (voxel de printemps ou voxel d'été). Si un nœud est rencontré, une branche est créée et se comporte ensuite comme un arbre.

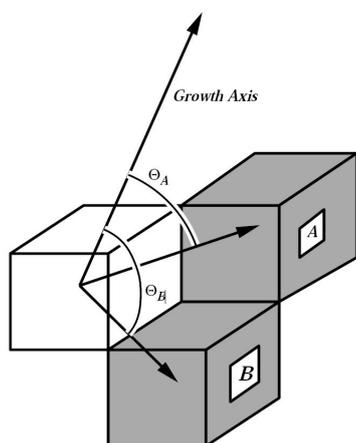


FIG. 2.8 – Détermination des voxels voisins dans le plan radial

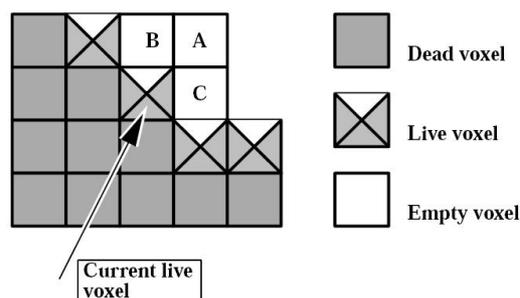


FIG. 2.9 – Vue 2D du choix de sélection du voxel (A, B, C sont trois voxels cibles pour le voxel pointé par la flèche)

Quand les cellules vivantes d'un voxel se divisent, elles doivent être distribuées à un des voxels voisins vides. Pour déterminer les voxels voisins candidats, l'auteur utilise la direction fournie par l'axe de croissance (ou la normale à la surface locale dans le cas de la croissance radiale) pour définir une direction de propagation. Le voxel cible doit être près du plan radial défini par l'axe de croissance et le voxel père. Une fois que tous les voxels cibles sont répertoriés, le choix définitif est effectué en comparant les angles formés entre l'axe de propagation et les centres du voxel père et des voxels candidats (*figure 2.8*). Un exemple de détermination du voxel suivant est donné par (*figure 2.9*). Bien que la normale extérieure indique que le Voxel A est un bon candidat pour recevoir les nouvelles cellules produites dans le voxel père, les voxels B ou C sont de meilleurs candidats car le voxel B est en contact direct avec trois voxels sur la surface et est ainsi tri-connecté et C quadri-connecté à la surface. Les auteurs distinguent donc les deux

types de croissance :

- la croissance axiale ou croissance en longueur. Elle est simulée en dupliquant un certain nombre de fois les voxels marqués comme étant au bout des branches ou du tronc et dans une direction radiale ;
- la croissance radiale : durant la croissance radiale du tronc, les cellules situées sur la partie extérieure du cambium se multiplient. Cette multiplication est simulée par le développement des voxels de la périphérie du tronc vers l'extérieur.

Une fois le processus de croissance terminé, une couleur est associée à chaque voxel de l'espace :

- une couleur claire pour un voxel de printemps,
- une couleur plus foncée pour un voxel d'été,
- rien pour un voxel vide.

Il en résulte ainsi un bloc de texture 3D dans lequel des objets sont plongés afin de valider les résultats. Pour créer des transitions plus douces, les auteurs introduisent une fonction de transfert de couleur qui tient compte des couleurs des voxels voisins pour le calcul définitif d'un point de l'espace ( *figure 2.10* et *figure 2.11* ). Bien que cette technique présente des résultats visuellement intéressants, elle présente également quelques problèmes spécifiques :

- le nombre et la position des nœuds doivent être fixés avant que la simulation du processus de croissance de l'arbre ne démarre ;
- les cernes ont un aspect non naturel dû à l'utilisation de voxels (effet d'escalier, . . . ) ;
- si l'article est replacé dans son contexte historique (1998), nous pourrions aussi reprocher des temps de calculs très longs (6-10 heures sur un PII avec 128 Mo de mémoire) pour une texture 3D de résolution 256\*256\*256. Cependant, avec les performances des machines actuelles, ce défaut doit s'être considérablement estompé.



FIG. 2.10 – Planche de bois coupé dans le sens axial

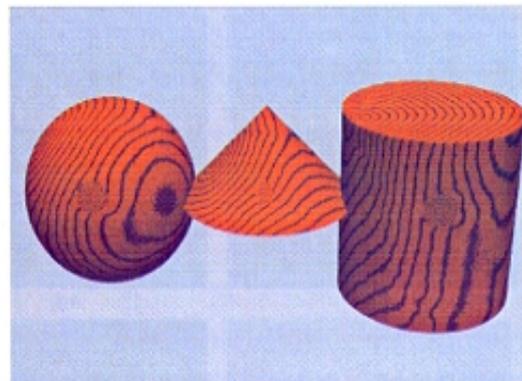


FIG. 2.11 – Application de la texture sur des objets

### Approche s'appuyant sur le logiciel de l'AMAP

Dans (Szafran *et al.*, Octobre 2003), les auteurs (botanistes et informaticiens) étudient des coupes d'arbres virtuels afin d'obtenir une sélection des meilleurs spécimens d'arbres pour l'industrie du bois. Ils décrivent la croissance d'un arbre comme une juxtaposition de couches de bois en forme de cône, qu'ils appellent des troncs de cônes ( *figure 2.12* ) et ( *figure 2.13* ).

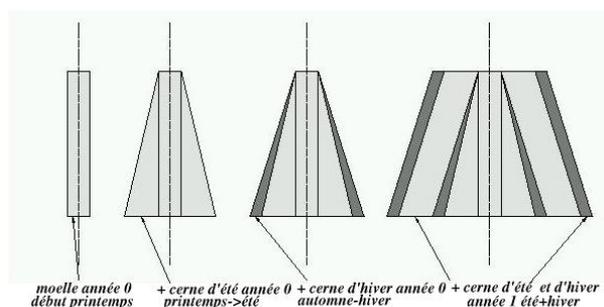


FIG. 2.12 – Modèle de croissance

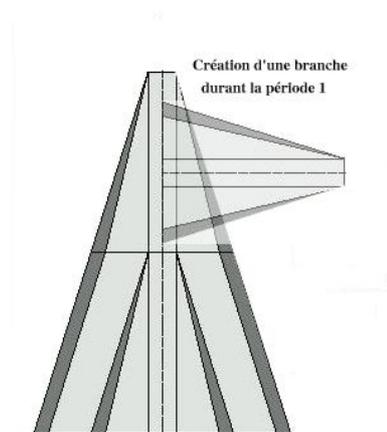


FIG. 2.13 – Gestion de la branche

Selon les auteurs le point le plus complexe est d'assurer une bonne gestion de la connexion entre le tronc et la branche au niveau des cernes. Ils proposent deux approches pour gérer ce problème :

- une approche surfacique
- une approche implicite

L'approche surfacique est l'approche la plus naturelle par rapport au modèle d'arbre développé dans le logiciel issue de l'AMAP. La première étape consiste à définir un modèle 3D continu géométrique, c'est à dire, de définir des opérations pour relier les entre-nœuds, les tiges et les branches à chaque niveau dans l'espace 3D. Ces raccordements exigent d'autres surfaces que des cônes et des cylindres. Les auteurs utilisent des tenseurs de surfaces bicubiques de Bézier. Ces surfaces sont définies par une structure géométrique polyédrale qui permet de relier les surfaces de manière naturelle. Pour calculer ces structures, chaque cerne est subdivisé en section. Chaque section d'un anneau est approchée par quatre, huit (ou plus) courbes cubiques de Bézier qui sont reliées entre elles de manière continue (G1). Ainsi chaque entre-nœud peut être défini par 4, 8 (ou plus) tenseurs de surfaces de Bézier.

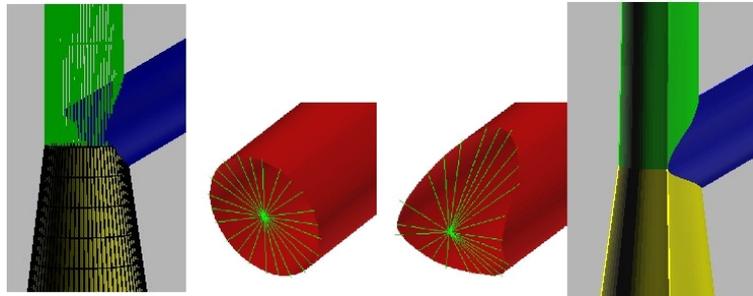


FIG. 2.14 – Calcul de l'intersection

Pour calculer les intersections entre les cernes du tronc et des branches, les auteurs supposent que les surfaces de Bézier sont très proches de leur structure polyédrale. Ils calculent donc l'intersection entre les structures, et en déduisent ainsi les nouvelles surfaces de Bézier (figure 2.14).

Puis, ils découpent des planches dans le nouveau modèle obtenu (figure 2.15). Le résultat est obtenu en calculant l'intersection entre le modèle et les plans qui délimitent la planche.

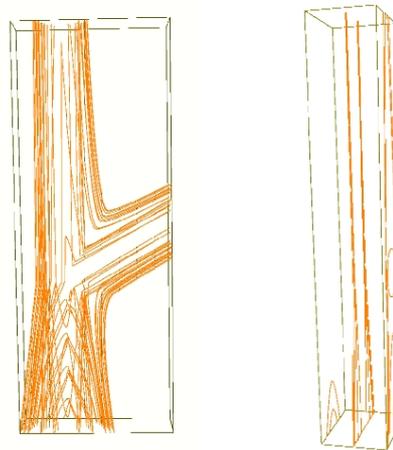


FIG. 2.15 – Planches obtenues à partir du premier modèle

Cependant, cette approche 3D continue dépend fortement du calcul des intersections 3D de ces surfaces. C'est la difficulté majeure de cette méthode. Selon les auteurs eux-mêmes, les calculs de raccords entre les surfaces 3D qui représentent les cernes du tronc et des branches, soulèvent des problèmes car les tailles de la tige et des branches sont trop différentes.

Pour répondre à ce problème, les auteurs proposent une autre approche, une approche implicite. Dans cette seconde méthode, il n'est pas possible d'obtenir une représentation volumique, mais simplement une matrice de pixels. La croissance d'un arbre (et de ses branches) est ici simulée sans tenir compte des problèmes de connexions

(ceci seront pris en compte d'une manière graphique ultérieurement). Il en résulte un recouvrement volontaire des objets appartenant aux cernes du tronc et aux branches (*figure 2.13*). Le modèle de croissance est analogue au cas précédent, sauf qu'au lieu d'utiliser des courbes de Bézier, les auteurs utilisent des interpolations linéaires entre les courbes paramétrées définies à hauteur des entre-nœuds.

La découpe de planche consiste à couper le pseudo-arbre ainsi généré par un plan quelconque. Pour chaque point de ce rectangle, il faut calculer la couleur du point qui est la couleur d'un des cernes dans laquelle il se trouve :

- si le point est à l'extérieur de l'arbre, alors sa couleur est la couleur extérieure.
- si le point se trouve sur plusieurs cernes, alors sa couleur sera celle du cerne le plus âgé (il faut sous entendre ici le cerne qui apparaît en premier lors de la croissance de l'arbre).

Un cerne d'âge donné d'un arbre est déterminé par l'union de toutes les unités de croissance de l'arbre (tronc et branches) de même âge (qui apparaissent à la même période). Pour déterminer à quel cerne appartient le point, tous les cernes de l'arbre sont parcourus en partant du plus jeune. Le processus s'arrête au premier cerne d'âge  $i$  qui ne contient pas le point. Le point est alors dans le cerne d'âge  $i-1$  et en dehors si  $i$  est nul.

Ce premier modèle conduit à dessiner un arbre dont les raccords entre le tronc et les branches sont pris en compte, mais ne sont pas très réguliers (apparition de discontinuités franches). Afin d'améliorer la régularité des raccords, les auteurs introduisent un paramètre de continuité. Ce paramètre permet de détecter les points qui sont au voisinage de plusieurs branches. Comme précédemment, tous les cernes de l'arbre sont parcourus en partant du plus jeune, mais cette fois-ci, le voisinage est pris en compte.

**Définition 2.1** Soit  $M$  un point de l'espace  $R^3$  et  $V$  un volume défini par une fonction d'évaluation  $F$ .

On dit que  $M$  appartient au volume  $V$  si  $F(M) > 0$ .

Si  $F(M) = 0$ , alors le point est dit sur la frontière du volume.

On dit que  $M$  est au voisinage de  $V$  avec un seuil  $\alpha$  si  $\alpha < F(M) < 0$ .

Dans le cas présent les volumes sont des cernes coniques. Ils sont donc approchés par des troncs de cônes dont la base et le sommet sont des cercles contenus dans des plans parallèles. Un cône est ainsi défini par les deux cercles le constituant. Les cercles sont définis par leur centre ( $O1$  et  $O2$ ), leur rayon ( $r1$  et  $r2$ ) et leur normale commune ( $N$ ) comme indiqué sur (*figure 2.16*). Cette approche permet de simplifier grandement le calcul de la fonction d'évaluation. De plus, les auteurs associent à chaque branche un paramètre de continuité  $A$  ( $A > 0$ ) qui permet d'indiquer la zone d'influence des cernes de la branche sur son environnement proche. Si le point est en dehors du cerne courant et que la fonction d'évaluation  $F$  appliquée au point  $M$  est supérieure à un seuil ( $-A$ ) alors la quantité qui représente le voisinage de  $M$  est incrémentée. Une fois tous les cernes parcourus, nous avons le cas échéant, le cerne qui contient le point  $M$  et l'influence de chaque cerne voisin sur le point  $M$ . Ceci permet en particulier d'effectuer des raccords elliptiques entre les cernes de même âges et de branches différentes (*figure 2.17*).

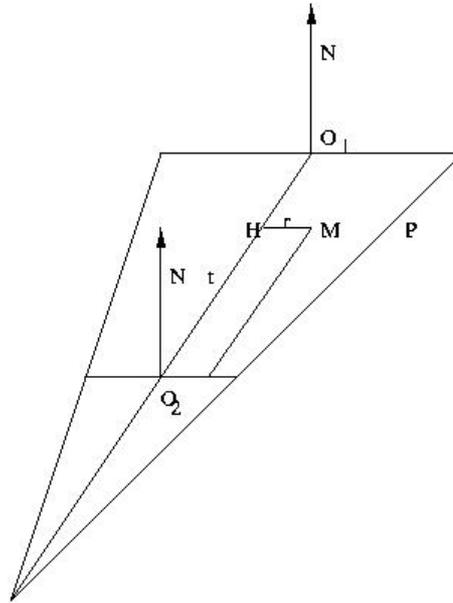


FIG. 2.16 – Paramètres des troncs de cônes

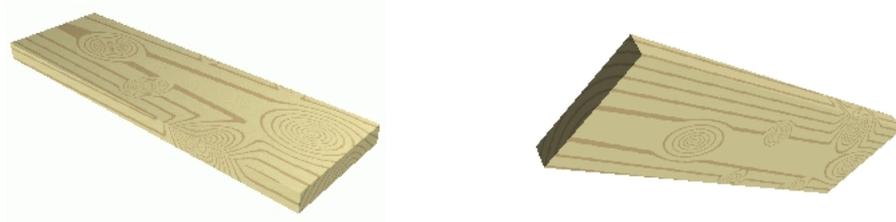


FIG. 2.17 – Planches obtenues à partir du second modèle

Dans cette seconde méthode (une approche implicite), il n'est pas possible d'obtenir une représentation volumique, mais simplement une matrice de pixels. La croissance d'un arbre (et de ses branches) est simulée sans tenir compte des problèmes de connexions. Il en résulte un recouvrement des objets appartenant aux cernes du tronc et aux branches. Les raccordements entre le tronc et les branches sont pris en considération d'une manière graphique en employant une fonction qui réalise un mélange des couleurs.

### ToPLSystem

Dans (Terraz, 1994) et (Terraz *et al.*, 1995), les auteurs proposent une approche procédurale permettant de calculer des métamorphoses d'objets (c'est à dire des évolutions durant lesquelles la forme, mais également la structure de l'objet peut varier). Pour

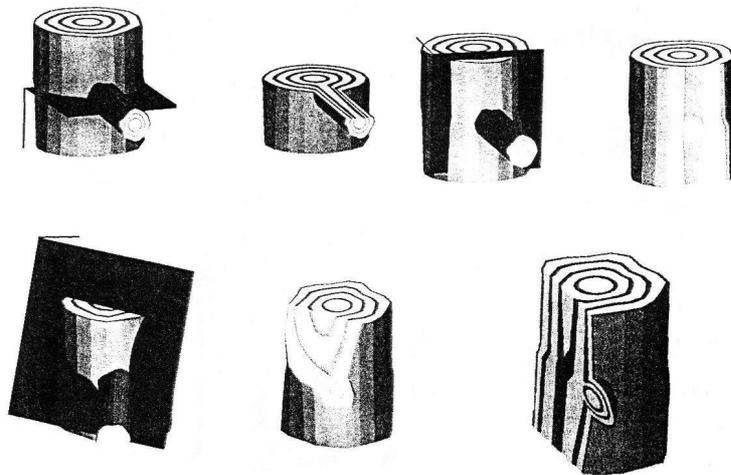


FIG. 2.18 – Images de la méthode de Ter94

cela, ils utilisent un modèle topologique, basé sur les 3Gcartes, qui permet de manipuler directement des subdivisions de volumes ToPLSystem<sup>3</sup>. Cette approche généralise les méthodes précédemment citées aux métamorphoses de toutes subdivisions de toutes surfaces et aussi de subdivisions d'espaces de dimension 3 (i.e. composées de sommets, arêtes, faces et aussi volumes). Afin de simplifier le contrôle de l'évolution d'objets qui peuvent être complexes, la méthode définit une structuration des objets en sous-objets de différents niveaux. Le contrôle de l'évolution se fait notamment au travers du contrôle d'un arbre de filiation lui-même structuré en axe de différents ordres, eux-mêmes composés de volumes, de faces, . . . Des opérations de différents niveaux sont définies sur ces différents sous-objets (ramification d'un arbre, allongement d'un axe, déformation d'un volume, collage de faces, . . .

En ce qui concerne la croissance interne du bois, les sous-objets manipulés sont les suivants : un arbre de filiation représentant le squelette de l'arbre, des axes de différents ordres représentant le tronc et les branches, des ensembles de volumes représentant les cernes et les sous-objets *classiques* qui sont les volumes, faces, arêtes et sommets. Des opérations sont définies sur chacun de ces objets afin de simplifier le contrôle de l'évolution. Quelques résultats sont illustrés sur la (figure 2.18). Bien que la trame de l'algorithme général de croissance soit simple à comprendre (par exemple l'épaississement du tronc ou d'une branche consiste à ajouter des volumes à partir des faces situées sur le pourtour des cernes existants), son contenu est plus complexe à mettre en place, et il en est de même pour les opérations d'ajouts et de recollements de volumes élémentaires. De plus, ce modèle est incomplet d'un point de vue botanique (il n'est que partiellement basé sur des données botaniques), d'un point de vue développement (de nombreux phénomènes peuvent être simulés) et la structure générée n'est pas exploitée.

<sup>3</sup>ToPLSystem propose une approche procédurale permettant de calculer les métamorphoses (i.e. des évolutions durant lesquelles non seulement la forme mais également la structure de l'objet varie) d'objets qui sont d'un point de vue géométrique, des objets surfaciques ou volumiques.

C'est pourquoi, nous avons décidé de reprendre ce modèle sous une approche différente (approche plus appliquée, basée sur la botanique) et de l'enrichir. Ainsi, le lecteur pourra trouver une étude plus approfondie des outils techniques dans (Terraz, 1994) et une présentation de l'évolution de notre modéleur basé sur ces travaux en annexe.

### 2.1.5 Logiciels

Les méthodes que nous venons de citer ont inspiré quelques produits commerciaux. La majorité des ces derniers proposent des modèles visuellement réalistes plutôt que des modèles botaniquement réalistes. Dans de tels modèles, c'est l'utilisateur qui configure de manière plus ou moins prononcée la forme de la plante qu'il souhaite modéliser. La société ONYX, avec son produit OnyxTreeMaker, propose un système fortement



FIG. 2.19 – Exemple de plantes modélisées avec OnyxGarden 6

interactif pour produire des arbres. Les utilisateurs configurent de manière interactive les niveaux d'embranchements d'un arbre graphiquement. Par exemple, les utilisateurs peuvent couper une tige et recevoir la rétroaction immédiate sur la géométrie nouvellement créée. Ce système permet de produire et modifier des modèles plus facilement, mais il est assez complexe à configurer. Ce principe a été étendu à la modélisation de plantes, de fleurs ou encore de bambous (*figure 2.19*)<sup>4</sup>. Ces logiciels se retrouvent dans la suite OnyxGarden 6.

La société Greenworks Organic Software commercialise le produit XFROG. XFROG est un programme de modélisation et d'animation d'objets 3D organiques. Ses développeurs ont voulu allier l'intuitivité des modèles procéduraux avec la puissance des méthodes basées sur des règles de productions (Lintermann *et al.*, 1999). Le système de modélisation développé dans XFROG implémente avec un certain succès cette approche

<sup>4</sup>Image extraite du site officiel d'Onyx : <http://www.onyxtree.com>

combinatoire (*figure 2.20*) . Il en résulte un modèle hautement configurable, qui peut produire n'importe quel forme de plante et dont le résultat est visible directement. Leur solution emploie une description par graphe. Les nœuds du graphe sont des composants mathématiques qui représentent des structures (arbre, branche, ...) ou des comportements (distorsion locale). La génération d'une plante se fait en plusieurs temps :

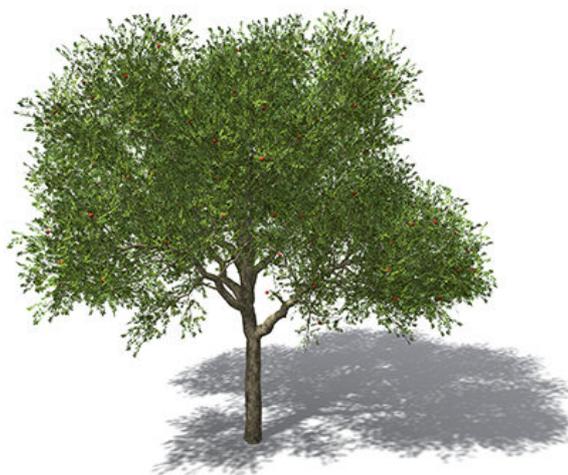


FIG. 2.20 – Arbre produit avec le logiciel XFROG de la société Greenworks

- l'utilisateur crée un premier graphe en fournissant une information structurale générique (l'utilisateur n'est pas obligé de fournir explicitement la gestion de toutes les branches) ;
- le graphe est ensuite étendu par propagation de l'information structurale. Grâce à une classe spéciale de composants, les nœuds fils du graphe sont multipliés de manière algorithmique (ce qui permet d'automatiser le processus de ramification). Il en résulte un nouveau graphe beaucoup plus complet que le premier ;
- enfin le système reparcourt le nouveau graphe pour pouvoir générer la géométrie associée.

Une fois le graphe produit, l'utilisateur peut continuer à configurer les paramètres et visionner immédiatement les modifications qu'il apporte (*figure 2.21*)<sup>5</sup>. Ce programme dispose également d'un module qui permet de gérer le temps pour créer des animations. Comme nous le verrons par la suite, nous réutiliserons cette classification hiérarchique du graphe. Il faut cependant noter que ce modèle ne respecte pas vraiment les principes de croissance botanique, c'est à l'utilisateur de créer lui-même les modèles et de configurer les évolutions au cours du temps (croissance du tronc, apparition de branches, ...).

Les travaux de recherche effectués au sein du CIRAD depuis 1978 (projet AMAP), ont conduit à l'élaboration d'un logiciel informatique qui simule la croissance des végé-

<sup>5</sup>Image extraite du site <http://www.grmstudio.it/software/XFROG/XFROG4.2/xfrog4.2.html>

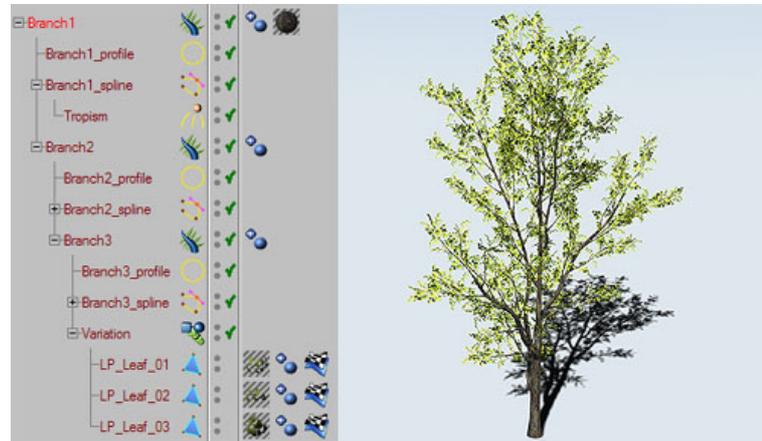


FIG. 2.21 – Interface d'XFROG : à gauche hiérarchie de l'arbre et à droite interprétation visuelle du graphe

taux tout en respectant leur programme génétique et restituée, en images de synthèse 3D, la morphologie et l'esthétisme naturel de la plante. Cette nouvelle technologie, dénommée *AMAP<sup>TM</sup>* (Atelier de Modélisation et d'Architecture des Plantes) a rapidement acquis une importante notoriété auprès de la communauté scientifique internationale et a donc été commercialisée par la société JMG Graphics qui est devenue Bionatics. Pour pouvoir toucher un large public, une gamme complète de produits dérivés a été développée :

- une modélisation de plantes 3D dynamique destinée aux professionnels de l'animation 3D (natFX) ;
- une modélisation de plantes 3D simple et réaliste destinée aux architectes, urbanistes, paysagistes (EASYnat) ;
- une modélisation optimisée de plantes 3D destinée aux environnements temps réel pour les simulateurs, jeux vidéos (REALnat) ;
- et bien d'autres : BLUEBERRY3D, LandSIM3D ...

Cette suite de logiciels est un peu particulière car son but premier est de créer des modèles botaniquement réalistes (il ne faut pas oublier que ce modèle a été développé pour des besoins botaniques à l'origine). Et donc, contrairement au logiciel XFROG, le modèle généré est soumis à de fortes contraintes botaniques. Ici l'élément de base est la graine. L'utilisateur charge une graine qui est fournie avec son code génétique, et peut configurer un nombre assez restreint de paramètres (âge, saison, floraison, ...). Il ne peut pas configurer par exemple le comportement d'une branche en particulier, tout est automatiquement configuré grâce au patrimoine génétique de la plante. Ensuite, l'utilisateur peut au choix lancer une simulation complète de la croissance (une vidéo) ou obtenir directement la plante à un âge donné. Chaque simulation donne une plante unique qui dépend du fonctionnement de ses bourgeons, de ses méristèmes et d'éventuels accidents de croissance.

L'avantage principal de cette technique est de pouvoir générer rapidement de nom-

breuses plantes sans avoir à les configurer manuellement et d'obtenir des représentations très réalistes (*figure 2.22*)<sup>6</sup>. Cependant cette technique ne gère pas les recolllements entre le tronc et les branches. Il en résulte un chevauchement entre les volumes constitutifs du tronc et des branches, mais visuellement cela ne se voit si l'arbre est observé de très près.

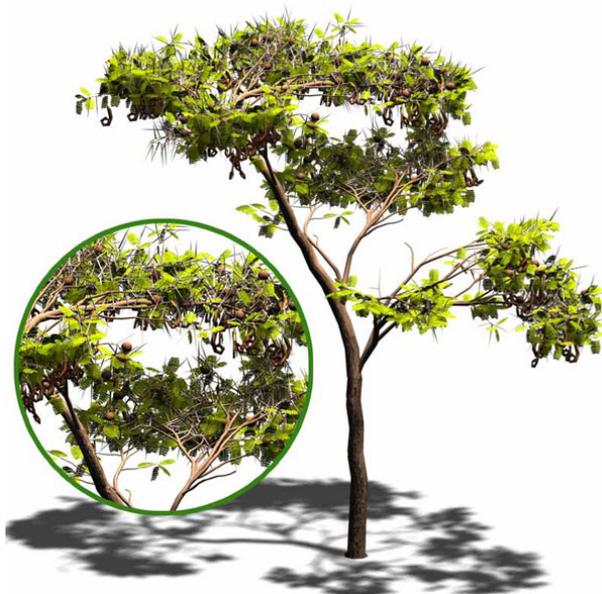


FIG. 2.22 – Exemple d'arbre produit avec le logiciel natFx de la société bionatics

Il existe bien sûr d'autres modèles de croissance :

- Forester Arboretum (<http://www.dartnall.f9.co.uk/>),
- POV-tree (<http://propro.ru/go/Wshop/povtree/povtree.html>),
- ...

Tous les travaux mentionnés ci-dessus ne concernent qu'une simulation de la structure du bois ou de l'arbre, grâce à un processus de croissance. Il existe d'autres méthodes qui se basent sur une approche différente. Leur objectif global n'est pas la simulation de bois à travers un processus de croissance, mais seulement la génération de textures de bois visuellement réalistes. Nous y reviendrons dans la *section 4.2*.

Comme nous venons de le voir, beaucoup de méthodes se sont intéressées à la croissance des arbres, mais peu de méthodes ont utilisé ce mode de croissance pour générer des structures internes botaniquement réalistes.

Dans ce travail, nous proposons d'employer un modèle topologique augmenté en dimension 3 pour représenter la structure interne du bois en simulant sa croissance. Notre méthode est basée sur les deux principes suivants : le respect des règles botaniques afin de réaliser un modèle proche du bois réel, et la division de la structure volumique du bois aux dimensions inférieures afin de contrôler plus précisément le processus d'évolution.

<sup>6</sup>Image issue du site [www.hxsd.com.cn/portal/news/5/166.html](http://www.hxsd.com.cn/portal/news/5/166.html)

Notre modèle propose :

- une simplification du processus de croissance grâce aux subdivisions du volume du bois. A chaque étape, les volumes représentant le cerne généré à l'étape précédente sont divisés afin de générer le nouveau cerne (la nouvelle unité de croissance) ;
- l'insertion appropriée d'une nouvelle branche avec une répartition spatiale correcte des cernes autour des nœuds (botaniquement un nœud est l'origine d'une branche). Les connexions entre les éléments du tronc et ceux des branches sont exactement manipulés, il en résulte une distribution spatiale plausible des cernes autour des nœuds.

## 2.2 Cartes généralisées et cartes généralisées augmentées

Les cartes généralisées (Lienhardt, 1989), (Bertrand *et al.*, 1993b), (Bertrand *et al.*, 1993a) font parties des modèles topologiques de représentation par les bords (Boundary Representation ou B-Rep) au même titre que les arêtes ailées. Ce modèle permet de représenter la topologie de toute subdivision régulière, orientable ou non, avec ou sans bord. D'un point de vue technique, il permet également de simplifier la définition des outils et des mécanismes utilisés pour simuler des métamorphoses de subdivisions d'objets volumiques, ainsi que leur implémentation (cf. *section 2.3* et *section 2.4*).

Dans cette section, nous introduirons les cartes généralisées de dimension 3 ou 3G-cartes. Puis, nous présenterons diverses notions associées qui seront utiles pour définir de manière rigoureuse les opérations que nous avons développées. La relation entre ces objets combinatoires et les objets géométriques qui peuvent leur être associés ainsi que les opérateurs que nous définirons pour la manipulation des 3G-cartes pour notre modèle seront étudiés dans les sections suivantes.

Dans toute la suite de cette partie, nous définirons les notions sur les cartes directement en dimension 3, mais elles sont valables aussi en dimension  $n$  ( $n \in \mathbb{N}$ ,  $n > 0$ ) moyennant parfois quelques restrictions.

### 2.2.1 Définition des 3Gcartes

**Définition 2.2** Une carte généralisée de dimension 3, ou 3G-carte est définie par un 5-tuple  $G = (\mathcal{B}, \alpha_0, \alpha_1, \alpha_2, \alpha_3)$ , où :

- $\mathcal{B}$  est un ensemble fini d'éléments appelés brins ;
- $\forall i \in [0, 3]$ ,  $\alpha_i$  est une bijection sur  $\mathcal{B}$  ;
- $\forall (i, j)$ ,  $0 \leq i < i+2 \leq j \leq 3$ ,  $\alpha_i \circ \alpha_j$  est une bijection sur  $\mathcal{B}$  sans point fixe.

Intuitivement, les brins peuvent être vus comme des demi-arêtes et les bijections  $\alpha_i$  peuvent être définies de la façon suivante :

- $\alpha_0$  : connecte deux brins pour former une arête «simple»(arête incidente à une face simple) ;
- $\alpha_1$  : connecte deux brins pour former un sommet «simple»(sommet incident à une face «simple») ;

- $\alpha_2$  : connecte deux faces «simples» incidentes au même volume le long des brins des arêtes de leur bord ;
- $\alpha_3$  : connecte deux volumes le long des brins des faces «simples» de leur bord.

Les définitions théoriques de ces bijections sont données dans (Lienhardt, 1994). La (figure 2.23) montre les détails des bijections  $\alpha_0, \alpha_1, \alpha_2$  entre deux faces recollées le long d'une arête. La (figure 2.24) montre la jonction de deux volumes par des bijections  $\alpha_3$  le long des faces de recollement.

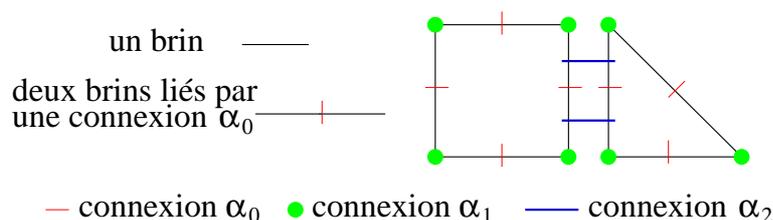


FIG. 2.23 – Détails des bijections  $\alpha_0, \alpha_1, \alpha_2$  de deux faces recollées le long d'une arête

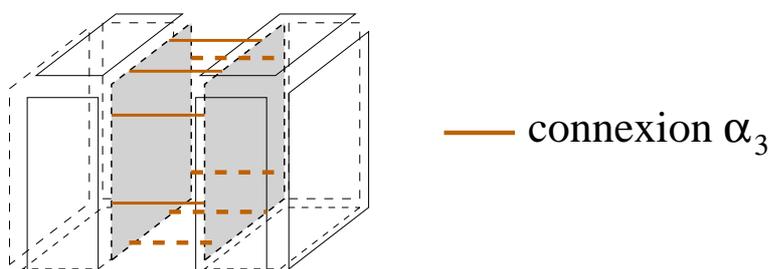


FIG. 2.24 – Détails des bijections  $\alpha_3$  permettant de recoller deux volumes le long d'une face

Nous allons maintenant introduire quelques notions supplémentaires sur les cartes. Nous utiliserons ces notions pour définir simplement les outils topologiques de bas et de haut niveau que nous développerons dans les *section 2.3* et *section 2.4*.

Comme nous venons de le voir, une face peut être définie à partir des arêtes qui la constituent. Chaque arête est constituée de deux brins liés par une liaison  $\alpha_0$ . Pour définir une face à partir d'un brin donné, il suffit donc de rechercher tous les brins liés par des liaisons  $\alpha_0, \alpha_1$  au brin de départ. Cette approche permet de définir intuitivement la notion d'orbite.

**Définition 2.3** *L'orbite d'un brin  $b$ , relativement à un ensemble d'applications  $\{\pi_0, \pi_1, \dots, \pi_n\}$  est l'ensemble de tous les brins accessibles en appliquant toute composition de ces applications. Cette orbite est notée  $\langle \pi_0, \pi_1, \dots, \pi_n \rangle(b)$ .*

Cette notion d'orbite permet de définir la notion de cellule directement.

**Définition 2.4** Soit  $G = (\mathcal{B}, \alpha_0, \alpha_1, \alpha_2, \alpha_3)$  une 3G-carte, un brin  $b \in \mathcal{B}$ , et  $i$  un entier,  $i \leq 3$ , la  $i$ -cellule incidente à  $b$  est l'orbite  $\langle \alpha_0, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_3 \rangle(b)$ .

Par exemple :

- la 0-cellule (le sommet) incidente à un brin  $b$  est définie par l'orbite  $\langle \alpha_1, \alpha_2, \alpha_3 \rangle(b)$  ;
- la 1-cellule (l'arête) incidente à un brin  $b$  par l'orbite  $\langle \alpha_0, \alpha_2, \alpha_3 \rangle(b)$  ;
- la 2-cellule (la face) incidente à un brin  $b$  par l'orbite  $\langle \alpha_0, \alpha_1, \alpha_3 \rangle(b)$  ;
- la 3-cellule (le volume) incidente à un brin  $b$  par l'orbite  $\langle \alpha_0, \alpha_1, \alpha_2 \rangle(b)$ .

Une subdivision est définie comme un ensemble de cellules de différentes dimensions (sommet, arête, face, volume), auxquelles sont associées des relations de bord. Ainsi, les G-cartes modélisent la topologie des subdivisions en utilisant un seul type d'élément de base (les brins) sur lesquels des applications sont définies.

Pour finir, nous introduisons la notion de degré de liberté. Cette notion est très importante car elle permet de caractériser formellement les relations de voisinage entre les différents objets topologiques que nous manipulerons. Dans notre modèle, nous utiliserons essentiellement les libertés de degré 3 (entre face de différents volumes).

**Définition 2.5** Soit  $b$  un brin d'une carte  $g$  et  $i$ , ( $i \in \mathbb{N}, i \geq 0$ ), un brin  $b$  est dit libre par une bijection  $\alpha_i$  si  $\alpha_i(b) = b$ .

Si tous les brins de la carte sont liés, alors la carte est dite fermée (sans bords) sinon elle est dite ouverte (à bords). A noter que les G-cartes des n-cellules sont sans bord.

Par extension, nous dirons qu'une face (topologique) est libre (par  $\alpha_3$ ) si tous les brins qui la constituent sont libres par  $\alpha_3$ .

Dans toute la suite de ce mémoire, les objets manipulés seront dits non dégénérés, c'est à dire qu'ils vérifieront la définition des cartes généralisées de dimensions 3. Par exemple, il n'y aura pas de face non refermée ou d'arête pendante, ou encore de face partiellement recollée... Cette restriction permet de simplifier la description des opérations topologiques développées.

Toutes les autres notions relatives aux subdivisions (composante connexe, orientation...) sont définies sur le même principe, en utilisant la seule notion d'orbite (voir (Bertrand *et al.*, 1993b) et (Lienhardt, 1994)) pour plus de détails). L'intérêt principal de ce concept est qu'il est très simple de recalculer les éléments durant l'évolution. Ainsi, la définition de nouvelles subdivisions et les opérations les manipulant sont simples à définir mais aussi à implémenter.

## 2.3 Mise en place du modèle

### 2.3.1 Structuration du modèle

A partir de ces principes, et comme dans (Terraz, 1994), nous utilisons une 3G-carte augmentée dans laquelle les subdivisions de bas niveau (cellule, bord...) sont définies en utilisant la notion d'orbite introduite précédemment. Pour définir les subdivisions de

haut niveau (axes, cernes. . .), nous associons à chaque volume des attributs spécifiques correspondant à l'ordre de leur axe de croissance (*cf. section 1.3.2*) et à leur étage dans l'axe de croissance (ce qui permet de différencier les cernes successifs). Nous proposons également une application appelée *filiation*, notée  $\alpha_f$ , qui permet de contrôler la structuration en axe de croissance. Ces simples ajouts nous permettent de définir les axes de croissance primaires et radiaux, les cernes et un arbre de filiation.

Pour décrire un objet subdivisé en fonction de différents critères (ici la structure anatomique), nous partirons de la subdivision la plus fine (le brin), puis nous structurerons cette subdivision en sous-objets de plus haut niveau jusqu'à obtenir notre objet de plus haut niveau : l'arbre. Cette idée n'est pas nouvelle, et a déjà été étudiée dans plusieurs méthodes de simulation de métamorphoses. Par exemple pour la méthode de l'AMAP, un arbre naturel est représenté par un arbre (au sens de la théorie des graphes), où une arête de cet arbre correspond à un entre-nœud de l'arbre naturel. Puis cet arbre est décomposé en unités de croissance, en axes, . . .

Nous réutilisons donc ici ce mécanisme général permettant de définir, pour une 3Gcarte, une structuration de celle-ci en sous-objets de plus haut niveau que les brins et les cellules. Il s'agit en fait d'introduire des mécanismes qui permettent de définir de nouvelles relations entre ces sous-objets, et d'associer des attributs et des comportements à ces sous-objets.

Bien que notre approche se base sur celle de (Terraz, 1994), pour des raisons de simplification de la description des objets, nous n'utiliserons pas le même formalisme pour définir les sous-objets. Ce formalisme ne sera utilisé que pour la relation de filiation, pour les autres relations nous utiliserons les propriétés des objets.

Dans la description classique des objets, toutes les notions sont rattachées aux brins. Ici, nous avons choisi d'associer à chaque objet que nous définissons, des notions propres. La plupart des notions appliquées aux cartes (orbite, cellule, bords) peuvent être retrouvées en descendant simplement les propriétés des objets de haut niveau aux sous-objets de plus bas niveau : les brins. Bien sûr, celles-ci pourraient être définies directement sur les brins, mais cela nuirait à une bonne compréhension du modèle et alourdirait considérablement la notation et l'occupation mémoire.

Ainsi, nous définirons un objet comme un ensemble de sous-objets qui vérifient certaines conditions. Ces conditions pourront s'appliquer aussi bien sur le brin lui-même que sur tout autre sous-objet de plus haut niveau auquel il appartient.

Les principaux sous-objets que nous manipulerons tout au long de ce mémoire sont les suivants (certains de ces sous-objets ne sont donnés qu'à titre indicatif, car nous ne les exploiterons pas directement) :

- le brin,
- l'arête «simple» : c'est l'union de deux brins connectés par une liaison  $\alpha_0$ ,
- la face «simple» : c'est un cycle fermé d'arêtes réunies par des liaisons  $\alpha_1$ ,
- le volume : c'est un cycle fermé de faces réunies par des liaisons  $\alpha_2$ ,
- l'axe de croissance : c'est un ensemble de volumes qui se développent dans une direction donnée,
- le cerne : nous distinguons le cerne de l'arbre, du tronc, de la branche et de chaque

- unité de croissance,
- l'unité de croissance : elle correspond à la portion de tige qui apparaît au bout d'une année,
- l'unité de branche : elle correspond à la portion de tige qui se développe lors de l'apparition d'une branche (la justification de cette distinction sera faite ultérieurement).

### Structuration de bas niveau

Dans notre modèle, l'unité de base est le brin. Mais, pour simplifier les explications, nous considérerons que nous ne manipulerons que des volumes (le lien avec les brins peut être retrouvé dans (Terraz, 1994)). Une carte est donc composée de volumes reliés entre eux par des liaisons  $\alpha_3$ .

Chaque volume peut :

- partager des informations : propriétés communes à des familles d'objets similaires,
- avoir des propriétés qui lui sont propres.

Cette distinction est faite par l'utilisateur lors de la création du volume, mais elle peut être modifiée à tout moment. Par exemple, elle permet de pouvoir contrôler la forme des volumes à différents niveaux. En effet, l'utilisateur peut agir sur un volume unique ou sur une famille de volumes<sup>7</sup>.

Chaque volume comprend :

- des propriétés pour calculer son repère relatif (3 angles, un facteur de translation et un sens d'applications de 3 rotations induites par les 3 angles, car l'application n'est pas bijective),
- un repère relatif qui se déduit des propriétés du volume, et le cas échéant du repère relatif de la face support pour le nouveau volume,
- des dimensions : longueur, largeur, hauteur<sup>8</sup>,
- un point central, centre du repère relatif,
- un ordre pour l'arbre de filiation (axes de croissance),
- un étage dans l'arbre de filiation (cela sert en particulier pour pouvoir sélectionner des familles de volumes comme les cernes),
- une date de création (dans le cas des arbres, demi-année où le volume apparaît),
- une saison de création (spécifique aux arbres),
- un sens d'application de textures (nous le détaillerons plus particulièrement dans la partie sur la génération de textures),
- un type de matière (choix de la couleur de fond du volume),
- un marqueur de forme (ce marqueur est utilisé pour connaître la forme générale du volume : cube, cylindre, volume dégénéré. Il est surtout utilisé pour le calcul des coordonnées textures),
- une boîte englobante (qui sert essentiellement à accélérer les calculs),
- des marqueurs pour les sauvegardes et les parcours,

<sup>7</sup>On entend ici par notion de famille toute réunion non nulle de volume.

<sup>8</sup>Si les dimensions sont communes à plusieurs volumes, alors les variables pointeront vers une zone mémoire commune.



- un ou plusieurs marqueurs de fente (nous verrons leur utilité dans la partie sur les fissures),
- un marqueur de face subdivisée ou dégénérée (nous verrons son utilité plus loin),
- une matière (si elle est définie, elle supprime celle du volume),
- des méthodes pour accéder aux sous-objets : liste des arêtes,...
- une liste de marqueurs : prise en compte du voisinage ou non pour le plongement,...
- nombre d'arêtes,
- ...

Nous avons choisi de ne pas posséder d'entité arête, car cela ne nous a pas semblé pertinent. Techniquement, nous n'utilisons pas directement d'informations spécifiques sur les arêtes.

Chaque brin est composé :

- de méthodes permettant d'accéder aux sur-objets,
- de méthodes pour le calcul du plongement,
- d'une liste de marqueurs.

Il convient de remarquer que, certaines données, peuvent être redondantes, mais ce choix est volontaire, car il permet de simplifier grandement les accès aux propriétés des objets en particulier. Ainsi, nous stockons les repères relatifs, alors qu'ils pourraient tout à fait être calculés à la volée au moment de leur utilisation. Cependant, comme nous l'avons signalé précédemment, chaque propriété de chaque objet peut être gérée manuellement ou automatiquement.

## Structuration de haut niveau

Pour définir les subdivisions de haut niveau (axe, cerne, ...), nous avons besoin d'informations supplémentaires.

Comme dans (Terraz, 1994), nous avons gardé la relation de filiation que nous notons  $\alpha_f$ . Cette application permet de contrôler la structuration en axes de croissance. Cette relation permet de regrouper les volumes sous forme d'un arbre ou graphe (au sens de la théorie des graphes). Ce dernier est ensuite utilisé pour déterminer l'ordre des volumes pour les calculs de plongement.

Comme décrit par les botanistes ( *sous-section 1.3.2* ), le modèle est décomposé en axes de croissance ( *figure 1.11* ). Il existe deux types d'axes de croissance : les axes de croissance primaires qui assurent la croissance en longueur, et les axes de croissances secondaires qui assurent la croissance en épaisseur. Un axe est donc un objet qui permet surtout de définir des opérations sur des familles particulières de volumes. Formellement, nous définissons un axe de croissance d'ordre  $n$  incident à un brin  $b$  par l'orbite  $\langle \alpha_0, \alpha_1, \alpha_2, \alpha'_f \rangle(b)$  où  $\alpha'_f$  est la restriction de  $\alpha_f$  aux brins adjacents aux volumes d'ordre  $n$ . Par exemple, il existe des méthodes permettant d'obtenir toutes les faces côtés libres des volumes d'un axe  $i$ . Ainsi, nous associons à chaque volume des attributs spécifiques correspondant à l'ordre de leur axe de croissance ( *Cf. section 1.3.2* ) et à leur étage dans l'axe de croissance (ce qui permet de différencier les cernes successifs).

De façon analogue, nous introduisons la notion de cerne à différents niveaux : cerne

d'arbre, cerne de branche, ou cerne d'unité de croissance <sup>10</sup>. Un cerne d'âge  $i$  d'ordre  $n$  (l'ordre ici sert à distinguer un cerne d'une branche d'un cerne du tronc) est défini comme l'ensemble des brins incidents à des volumes d'étage  $i$  et d'ordre  $n$ .

Ces simples ajouts nous permettent de définir les axes de croissance primaires et radiaux, les cernes et un arbre de filiation.

### 2.3.2 Opérations topologiques de base

De nombreuses opérations de bas et de haut niveau ont été définies pour manipuler les G-cartes (c.f. (Dufourd, 1991), (Bertrand *et al.*, 1993a), (Lienhardt, 1994)). Dans notre modèle, seulement quatre opérations topologiques simples sont utilisées<sup>11</sup> :

- **créer un volume cylindrique**<sup>12</sup>,
- **éclater une arête**,
- **identifier deux faces**,
- **éclater une face**,
- **éclater un volume**.

Par souci de clarté, nous ne développerons pas tous les détails techniques liés à la création de ces opérations. Ces dernières sont, pour la plupart, déjà étudiées de manière approfondie dans (Terraz, 1994). Nous verrons également dans la *sous-section 4.4.1* qu'une opération similaire à l'éclatement a due être développée : la subdivision.

#### 2.3.2.1 Création d'un volume cylindrique

Cette opération peut prendre jusqu'à neuf paramètres dont :

- le degré des faces origines et extrémités,
- un pointeur sur la face génératrice,
- l'ordre du volume,
- l'étage du volume dans son axe,
- un facteur de translation pour le centre du volume,
- la hauteur du volume,
- la longueur du volume,
- la largeur du volume,
- un pointeur vers le volume père de l'axe.

Nous tenons à rappeler que l'utilisateur peut intervenir sur de nombreuses autres propriétés du nouveau volume (cette intervention peut être faite n'importe quand). Cette opération n'agit que sur la topologie du volume. Ces paramètres ne seront pris en compte d'un point de vue géométrique que lors du calcul du plongement du volume.

Cette opération permet de créer un volume de type prisme régulier de degré  $k$  ( $k \in \mathbb{N}, k > 1$ ). Elle est similaire à l'extrusion d'une face de degré  $k$  (le degré d'une face est déterminé par le nombre de ses arêtes).

<sup>10</sup>La notion d'unité de croissance sera détaillée dans la section suivante.

<sup>11</sup>Nous verrons par la suite que d'autres opérations peuvent en être déduites.

<sup>12</sup>Nous avons défini d'autres types de volumes, mais ils ne sont pas utilisés dans cette partie.

Tout d'abord, nous commençons par créer deux faces de même degré qui serviront de support au volume. Nous appelons ces faces : face *Origine* (notée O) et face *Extrémité* (notée E). Cette notation permet essentiellement de mieux se repérer dans le volume. Puis nous accolons sur chacune des arêtes de chaque face support, des faces de degré 4 que nous appelons faces *Côté* et que nous notons C (*figure 2.26*).

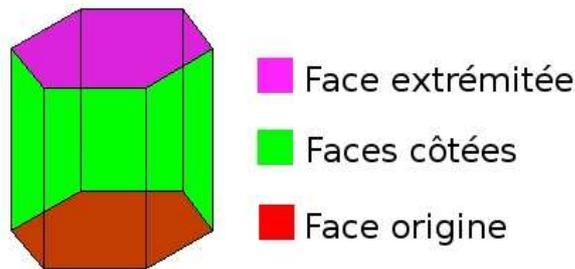


FIG. 2.26 – Création d'un volume cylindrique de degré 6

### 2.3.2.2 Éclatement d'une arête

Cette opération consiste à créer une nouvelle arête à partir de la précédente. Intuitivement l'arête est coupée en deux et deux demi-arêtes sont insérées au milieu (voir le sommet du cube, (*figure 2.28*) a et (*figure 2.28*) b).

D'un point de vue technique, cette opération se déroule en 4 étapes :

- étape 1 : deux nouveaux brins sont créés et liés par une liaison  $\alpha_1$  ;
- étape 2 : les liaisons  $\alpha_0$  des deux brins constituant l'arête de départ sont modifiées pour qu'elles pointent vers les deux nouveaux brins (le voisinage est pris en compte à l'étape suivante) ;
- étape 3 : prise en compte du voisinage : si l'arête éclatée possède un voisinage non nul (les liaisons  $\alpha_2$  et  $\alpha_3$ , ne renvoient pas le brin support), il faut alors éclater toutes les arêtes voisines. Nous remontons donc à l'étape 1 avec chaque brin support de chaque arête voisine ;
- étape 4 : tout le voisinage a été éclaté, il faut donc maintenant recréer si nécessaire, les relations de voisinage des nouveaux brins. Pour cela, il suffit de suivre les informations de voisinages fournies par le brin support (techniquement, l'orbite  $\langle \alpha_2, \alpha_3 \rangle$  du brin support et des brins nouvellement créés est parcourue en parallèle).

### 2.3.2.3 Identification de deux faces

Premièrement, cette opération contrôle le degré des faces qui doivent être identifiées (*figure 2.27*) . Si les faces ne sont pas isomorphes (si leurs degrés respectifs ne sont pas égaux), les bords des faces sont modifiés en éclatant successivement leurs arêtes. Une fois que les faces sont isomorphes, elles sont identifiées par une simple modification des bijections  $\alpha_3$  associées aux brins. Cette opération est utilisée pour identifier les faces

de contact entre deux volumes. Par exemple, cela permet d'identifier la face génératrice d'une branche, qui est une face latérale d'un cylindre (axe d'ordre 1) et donc de degré 4, à la face Origine du premier élément de la branche (axe d'ordre 2) qui est de degré 32 ( *sous-section 2.4.2* ).

Lorsque les deux faces n'ont pas le même degré, la répartition des éclatements des arêtes se fait en fonction du type des arêtes de la face d'ordre supérieur.

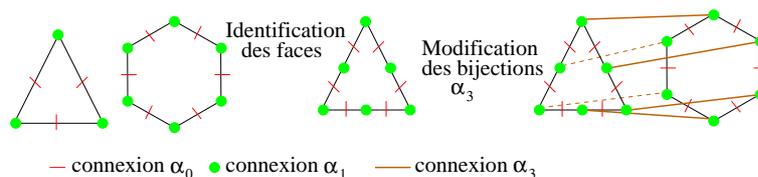


FIG. 2.27 – Identification de deux faces non isomorphes

### 2.3.2.4 Éclatement d'une face

Cette opération prend en paramètre trois brins : deux brins sur les arêtes à éclater et le brin de départ d'un chemin. Puis elle se déroule en quatre étapes :

- étape 1 : il faut vérifier que la face à éclater est libre (par  $\alpha_3$ )<sup>13</sup> ;
- étape 2 : les deux arêtes support (arêtes contenant les brins supports) sont éclatées ;
- étape 3 : le chemin est dédoublé (création d'un deuxième chemin topologique et modification des liaisons  $\alpha_2$ ) ;
- étape 4 : les bijections  $\alpha_1$  entre les deux extrémités des chemins de la phase 2 et les nouvelles liaisons  $\alpha_1$  créées lors de l'éclatement des arêtes de la phase 1 sont identifiées.

Pour notre modèle, nous utilisons un chemin minimaliste (i.e. restreint à une seule arête) et la face est partagée en deux parties égales (donc il suffit de donner une arête en paramètre, l'arête opposée est obtenue facilement) :

- étape 1 : il faut vérifier que la face à éclater est libre (par  $\alpha_3$ ) ;
- étape 2 : les deux arêtes opposées et incidentes à une même face sont éclatées ( *figure 2.28* a et *figure 2.28* b) ;
- étape 3 : deux nouvelles arêtes, reliées par des liaisons  $\alpha_2$ , sont créées ( *figure 2.28* c) ;
- étape 4 : les bijections  $\alpha_1$  entre les deux nouvelles arêtes de la phase 2 et les nouvelles liaisons  $\alpha_1$  créées lors de l'éclatement des arêtes de la phase 1 sont identifiées ( *figure 2.28* d).

<sup>13</sup>Nous verrons dans la partie sur le vieillissement que ce modèle a été aussi implémenté pour les faces non libres.

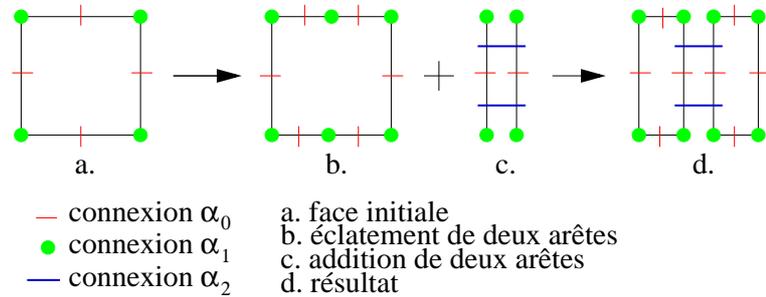


FIG. 2.28 – Éclatement d'une face

### 2.3.2.5 Éclatement d'un volume

Cette opération est une généralisation de l'éclatement de face à la dimension 3. Elle consiste en un éclatement des faces d'un volume ( *figure 2.29* b et *figure 2.29* c), un ajout de deux nouvelles faces générées par les nouvelles arêtes et reliées par des bijections  $\alpha_3$  ( *figure 2.29* d) et une modification des bijections  $\alpha_2$  afin de recoller les nouvelles faces à leur volume respectif, le tout ayant comme résultat deux volumes. Cette opération est utilisée pour simuler la croissance selon les axes radiaux.

Nous avons volontairement choisi de ne détailler que le modèle développé afin d'alléger les explications.

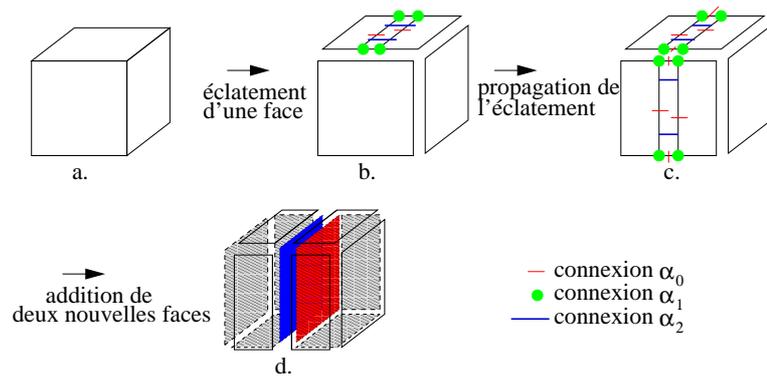


FIG. 2.29 – Éclatement d'un volume

Ces quelques opérations de bas niveau vont nous permettre de définir simplement des opérations de plus haut niveau pour décrire la croissance des arbres. Mais auparavant, nous allons présenter les différents types de plongements que nous avons développés. Cette technique permet d'associer une géométrie à la topologie des objets que nous venons de créer.

### 2.3.3 Modèles de plongements

Pour visualiser la topologie de notre modèle, nous avons développé plusieurs types de plongements. Nous commencerons par détailler le modèle utilisé par défaut, puis nous verrons comment nous résolvons certains problèmes liés au plongement par défaut en utilisant de nouveaux types de plongements.

#### 2.3.3.1 Plongement géométrique par défaut

Dans notre application, les 3G-cartes sont plongées linéairement dans  $R^3$ , par association d'un point de  $R^3$  à chaque sommet (0-cellule, cf. *définition section 2.2*). Le plongement des arêtes, faces et volumes est déduit du plongement des sommets d'une manière classique (une arête est plongée comme un segment de droite dont les extrémités sont les points associés aux sommets incidents à l'arête...).

Pour visualiser notre modèle, nous utilisons un mécanisme de plongement hiérarchique, appliqué à chaque étape de l'évolution :

- calcul du système de référence,
- plongement par défaut du volume,
- calcul des coordonnées finales des sommets.

#### 1 Système de référence

Comme pour les systèmes de particules (Reeves, 1983) et pour un bon nombre d'autres méthodes ((Lienhardt, 1988), (Prusinkiewicz *et al.*, 2001)...), chaque volume a son propre système de référence. Dans notre étude, ce système se calcule à partir de celui de la face support (ou face mère)<sup>14</sup> sur le volume père et des propriétés géométriques du nouveau volume. Si le nouveau volume n'a pas de support, nous utilisons le repère absolu comme support pour le nouveau volume. Le système de référence par défaut est calculé à partir du système de référence de la face support sur le volume père et des dimensions théoriques du nouveau volume. Ce repère est ensuite modifié en fonction des attributs de position du nouveau volume (3 angles et un facteur de translation). Le système de référence  $(O, \vec{i}, \vec{j}, \vec{k})$  est calculé à partir de ces attributs associés, en appliquant les trois rotations et la translation au système de référence  $(O', \vec{i}', \vec{j}', \vec{k}')$  associé à la face support du volume **père** dans l'arbre de filiation. Le calcul de ce système de référence est obtenu en utilisant la relation de filiation. Le système de référence peut, bien sûr, ne pas être automatiquement calculé et défini entièrement par l'utilisateur.

#### 2 Forme du volume : plongement par défaut

A partir du système de référence calculé précédemment, un volume par défaut est calculé. Pour calculer le plongement du volume par défaut, nous ne tenons pas compte des relations de voisinage autres que celles intrinsèques au volume  $(\alpha_0, \alpha_1, \alpha_2)$ . La forme du volume par défaut est calculée en fonction des propriétés géométriques associées à chaque face lors de la construction du volume ( *sous-section 2.3.2* ). Ces propriétés

---

<sup>14</sup>Il ne faut pas confondre la face support pour la construction d'un nouveau volume et la face origine du nouveau volume.

géométriques dépendent du type et des dimensions du volume modélisé, mais également du nombre de faces qui constituent le volume. Pour calculer le point représentatif<sup>15</sup> de chaque face du volume à plonger, nous effectuons les trois rotations associées à la face  $(\alpha, \beta, \gamma)$  suivi du facteur de translation associé au repère du volume (*figure 2.30*).

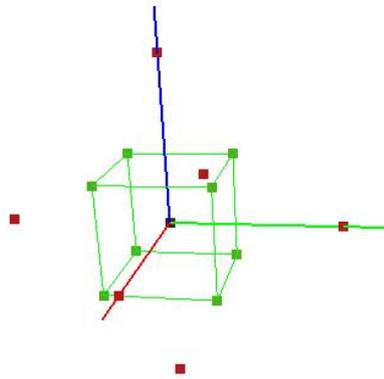


FIG. 2.30 – Les principaux éléments du plongement : repère du volume, point représentatif des faces (marron) et sommets (verts)

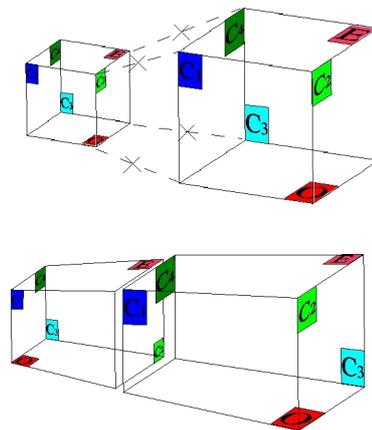


FIG. 2.31 – Calcul final des coordonnées des sommets en tenant compte des relations de voisinage

<sup>15</sup>Il ne faut pas confondre le centre d'une face (barycentre des sommets qui la composent) et le point représentatif qui sert au calcul des sommets.

### 3 Calcul des coordonnées finales des sommets

La position finale d'un sommet est ensuite calculée comme le barycentre des points représentatifs des faces incidentes à celui-ci (*figure 2.31*). Pour cela, nous récupérons le brin représentatif du sommet, puis nous parcourons tous les brins qui constituent la 0-cellule associée<sup>16</sup>. Cette approche nous permet de récupérer toutes les faces incidentes au sommet. Nous utilisons les marqueurs vus dans la *sous-section 2.3.1* pour éviter de traiter plusieurs fois la même face.

#### 2.3.3.2 Plongement manuel

C'est le mode de plongement le plus simple. Dans ce modèle, les coordonnées des sommets ne sont pas calculées en fonction des propriétés des objets, mais fournies directement par l'opérateur au moment de la construction des volumes. Plus particulièrement, lorsqu'un cube est construit, il faut fournir explicitement les coordonnées des sommets qui vont le représenter géométriquement. Bien sûr, toutes les propriétés associées au volume (attributs géométriques, ...) n'ont plus aucun sens, mais nous les gardons en mémoire (lorsqu'elles sont définies) pour des raisons de souplesse au niveau de l'affichage.

Par exemple lorsqu'une grume de bois est découpée, il est particulièrement intéressant de garder sa structure (axe de croissance, cerne, ...) car elle peut encore apporter des informations au niveau de l'orientation des fibres du bois sur les objets ainsi générés.

#### 2.3.3.3 Plongement géométrique hiérarchique pour les subdivisions

Comme nous l'avons vu dans la partie sur les opérations topologiques de base, nous introduisons un plongement particulier pour définir les subdivisions de faces. Ce modèle particulier de plongement est complémentaire du modèle par défaut. En effet, pour définir les coordonnées des sommets subdivisés, nous avons besoin de connaître les coordonnées des sommets support. Ces sommets support peuvent eux-mêmes être des sommets subdivisés. Le plongement se déroule en deux phases en fonction du type de sommet à traiter :

- le sommet à traiter n'est pas marqué comme faisant partie d'une subdivision, nous le traitons alors avec la méthode de plongement par défaut ;
- le sommet est marqué comme faisant parti d'une subdivision, dans ce cas là, nous devons vérifier que les sommets qui serviront au calcul sont déjà évalués, si ce n'est le cas, nous les calculons. Les coordonnées du sommet sont simplement obtenues en calculant le barycentre des sommets qui participent, affecté de leur poids correspondant. Pour récupérer les sommets qui participent nous utilisons des relations d'appartenance. Celles-ci ne s'expriment pas sur le sommet, mais sur le brin représentatif du sommet. En effet si ces relations étaient stockées sur le sommet directement, lors d'un décollement, nous aurions quelques difficultés à déterminer quel sommet intervient avec quel autre. Techniquement, à chaque

---

<sup>16</sup>Nous verrons plus loin qu'une autre condition peut être introduite comme lors de la création de fentes.

brin représentatif d'un sommet de subdivision est affecté une liste de brins avec un poids. Il suffit ensuite de calculer la moyenne pondérée des sommets attachés pour obtenir le sommet final.

Il convient de noter qu'il serait intéressant de considérer un plongement non linéaire (en associant des courbes aux arêtes et donc des surfaces aux faces...). L'utilisation d'un plongement non linéaire, devrait nous permettre de diminuer le nombre de volumes utilisés, mais il en résulterait une perte de précision dans le contrôle, en particulier dans la modélisation des phénomènes spécifiques comme l'éclatement du bois. De plus, en utilisant un plongement non linéaire le modèle d'application de texture devrait être entièrement revu (nous pourrions toujours calculer la coordonnée texture en bordure de volume, mais cela deviendrait plus compliqué pour la calculer dans d'un volume).

## 2.4 Opérations topologiques de plus haut niveau

Le contrôle de l'évolution topologique est réalisé en définissant des opérations applicables aux sous-objets de haut niveau (axes, cernes...). Ces opérations de haut niveau sont définies grâce à des compositions d'opérations de base.

Le modèle de croissance que nous allons détailler dans cette partie ainsi que certaines des opérations de haut niveau qui sont nécessaires pour simplifier l'expression de la croissance, ont fait l'objet d'une publication (Guimberteau *et al.*, 2006).

Nous avons choisi de décomposer les différents phénomènes qui interviennent lors de la croissance du bois en plusieurs opérations (ces justifications se retrouvent en grande partie dans la décomposition de la croissance qu'en font les botanistes) :

- **les unités de croissance en longueur (ou UCL)** : Elles correspondent à la portion de bois qui se développe en une année à partir d'un bourgeon terminal. Ces unités ne gèrent que partiellement l'insertion des branches (le support de la nouvelle branche est marqué, mais c'est une autre opération qui modélise la branche).
- **les unités de branche (UB)** : Elles servent à modéliser la naissance de la branche sur le pourtour du tronc (bourgeons latéraux). Dans notre modèle, elles apparaissent en même temps que les unités de croissance, mais nous pouvons également les faire apparaître plus tard (bourgeons dormant). Dans ce cas là, le support est marqué lors de l'opération de croissance en épaisseur, puis il sera utilisé quand le bourgeon se développera.
- **les unités de joints, ou unités de transition (UT)** : Pour simplifier certaines opérations, nous avons choisi de distinguer les phénomènes qui se passent au niveau des bourgeons terminaux du reste du processus de croissance. D'un point de vue pratique, ces unités assurent un raccordement cohérent entre les différentes unités de croissance.
- **l'opération de croissance en épaisseur (OCE)** : Elle sert à modéliser la croissance en épaisseur de l'arbre. Nous n'utilisons pas le terme Unité de Croissance en Epaisseur (UCE) car les nouveaux volumes créés lors de cette opération sont obtenus pas éclatement des volumes des couches inférieures.

L'avantage indéniable d'une telle subdivision est de permettre de hiérarchiser le modèle de croissance. Ainsi, nous décomposons le modèle en opérations plus élémentaires. Cependant, elles restent tout de même assez complexes à mettre en place et sont en partie interdépendantes.

Notre croissance est donc modélisée à partir des quatre opérations définies ci-dessus.

### 2.4.1 Unité de croissance en longueur (UCL)

Dans notre modèle, pour simuler la croissance en longueur de l'arbre, nous rajoutons des groupes d'éléments aux extrémités des axes de croissance primaires (support des bourgeons terminaux). Nous appelons ces éléments des unités de croissance en longueur (UCL). Intuitivement, à chaque étape une nouvelle unité de croissance (UCL) est ajoutée au sommet de chaque axe de croissance. Tous ces nouveaux volumes sont naturellement joints à leurs voisins par des bijections  $\alpha_3$  appropriées. Plus précisément, la croissance en longueur du tronc et des branches est réalisée en trois étapes :

- l'obtention de toutes les faces extrémités libres de tous les axes principaux (ordre 1,2...);
- l'ajout d'une UJ, puis d'une UCL sur ces faces (une UCL pouvant aussi bien s'appliquer sur le tronc que les branches) ;
- la modification des bijections  $\alpha_f$  ;
- le recollement des faces côtés entre elles.

La notion d'unité de croissance telle qu'elle est utilisée ici est quelque peu différente de celle décrite par les botanistes. En effet, pour les botanistes, une unité de croissance comprend la portion de tige qui peut apparaître en une année, branches comprises. Dans notre modèle une UCL représente la portion de tige qui peut apparaître au cours d'une année avec une gestion partielle des branches. En effet cette unité ne gère que le support de la future branche. Ce support est dit dormant si le bourgeon ne se développe pas tout de suite (par analogie avec les bourgeons dormants). Le reste de la gestion de la création de la branche (création de la matrice) est assuré par l'unité de branche (UB) que nous détaillerons un peu plus loin.

Une UCL, sans la gestion partielle du support des branches, peut se décrire simplement comme une succession de cylindres sur lesquels des cubes <sup>17</sup> sont accolés.

Une UCL se décompose en deux types de volumes (*figure 2.32*) :

- un empilement fini de volumes cylindriques de degré  $k$  ( $k=32$  dans notre modèle) qui constituent le squelette principal de l'arbre. Ces volumes sont dit d'ordre entier (1 quand ils constituent le squelette du tronc, 2 pour le squelette des branches primaires, 3 pour le squelette des branches secondaires,...)
- un amoncellement de volumes cubiques (volumes cylindriques de degré 4) apposé sur chacune des faces côté des cylindres (sauf si un bourgeon apparaît et se développe, dans ce cas c'est un cylindre). Ces volumes modélisent la base des axes de croissance secondaires, ils sont d'ordre dit flottant  $X.Y$  où  $X$  représente l'ordre du volume cylindrique à la base du cube et  $Y$  représente le numéro de la face à

<sup>17</sup>La notion de cube est utilisée au sens topologique du terme.

l'origine du cube<sup>18</sup>.

La base est donc un empilement de cylindres dont les dimensions peuvent varier. Ces volumes cylindriques se décomposent en deux sous-catégories (*figure 2.32*) :

- les cylindres hors de la zone de branchaison (ceux au dessus et en dessous)
- les cylindres de la zone de branchaison

Cette décomposition a été choisie afin de pouvoir augmenter le nombre d'outils de contrôle pour la forme des unités de croissance en longueur (bois de torsion, courbure de l'arbre, ...), mais également la position et l'angle d'insertion des branches.

Les cylindres au dessus et en dessous de la zone de branchaison sont traités de manière analogue. Nous rajoutons simplement des cubes sur les faces côté de chaque cylindre situé au dessus ou en dessous de la zone de branchaison. Pour cela, nous parcourons l'orbite  $(\alpha_0, \alpha_1)$  de la face origine de chacun de ces cylindres, et nous prenons à chaque étape la face adjacente au brin courant (liaison  $\alpha_2$ ) qui sert de support au nouveau cube. Le parcours ainsi défini, nous assure que les faces seront toujours traitées dans le même sens et de proche en proche.

Lors du traitement de chaque face, une fonction qui gère la probabilité d'apparition d'une nouvelle branche est évaluée. Cette fonction est composée<sup>19</sup> :

- d'une fonction aléatoire qui représente les caractéristiques de l'espèce modélisée (taux d'apparition, ...);
- d'une fonction de correction topologique qui assure que le voisinage proche ne pose pas de problème (branche trop proche par exemple);
- d'une fonction de correction géométrique qui permet d'évaluer la cohérence au niveau de la répartition des branches sur le tronc (phyllotaxie primaire). Cette fonction n'est exécutée qu'une fois la moitié des faces parcourues et ne prend en compte que les faces déjà traitées et marquées.

Si la fonction retourne un résultat positif, la face est marquée pour un traitement ultérieur et nous passons à la face suivante. Si la fonction retourne un résultat négatif, aucune branche ne peut apparaître. Un cube est donc ajouté sur la face pour modéliser la portion de cerne qui se développe et nous passons à la face suivante.

Pour déterminer les dimensions du nouveau cube, nous nous basons sur :

- des probabilités statistiques liées aux propriétés géométriques intrinsèques à l'espèce que nous modélisons (hauteur moyenne du cerne, variation, ...);
- l'étude du voisinage (comme les recollements ne sont pas encore effectués à ce stade, nous utilisons le voisinage de la face support pour connaître les volumes voisins) :
  - la prise en compte des volumes de la saison en cours de traitement permet d'éviter les variations trop importantes de l'épaisseur du cerne, d'un volume à

---

<sup>18</sup>Nous utilisons donc le numéro de face pour distinguer les axes de croissance secondaires. Par exemple, pour un volume cylindrique d'ordre 1 et de degré 16, la première couronne de cube sera 1.1, 1.2, ..., 1.16. Ce système de notation est également utilisé pour l'affichage comme nous le verrons plus loin.

<sup>19</sup>La liste donnée ici n'est que succincte, nous pourrions aussi tenir compte de l'environnement externe (aspect climatique, concurrentiel entre les espèces du biotope, ...) mais ce n'est pas le sujet de ce mémoire.

- l'autre ;
- la prise en compte des volumes des saisons précédentes permet de prendre en compte les phénomènes de torsion, de bois de compression, . . . que le bois a déjà subit. C'est ce que nous appelons l'héritage des parents. Il a un impact non négligeable sur le nouveau volume.

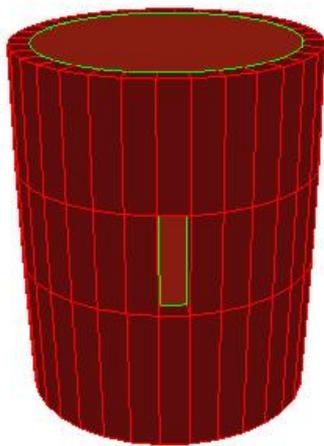


FIG. 2.32 – Une UCL avec l'emplacement pour une branche

Une fois que tous les cubes nécessaires ont été ajoutés, les faces marquées sont traitées. Deux cas de figures se présentent :

- le bourgeon se développe, alors une unité de branche (UB) est ajoutée à l'emplacement marqué ;
- le bourgeon ne se développe pas tout de suite, alors un cube est ajouté comme s'il n'y avait pas de bourgeon et sa face extrémité est marquée pour un traitement lors de la prochaine étape de croissance.

Une fois que tout les cubes nécessaires ont été ajoutés, nous utilisons des fonctions de recollement pour identifier les faces cotés des différents cubes entre elles<sup>20</sup>.

La fonction qui gère les UCL prend en paramètre :

- un nombre moyen de branche,
- une espérance d'apparition de branche (nous ne gérons que le support dans cette fonction),
- une hauteur d'unité de croissance (UC) qui se décompose comme suit :
  - un nombre de cylindre sous la zone de branche
  - un nombre de cylindres qui peuvent servir pour l'insertion de branches
  - un nombre de cylindre sur la zone de branche
  - une hauteur par cylindre.

<sup>20</sup>Le modèle de labellisation des faces que nous avons développé pour cette partie ayant rapidement montré ses limites, nous avons choisi de ne pas le détailler ici. Cependant, il est englobé dans celui décrit dans la *section 3.2.1*.

Cette fonction renvoie :

- une liste de brins (qui peut être nulle si aucune branche n’apparaît). Chaque brin appartient à une face qui servira de support pour l’insertion d’unité de branche (UB) ;
- un brin indiquant la face Origine du dernier volume cylindrique créé (ce dernier sera utilisé pour déterminer la face support pour la prochaine étape de croissance en longueur).

Les propriétés physiques et géométriques sont bien sûr prises en compte, mais de manière globale. Elles interviennent à plusieurs niveaux (cernes, volume, silhouette de l’arbre, ...). Pour des raisons de commodités, pour une espèce donnée, les longueurs et largeurs des cylindriques sont égales et fixes, nous ne jouons que sur la hauteur et les angles d’insertion de ces derniers. La probabilité d’apparition d’une branche tient compte des branches qui sont déjà apparues sur le même niveau de branchaison. Une fois que tous les supports ont été intégrés, une méthode permet de vérifier que la phyllotaxie locale est cohérente. Si ce n’est pas le cas, les supports de branche incriminés sont alors remplis. Cependant, dans le modèle actuel, la phyllotaxie avec les branches antérieures n’est pas supportée<sup>21</sup>.

Une fois que l’UC est terminée, elle renvoie un ensemble de brins qui sont des supports de branches. Si l’ensemble est non vide, la fonction d’ajout de branche est appelée avec chaque brin.

Nous pouvons remarquer que les emplacements pour les insertions de branche ont la forme de parallélépipèdes rectangles alors que la base de la branche a la forme d’un cylindre. Pour corriger ce défaut, il nous suffit simplement de préciser au modèle lors de la création de la branche que les volumes du tronc n’interviennent pas dans le calcul du plongement avec ceux de la branche ; ce qui induit que la forme définitive dépend de celle de la branche et non du tronc.

## 2.4.2 Unité de branche (UB)

L’unité de branche (ou UB), est une opération qui ne se déroule que si un emplacement a été prévu lors de la phase de développement de l’UCL (nouveau bourgeon latéral ou bourgeon dormant qui se réveille).

Bien que l’UB et l’UCL aient un comportement similaire, nous avons choisi de les distinguer pour les raisons suivantes :

- cette distinction nous permet de gérer plus simplement les différences de comportements physiques (taille des cernes,...) et visuels (texture,...) entre le tronc et la branche<sup>22</sup> ;
- le cadre de nos travaux se situant essentiellement au niveau de la grume de bois, il est donc inutile de proposer un modèle plus complexe, c’est à dire qui gère conve-

<sup>21</sup>Un modèle plus complet pourrait être proposé en se basant sur la géométrie de l’arbre. Nous pourrions proposer un modèle plus complexe pour gérer complètement la phyllotaxie, il suffirait d’associer, par exemple, une masse à chaque centre des volumes et de calculer le point d’équilibre

<sup>22</sup>Il est possible d’ôter cette distinction, mais en augmentant considérablement la difficulté de l’implémentation.

- nablement l'ajout de branches secondaires au niveau des branches primaires<sup>23</sup> ;
- nous avons dû développer des opérations de recollement spécifiques entre les volumes constitutifs de la branche et ceux du tronc.

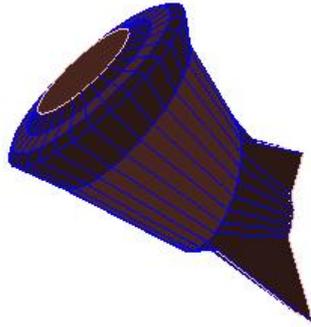


FIG. 2.33 – Une UB de deux ans

Une UB, sans la gestion du raccordement au support, se décrit de manière analogue à une UCL sans la zone de gestion du support des nouvelles branches. Une UB est donc constituée d'un empilement d'un nombre fini de volumes cylindriques (*figure 2.33*). Sur chacune des faces côté des cylindres, nous accolons des volumes cubiques. Ces derniers sont ensuite identifiés entre eux comme décrit au paragraphe précédent. Le nombre de cylindres ajouté est déterminé en fonction du niveau de détail souhaité pour modéliser les courbures de la branche. Naturellement, plus le nombre de cylindre augmente (courbure de branche plus précise), plus la hauteur de chaque cylindre diminue.

La dernière étape consiste à identifier convenablement les faces côtés des cubes constitutifs de la branche et celles de ceux du tronc<sup>24</sup>.

La fonction qui gère les UB prend en paramètre :

- un brin support de la nouvelle branche ;
- un nombre de cylindres ;
- une silhouette générale (caractéristique de l'espèce). Cette silhouette sera conservée pendant tout le processus de croissance, mais pourra être altérée en fonction du développement des autres parties de l'arbre ;
- un ensemble de critères géométriques de placement du nouveau volume de départ de la branche (angles d'insertion, vecteurs de translation, ...).

Comme pour les UCL, d'autres propriétés physiques et géométriques sont bien sûr prises en compte, elles interviennent à plusieurs niveaux (cerne, zone de raccord, ...).

<sup>23</sup>Cependant, quelques essais encourageants ont été réalisés dans ce sens, mais sans la gestion des chevauchements, ou collisions entre les branches.

<sup>24</sup>Cette opération est mieux détaillée dans la *section 3.2.1*.

### 2.4.3 Unité de jointure (UJ)

L'unité de jointure, ou unité de transition (UT) est une opération qui permet d'assurer un raccordement cohérent entre deux UCL successives. En effet, lors de l'étape  $n+1$ , l'UCL issue de l'étape  $n$  contient 3 cernes successifs (été, printemps et été), alors que la nouvelle UCL ne contient qu'un seul cerne (été).

L'ajout d'une (UT) est une opération qui se déroule juste avant l'étape de croissance en longueur des axes (ajout des UCL). Elle permet d'assurer un raccordement cohérent entre l'ancienne unité de croissance et la nouvelle UCL. Cette unité n'a aucun sens, d'un point de vue botanique, mais elle permet également de segmentariser les différentes opérations qui interviennent durant le processus de croissance.

Pour définir cette structure, nous avons besoin d'introduire quelques modifications sur un type de volume. Nous n'utilisons jamais les volumes cylindriques d'ordre 3 sous la forme classique décrite précédemment. Aussi, par abus de langage, nous appellerons volume cylindrique d'ordre 3, un volume décrit comme suit (*figure 2.34*) :

La face de départ est une face d'ordre 4. Puis, nous parcourons le bord de la face, en ajoutant successivement : une face côté d'ordre 3, une face extrémité d'ordre 4, une face côté d'ordre 3 et une face côté d'ordre 4. Nous refermons ensuite le volume ainsi créé. Intuitivement, les volumes ainsi définis peuvent être vus comme des volumes dont les faces ont été relabellisées.

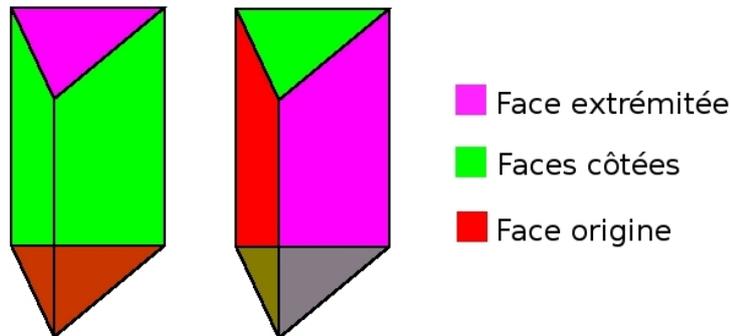


FIG. 2.34 – Labellisation des volumes cylindriques d'ordre 3 : modèle d'origine (à gauche), modèle utilisé (à droite)

De manière plus formelle, une demi-unité de jointure est composée de (*figure 2.35*) :

- un volume cylindrique d'ordre  $k$  ( $k=32$  dans notre modèle usuel),
- de volumes d'ordres 3 modifiés (comme décrit ci-dessus) collés sur chaque face côté du cylindre ;
- d'une opération de recollement entre les faces côtés libres.

Une unité de jointure est composée de (*figure 2.36*) :

- d'une juxtaposition de deux demi-unités de jointures,
- de volumes d'ordre 4 collés aux faces extrémités des volumes d'ordre 3 de la base,
- d'une opération de recollement entre les faces côtés libres des volumes d'ordre 3 et celles des volumes d'ordre 4.

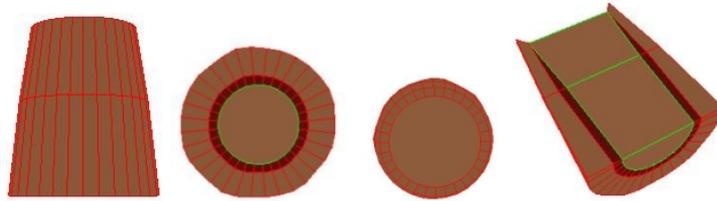


FIG. 2.35 – Demi-unité de jointure sur son support (de la gauche vers la droite) : vue de côté, de dessous, de dessus et coupe de trois-quarts

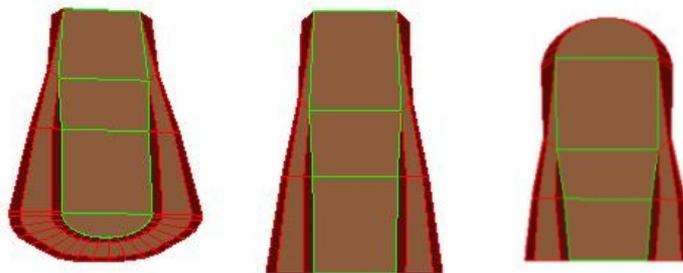


FIG. 2.36 – Différent points de vue d'une coupe transversale d'une unité de jointure sur son support

#### 2.4.4 Croissance en épaisseur (OCE)

L'opération de croissance en épaisseur (OCE) est une opération qui permet de simuler la croissance en épaisseur du tronc et des branches.

Plutôt que simuler cette croissance en ajoutant des volumes cubiques aux extrémités des axes de croissances (comme dans (Terraz, 1994)), la croissance en épaisseur est simulée par l'éclatement de tous les volumes des axes radiaux, *i.e.* d'ordre 1.1, 2.1..., incidents au bord de l'objet (volume à la fin des axes radiaux, *i.e.* ayant une face extrémité libre). L'opération d'éclatement de tels volumes peut être vue comme une propagation de l'éclatement le long des volumes des couches externes (adjacentes au bord). L'éclatement va des volumes du bord du tronc (ordre 1.1) jusqu'aux volumes du bord des branches principales (ordre 2.1), (figure 2.37). Cette opération se déroule en quatre temps pour éviter les problèmes de gestion du voisinage (ce problème est abordé sous-section 3.2.2.1) :

- sélection de tous les volumes candidats ;
- désaccouplage de toutes les faces côtés des volumes sélectionnés ;
- éclatement des volumes. Durant cette étape les nouveaux volumes créés héritent en parti des propriétés des volumes pères (les propriétés sont bien sûr modulées en fonction de la saison) ;
- reconstruction des relations de voisinage ;
- identification des faces côté libres restantes des volumes sélectionnés.

Sur chaque période (demi-année), le contrôle de l'évolution de la structure interne

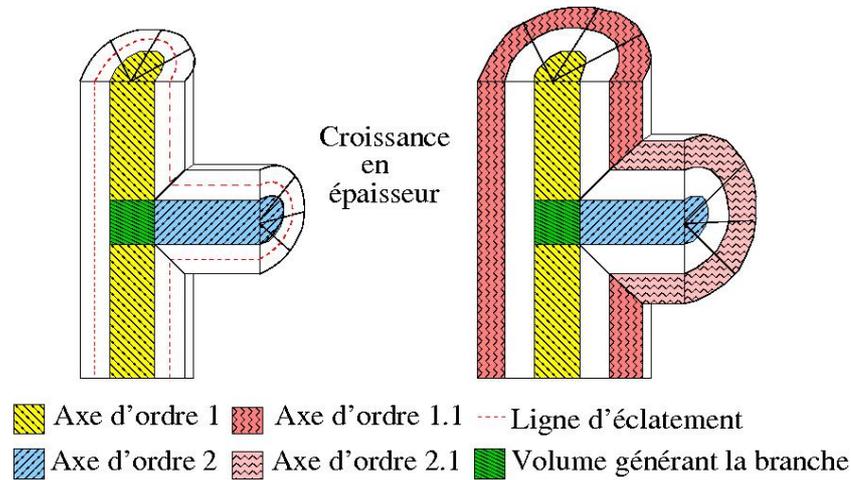


FIG. 2.37 – Croissance en épaisseur par propagation de l'éclatement des volumes des couches externes

de l'arbre peut donc se résumer ainsi :

- pour chaque axe de croissance radial : éclater le volume adjacent au bord (OCE) ;
- pour chaque axe de croissance principal : ajouter une UJ, puis une UCL ;
- ajouter les UB si nécessaires (nouvel axe de croissance principal) selon les spécificités du bois (nombre de branches...).

Comme dans la partie topologique, le contrôle du plongement est effectué à un haut niveau, en associant des valeurs spécifiques aux cernes et aux axes d'ordres différents. Nous associons à un cerne une hauteur maximum et une hauteur minimum, afin de calculer une valeur aléatoire dans ce segment pour chaque élément de celui-ci. Excepté dans le cas de la modélisation de phénomènes particuliers (endommagement du cambium, obstacle à la croissance ou bois de réaction), cette valeur est pondérée par les valeurs des volumes voisins afin de ne pas introduire trop de variations entre les différents éléments constitutifs d'un cerne (une cassure par exemple). Pour représenter la courbure des axes de l'arbre (tronc et branches), nous associons des angles d'insertions aux volumes représentant les axes de croissance principaux. Toutes ces données peuvent être trouvées dans des ouvrages spécialisés ((Vigue, 2002) et (Barry-Lenger *et al.*, 1999)).

#### 2.4.5 Algorithme général

Nous terminons ce chapitre en illustrant les imbrications des différentes briques de base que nous venons de présenter par un algorithme. Nous tenons à remarquer que, par soucis de clarté, l'algorithme présenté ici est une version extrêmement simplifiée du code développé.

```

1 void UneEtape()
2 {
3     if (carte_active->Horloge_val() == 0) { // phase initiale
4         RajouterMorceauTronc(NULL); // UCL
5         if (NbrNouvellesBranches>0) { //presence de branches
6             for (int i=0 ;i<NbrNouvellesBranches;i++) {
7                 RajouterBranche(NouvellesBranches[i]); } //ajout d'une UB
8                 Recoller(1.1,2.1); } } //Recollement des faces côtés des volumes d'ordre secondaires
9     else { //propagation
10        for (int i=0; i<2; i++) { // rajout des cernes de printemps et d'\et\`e
11            if ((NbBranche>0)&&(UneBrancheCree>0)) {
12                RajouterCouche(2.1); //croissance en \`epaisseur des branches (volume d'ordre 2)
13                NbrNouvellesBranches = 0; }
14            RajouterCouche(1.1); //croissance en epaisseur du tronc
15            Recoller(1.1,2.1); } //Recollement
16            if (NbUniteTronc>=(int)(carte_active->Horloge_val()/2.)) { //test sur la croissance en longueur
17                brin_marq = RajouterJoint(carte_active->traversCube(bFinTronc)); //rajout d'une UT
18                brin_marq->ptFace()->seulPourPl(brin_marq); //parametre pour le plongement
19                RajouterMorceauTronc(brin_marq); //UCL
20                if (NbrNouvellesBranches>0) { //ajout des nouvelles branches
21                    for (int i=0 ;i<NbrNouvellesBranches;i++)
22                        RajouterBranche(NouvellesBranches[i]); //UB
23                    Recoller(1.1,2.1); //Recollement
24                    NbrNouvellesBranches = 0; } }
25                carte_active->plongement(); //Calcul du plongement
26                main_window->MiseAJour(); //Mise a jour de la fenetre affichage
27        }

```

#### Annotations :

La variable *NbrNouvellesBranches* est une variable interne de la classe associée à la méthode présentée, elle est modifiée dans la méthode *RajouterMorceauTronc* et dans la méthode *UneEtape*. Elle sert à déterminer le nombre de branches qui vont se développer, et elle correspond à la taille du vecteur *NouvellesBranches*.

le vecteur *NouvellesBranches* est une variable qui stocke tous les brins supports des nouvelles branches qui se développent. Les bourgeons qui ne se développent pas sont stockés dans un autre vecteur. Ces vecteurs sont remplis au cours de l'ajout de l'UCL.

la variable *UneBrancheCree* permet de savoir si une branche a déjà été créée dans une étape précédente.

la fonction *Recoller(a,b)* sert à recoller les faces côté libres des volumes d'ordre a et b.

## 2.5 Premiers résultats

Après avoir présenté le modèle, nous devons introduire une nouvelle opération pour pouvoir le valider. Nous avons opté pour une validation visuelle des résultats plutôt qu'une validation technique (calcul de la masse de bois, ...). Nous allons donc introduire une nouvelle opération qui permet de simuler des coupes de bois pour visualiser *l'intérieur du bois*. Ensuite, en s'appuyant sur cette opération, nous pourrions illustrer la technique de validation du modèle.

### 2.5.1 Opération de coupe

Pour pouvoir valider la structure interne des grumes de bois modélisées, nous devons réaliser des coupes. Comme notre modèle est un maillage complexe, et que les opérations booléennes classiques ne s'appliquent aisément que sur des volumes isolés, nous avons choisi une approche modifiée et simplifiée de la méthode basée sur le co-raffinement (une étude plus complète sur le co-raffinement appliqué aux 3Gcartes peut être trouvée dans (Guiard, 2006)).

Le co-raffinement de deux objets est une opération qui génère un nouvel objet qui contient tous les éléments des deux premiers, modifiés, découpés pour prendre en compte les interactions et les superpositions existant entre eux ( *figure 2.38* ), image extraite de (Guiard, 2006). A partir de ces 3 objets, il est ensuite plus simple de définir les opérations booléennes classiques (union, intersection, différence).

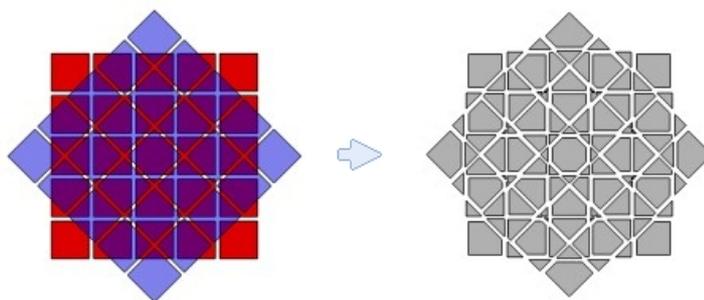


FIG. 2.38 – Co-raffinement de deux maillages 2D

Dans notre modèle, l'un des deux objets est un demi-plan<sup>25</sup>, ce qui permet de faciliter les calculs (pas de calculs d'intersections entre facettes, ...). De plus, comme notre but n'est de calculer qu'une intersection, nous ne créons pas de nouvel objet mais nous modifions celui fourni en entrée. Cette opération est, en partie, illustrée sur la ( *figure 2.39* ).

Comme cette opération s'appuie sur des critères de localisation spatiale (donc géométrique), nous ne pouvons plus garder le mode de plongement par défaut. C'est donc l'utilisateur qui gère le calcul du plongement manuellement, en fournissant directement

<sup>25</sup>Le demi-plan peut être vu comme un cube (maillage simple) surdimensionné par rapport à la grume incluse dans le demi-plan et dont une des faces est comprise dans le plan.

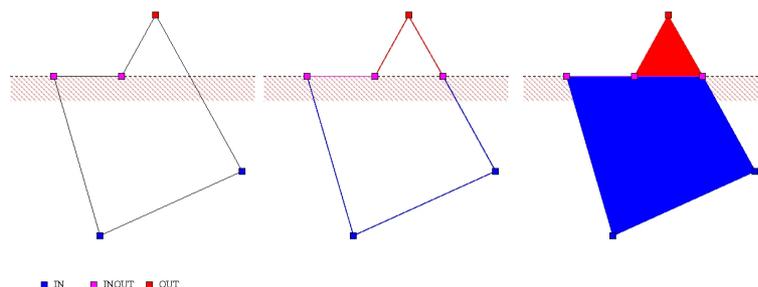


FIG. 2.39 – Marquage des objets pour la découpe (de gauche à droite) : marquage des sommets, éclatement et marquage des arêtes, éclatement et marquage des faces

les coordonnées des nouveaux sommets. Ce type de plongement manuel, nous permet de conserver la structure de l'objet (topologique et géométrique) pour pouvoir réitérer, le cas échéant le processus de découpe.

Pour le processus de traitement, nous utilisons un seul type de marqueur qui peut prendre les valeurs suivantes :

- **INDEF** : l'entité n'a pas encore été traitée,
- **IN** : l'entité est incluse dans le demi-plan,
- **OUT** : l'entité n'est pas incluse dans le demi-plan,
- **INOUT** : l'entité est dans le plan du demi-plan (à la frontière).

Les entités que nous manipulons sont les sommets, arêtes, faces et volumes (au sens topologique).

Nous commençons par parcourir tous les sommets de la carte. Si le sommet en cours de traitement est dans le demi-plan, il est marqué **IN**, si celui-ci est dans le plan, il est marqué **INOUT**, sinon il est marqué **OUT**. Si tous les sommets possèdent un marqueur identique, alors le processus s'arrête.

Sinon, nous passons au traitement des arêtes. Si l'arête n'est pas marquée comme déjà traitée, nous regardons les marqueurs sur les sommets qui la constitue :

- les sommets sont marqués **IN** et **IN** ou **IN** et **INOUT** : l'arête est marquée **IN**, et nous marquons de la même manière toutes les arêtes adjacentes (orbite  $(\alpha_2, \alpha_3)$ ) ;
- les sommets sont marqués **OUT** et **OUT** ou **OUT** et **INOUT** : l'arête est marquée **OUT**, et nous marquons de la même manière toutes les arêtes adjacentes (orbite  $(\alpha_2, \alpha_3)$ ) ;
- les sommets sont marqués **INOUT** et **INOUT** : l'arête est marquée **INOUT**, et nous marquons de la même manière toutes les arêtes adjacentes (orbite  $(\alpha_2, \alpha_3)$ ) ;
- les sommets sont marqués **IN** et **OUT** : l'arête est subdivisée (en tenant compte du voisinage) et le nouveau sommet ainsi créé prend comme coordonnées le point d'intersection entre le demi-plan et l'arête. Nous obtenons ainsi une arête avec des sommets **IN** et **INOUT** et une autre avec des sommets **OUT** et **INOUT**.

Puis, nous passons au traitement des faces. Pour chaque face, nous parcourons ses arêtes de proche en proche (bijection  $\alpha_0\alpha_1$ ) :

- toutes les arêtes constituant la face sont de type **INOUT**, alors la face est marquée **INOUT** ;

- toutes les arêtes constituant la face sont de type **IN** et **INOUT** (resp. **OUT** et **INOUT**), alors la face est marquée **IN** (resp. **OUT**) ;
- sinon nous parcourons de proche en proche jusqu'à trouver une première arête **IN** ou **OUT** qui sera le nouveau point de départ. Depuis celui-ci nous parcourons dans l'ordre les arêtes :
  - soit l'arête est marquée **INOUT**, alors celle-ci devient temporairement du même type que l'arête de départ et la nouvelle référence ;
  - soit l'arête est de type opposé, alors nous marquons le sommet de transition qui nous sert de demi-support pour l'éclatement de la face. Puis nous réitérons le processus sur l'arête courante pour trouver l'autre demi-support. La face est alors éclatée suivant le chemin déterminé par les deux demi-supports. Les deux faces ainsi obtenues sont alors traitées.

Une fois toutes les faces traitées, le modèle nous assure que les faces sont forcément **IN**, **OUT** ou **INOUT**.

Et pour finir, nous parcourons les volumes. Pour chaque volume, nous supprimons toutes les faces **OUT**, ce qui nous conduit à distinguer quatre cas :

- aucune face n'est supprimée, le volume reste tel quel ;
- le volume ne contient plus de faces, il est supprimé du modèle (en modifiant en conséquent les relations de voisinage) ;
- le volume ne contient plus que des faces **INOUT**, il est supprimé du modèle (en modifiant en conséquent les relations de voisinage) ;
- le dernier cas est le cas où des faces ont été supprimées et la structure restante contient des faces **IN** : nous devons refermer le volume en ajoutant une ou plusieurs nouvelles faces. Pour trouver les arêtes supports pour les nouvelles faces, il suffit de chercher les arêtes ayant des brins libres par  $\alpha_2$ . Pour déterminer le nombre et l'ordre de chaque face à rajouter, il suffit de parcourir de proche en proche les arêtes **INOUT** et libres par  $\alpha_2$ .

Le résultat final est donc la carte d'entrée modifiée qui correspond à la coupe.

Nous venons de décrire le modèle de base sur lequel s'appuie notre méthode. Cependant, pour des soucis de performances, certaines affectations de marqueurs sur les faces, arêtes et volumes ont été remontées à l'étape de traitement des sommets.

Une implémentation complète du co-raffinement a été envisagée, mais cette technique a été jugé trop complexe à mettre en place pour l'utilité que nous en avons à ce stade des travaux.

### 2.5.2 Résultats

La procédure de validation des résultats est très simple, nous nous basons sur des photos de coupe d'arbre pour valider visuellement la cohérence des résultats en fonction des espèces. Le but ici n'est pas de proposer une copie conforme de la photo, mais de proposer des images plausibles. Nous avons donc choisi des images de coupes transversales de bois pour mieux visualiser les fibres du bois et les jonctions des branches sur le tronc.

Dans cette première version, nous associons simplement des couleurs aux volumes

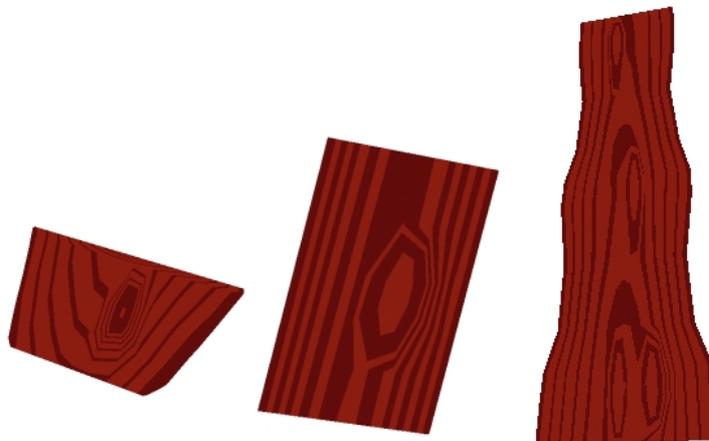


FIG. 2.40 – Quelques coupes transversales issues du premier modèle

modélisés (*figure 2.40*) . Ces premiers résultats sont encourageants, les déformations autour des nœuds sont bien visibles, mais l'aspect n'a pas l'air très réaliste. Nous développerons des techniques pour améliorer les rendus dans la *sous-section 4.1.1* .

## 2.6 Conclusion

Les travaux que nous venons de présenter ont fait l'objet d'une première communication à l'AFIG en 2004 (Guimberteau *et al.*, 2004) qui a obtenu le 2<sup>eme</sup> prix du meilleur papier, puis d'une publication dans la revue TSI (Guimberteau *et al.*, 2006).

Bien que les résultats soient visuellement acceptables, la technique de modélisation et l'expression de la croissance n'en restent pas moins complexes à exprimer.

Dans le chapitre suivant, nous formaliserons ce processus à l'aide d'une extension des L-systems que nous appellerons les 3Gmaps L-systems.

## Chapitre 3

# Formalisation du modèle de croissance

Comme nous l'avons déjà remarqué précédemment, des formes d'objets naturels parfois très complexes comme les arbres peuvent résulter de l'application récursive de règles de croissance simples à décrire. Ainsi des méthodes comme les L-systems et les grammaires de Chomsky ont connu un grand succès, et plus particulièrement en botanique et en imagerie graphique. L'application principale des L-systems porte sur la modélisation de la croissance de plantes et de coquillages.

Bien que la majorité de ces méthodes ne s'intéressent qu'à un aspect externe de l'arbre (même si le résultat final est un volume), nous nous appuyerons sur ce formalisme particulier pour formaliser nos travaux. Les 3Gmaps L-systems, que nous développerons dans la *section 3.2* sont une extension non pas des Cellwork L-systems, mais des L-systems appliqués aux cartes généralisées de dimension 3.

### 3.1 Etat de l'art

Les L-systems, ou systèmes de Lindenmayer, ont été introduits en 1968 par le botaniste Aristid Lindenmayer pour tenter de décrire des phénomènes qui interviennent lors de la croissance des plantes ou des cellules. Ces travaux ont ensuite été étendus par Prusinkiewicz pour décrire formellement les structures végétales. Il s'appuie sur le fait que lors de la croissance d'une plante, un même processus de croissance peut s'appliquer à différents niveaux de la plante (par exemple, le développement d'une branche se comporte comme celui du tronc). De la répétition d'un même processus de fabrication à différente échelle, naît une autosimilarité. Ces exemples de processus répétés peuvent être souvent définis de façon concise par des règles de réécriture assez simples (*sous-section 1.3*).

Dans les L-systems, les règles de production suivent les principes de la nature, elles sont donc appliquées parallèlement. Cette méthode d'application des règles de production est le point principal qui les diffère des grammaires de Chomsky (Chomsky, 1956) : en effet, les règles de production sont appliquées séquentiellement dans les grammaires de

Chomsky alors qu'elles sont appliquées en parallèle dans les L-systems. Cette différence reflète la motivation biologique des L-systems. L'application des règles de production a un impact essentiel sur les propriétés formelles d'un système de réécriture. Les langages décrits par les L-systems et les grammaires de Chomsky sont donc différents. Il existe par exemple des langages générés par les L-systems non-contextuels (dits 0L-systems) mais qui ne peuvent pas être générés par les grammaires de Chomsky non-contextuelles.

### 3.1.1 Présentation des L-systems

#### 3.1.1.1 Introduction par l'étude d'un cas pratique

Nous allons introduire les L-systems par l'étude d'un cas pratique de référence : la croissance des bactéries *Anabaena catenula* ou algue bleu-verte (*figure 3.1*) .



FIG. 3.1 – Bactéries *Anabaena catenula*

L'algue bleu-verte *Anabaena Catenula* est l'un des exemples les plus simples d'organismes multicellulaires avec une spécialisation des cellules (Golden *et al.*, 1998). C'est une algue en forme de filament, c'est à dire, une chaîne linéaire de bactéries. Sous le microscope, elle ressemble à collier constitué de perles cylindriques de différentes longueurs. Deux types de cellules sont responsables de la croissance d'*Anabaena Catenula*. Ces deux bactéries se différencient par leur dimension ainsi que par leur comportement lors de leur division cellulaire. Les cellules végétatives sont responsables de la photosynthèse (et donc de la subdivision cellulaire) et les cellules Hétérocysts sont responsable de la fixation de l'azote. Dans la suite, nous appellerons A la cellule végétative qui se subdivise pour donner deux cellules A et B, et B la cellule hétérocyte qui se métamorphose en cellule A (*figure 3.2*) . Comme les deux types de cellules se développent asymétriquement, elles peuvent être orientées soit vers la gauche, soit vers la droite dans la chaîne. Il y a donc, d'un point de vue pratique, quatre éléments différents, à savoir :

- l'élément  $\vec{A}$  définit la cellule de type A orientée vers la droite ;
- l'élément  $\overleftarrow{A}$  définit la cellule de type A orientée vers la gauche ;
- l'élément  $\vec{B}$  définit la cellule de type B orientée vers la droite ;
- l'élément  $\overleftarrow{B}$  définit la cellule de type B orientée vers la gauche.

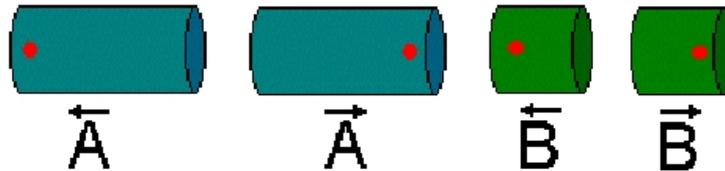


FIG. 3.2 – Les 4 types de bactérie

Comme nous venons de le voir, le processus de transformation cellulaire de chacune de ces 4 bactéries est le suivant :

- une  $\vec{A}$  (bactérie A orientée vers la droite) se divise après un certain temps en deux bactéries : une bactérie B orientée vers la droite ( $\vec{B}$ ) et une bactérie A orientée à gauche ( $\overleftarrow{A}$ ) ;
- de manière analogue, une  $\overleftarrow{A}$  se divise en deux bactéries : une  $\overleftarrow{B}$  et une  $\vec{A}$  ;
- une  $\vec{B}$  se transforme en  $\vec{A}$  ;
- une  $\overleftarrow{B}$  se transforme en  $\overleftarrow{A}$ .

Le processus de croissance appliqué plusieurs fois de suite peut être vu sur la (figure 3.3) .

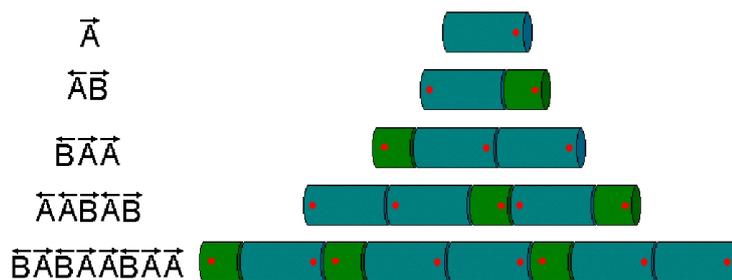


FIG. 3.3 – Processus de croissance

Nous venons ainsi de définir un L-system simple qui contient 4 types d'éléments. L'ensemble  $(\vec{A}, \overleftarrow{A}, \vec{B}, \overleftarrow{B})$  est appelé *l'alphabet* du L-system, chaque élément est appelé *symbole, lettre ou module*, et chaque combinaison d'éléments de l'alphabet est un *mot*. Les formules suivantes, qui sont appliquées simultanément lors du processus de croissance, sont appelées *règles de réécriture* ou *règles de production* :

$$\begin{array}{l}
- \overrightarrow{A} \rightarrow \overleftarrow{AB} \\
- \overleftarrow{A} \rightarrow \overrightarrow{BA} \\
- \overrightarrow{B} \rightarrow \overrightarrow{A} \\
- \overleftarrow{B} \rightarrow \overleftarrow{A}
\end{array}$$

De manière plus formelle, soit  $V$  un alphabet. Un L-system  $G$  est défini par un triplet  $G = \langle V, \omega, P \rangle$ , où  $\omega$  est un mot, non vide de  $V$  appelé axiome et  $P$  est un ensemble de règles de production sur  $V^*V$ .

### 3.1.1.2 L-systems de dimension 1

La classe de L-system la plus simple existante est appelée OL-system (OL pour 0-contexte ou sans contexte), c'est l'ensemble des L-systems qui n'ont pas d'interdépendance.

**Définition 3.1** Soit  $V$  un alphabet,  $V^*$  l'ensemble de tous les mots possibles de  $V$ , et  $V^+$  l'ensemble de tous les mots non vides de  $V$ . Un OL-system est un triplet ordonné  $G = \langle V, \omega, P \rangle$  où :  $V$  est l'alphabet du système,  $\omega \in V^+$  est un mot appelé axiome et  $P \subset V \times V^*$  est un ensemble fini de règles de production.

**Définition 3.2** Une règle de production, ou règle de réécriture, est définie par un doublet  $(a, \chi) \in P$  (on peut le réécrire  $P : a \rightarrow \chi$ ). La lettre  $a$  est appelée prédécesseur et  $\chi$  est appelée successeur.

Nous pouvons remarquer que  $a$  est un symbole alors que  $\chi$  peut représenter un mot. De plus, si aucune règle de réécriture n'est définie de manière explicite, alors la règle de production *identité* utilisée de manière implicite.

**Définition 3.3** Pour toute lettre  $a \in V$  qui n'a pas de règle de réécriture explicite, alors la règle de réécriture identité est définie comme suit :  
 $P : a \rightarrow a$ .

Ainsi ceci nous assure que chaque symbole de  $V$  possède au moins une règle de production.

Un DOL-system est un L-system déterministe, c'est à dire que chaque symbole de  $V$  possède exactement une règle de production, ou encore de manière formelle :  
 $\forall a \in V, \exists ! \chi \in V^*$ .

A partir des ces définitions, un grand nombre de L-systems ont été définis :

*Les L-systems stochastiques* (Prusinkiewicz et al., 1990b)

Contrairement aux DOL-systems, qui à partir d'un axiome et de règles de production fixes donneront toujours le même mot à la  $n^{eme}$  étape, les L-systems stochastiques permettent d'associer à un même symbole plusieurs règles de production différentes grâce à des probabilités d'application de celles-ci. La syntaxe est la suivante :

$P : \text{prédécesseur} \rightarrow \text{successeur} : \text{prob}$

*Les L-systems paramétriques* ((Prusinkiewicz *et al.*, 1990a) et (Hanan, 1992))

Comme pour les L-systems stochastiques, il est possible d'associer à des symboles un ou plusieurs paramètres. Un symbole composé de symboles et de paramètres est appelé *module*. Les règles de productions peuvent utiliser ces paramètres pour évaluer une condition booléenne  $\beta$ . Le prédécesseur est remplacé par son successeur si et seulement si la condition booléenne est vérifiée :

P : prédécesseur  $\{bloc_1\}\beta\{bloc_2\} \rightarrow$  successeur

Les blocs optionnels  $bloc_n$  sont utilisés pour des variables qui peuvent être utilisées dans la condition ou comme nouveaux paramètres pour le module. Le premier bloc s'exécute avant la condition tandis que le deuxième ne s'exécute que si la condition est vérifiée.

*Les L-systems sensibles au contexte ou contextuels* ((Lindenmayer, 1968) et (Prusinkiewicz *et al.*, 1990b))

Les règles de production appliquées dépendent du contexte du prédécesseur dans la relation de voisinage. Le prédécesseur doit se produire juste après un contexte gauche ( $lc$ ) concordant et juste avant un contexte droit concordant ( $rc$ ) afin que la règle de production puisse s'appliquer. La syntaxe est la suivante :

P :  $lc < \text{prédécesseur} > rc \rightarrow$  successeur

Nous pouvons distinguer deux types de L-systems sensibles au contexte :

- dans les *1L-systems*, les règles de production qui peuvent s'appliquer dépendent soit du contexte gauche, soit du contexte droit de l'ensemble prédécesseur ;
- dans les *2L-systems*, les règles de productions qui peuvent s'appliquer dépendent du contexte gauche ou droit de l'ensemble prédécesseur.

Nous pouvons remarquer que les différents types de L-systems ne s'excluent pas mutuellement. En effet une règle de production peut combiner les différents types de L-systems, et donc être de la forme :

P :  $A(x) < B(y) > C(z) : \{m=x+1\}m > z \{n=z-1\} \rightarrow A(m)B(n) : 0.2$

### 3.1.1.3 Interprétation graphique

Un L-system peut être représenté graphiquement au moyen d'une tortue graphique. La tortue graphique est un artéfact informatique introduit par Seymour Papert, créateur principal du langage LOGO, ((Abelson *et al.*, 1981), (Szilard *et al.*, 1979) et (Prusinkiewicz *et al.*, 2001)).

#### Dans le plan

Une tortue graphique est capable de lire le mot du L-system et d'interpréter la suite d'instructions associées à chaque symbole. L'état de la tortue graphique est défini par un triplet correspondant aux coordonnées de la position de la tortue (x,y) et à la direction de la tête de la tortue (angle par rapport à l'axe des abscisses).

Par exemple, prenons l'alphabet suivant : F,f,+,-, où à chaque symbole sont associées les instructions suivantes :

- F : avancer tout droit en dessinant une ligne d'une longueur  $l$ ,

- f : avancer tout droit sans dessiner de ligne d'une longueur  $l$ ,
- + : tourner à droite d'un angle  $\alpha$ ,
- - : tourner à gauche d'un angle  $\alpha$ .

Considérons maintenant le mot suivant :  $F-F+F+f-FF$ . Le mot va donc être interprété symbole par symbole de la gauche vers la droite :

- F avancer tout droit en dessinant une ligne,
- - tourner à gauche d'un angle  $\alpha$ ,
- F avancer tout droit en dessinant une ligne,
- + tourner à droite d'un angle  $\alpha$ ,
- ...

En posant  $\alpha=\pi/2$  et  $l=1$ , nous obtenons la (figure 3.4) .



FIG. 3.4 – interprétation du mot  $F-F+F+f-FF$

Dans l'exemple précédent chaque lettre correspondait à une instruction unique, mais dans la pratique ce n'est pas toujours le cas. Dans la nature, chaque organisme peut être modélisé comme un assemblage d'unités discrètes appelées *modules*. Ensuite à chaque symbole est associé un module. Ce symbole qui représente donc le type du module utilisé pour l'interprétation est souvent appelé aussi par extension module. La nature du module n'est pas prédéfinie par le formalisme, car elle peut aussi bien représenter une cellule dans les cas de petits organismes que des unités fonctionnelles (entre-nœuds, fleurs) dans le cas de plantes plus complexes. De plus, chaque module peut être caractérisé (dans le cas des L-systems paramétriques) par des valeurs numériques spécifiées pour chaque volume.

### Dans l'espace

Dans l'espace, l'état de la tortue graphique ne peut plus être défini par un triplet (il manquerait alors des informations quant à la direction de la tortue et son orientation sur cette direction). Il faut donc augmenter le nombre de paramètres à fournir au modèle. Outre la position de la tortue, trois nouvelles valeurs sont introduites qui correspondent respectivement au lacet, au roulis et au tangage dans un avion (figure 3.5) . Nous partons du principe que la tortue est munie d'un repère local orthonormé où les 3 axes, qui passent par le centre de la tortue sont définis comme suit :

- le vecteur *tête* qui regarde dans la direction courante de la tortue (ceci correspond à l'axe de roulis dans un avion) ;
- le vecteur *gauche* qui regarde à gauche de la tortue (ceci correspond à l'axe de tangage dans un avion) ;

- et le vecteur *dessus* qui regarde au dessus de la tortue (ceci correspond à l'axe de lacet dans un avion) ;

Puis, dans ce repère local, nous définissons 3 angles de rotations (un angle par axe).

Dans l'espace, l'état graphique de la tortue est donc défini par un sextuplé  $(x,y,z,h,u,l)$  qui correspond à :

- sa position dans l'espace  $(x,y,z)$  ;
- sa direction et son orientation dans l'espace  $(h,u,l)$ .

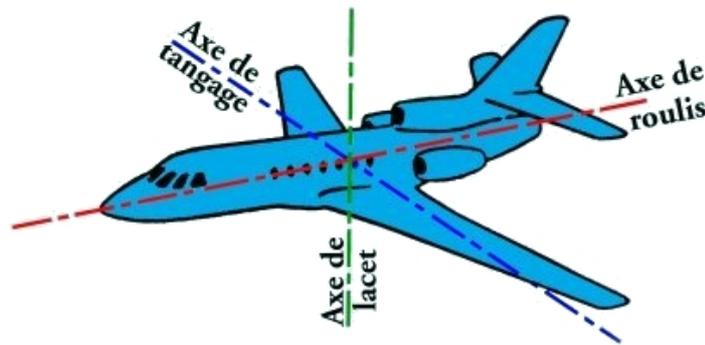


FIG. 3.5 – Les notions de lacet, roulis, tangage dans un avion

Les règles sont ensuite appliquées comme dans le plan. Mais le plus souvent des objets sont appliqués à chaque position de la tortue.

#### 3.1.1.4 Application aux arbres, les Tree L-systems

Comme nous l'avons vu précédemment, les différents axes de croissance des plantes se développent simultanément. Par exemple, les branches d'un arbre se développent presque indépendamment les unes des autres (*figure 3.6*)<sup>1</sup>. Cette singularité de la croissance peut être simulée avec des systèmes de réécriture en parallèle et l'introduction de la dépendance au contexte.

Cependant, jusqu'ici, la tortue interprète un caractère comme une séquence de segments de lignes ou un module. La longueur et les angles dépendent du module lui-même. Elle ne peut simuler les ramifications des plantes qu'en revenant en arrière grâce à une nouvelle chaîne de symboles. Il devient donc délicat et lourd de modéliser de manière géométrique ces branchaisons successives. Pour pallier à ce problème, deux nouveaux symboles sont introduits pour délimiter une branche, *le crochet gauche* [ *et le crochet droit* ]. Ces symboles utilisent la notion de pile d'accumulation :

- [ : stocke l'état courant de la tortue dans la pile d'accumulation (position, direction, couleur, ...);

<sup>1</sup>D'un point de vue plus rigoureux, les interdépendances principales se situent au niveau de la phyllotaxie, du transport de la sève ou de la concurrence lorsque ces dernières sont trop proches.



FIG. 3.6 – Ramifications d'un arbre

- ] : récupère le dernier état stocké dans la pile d'accumulation et modifie l'état de la tortue en conséquent.

Ces symboles permettent, en particulier, de *simuler* simplement le retour en arrière dans la structure pour générer les ramifications. Munis de ce nouveau symbole, et du fait que les règles de réécriture soient appliquées en parallèles, les L-systems sont bien adaptés pour générer des arbres ou des plantes réalistes (*figure 3.7*) :

Durant l'étape de réécriture, le prédécesseur est remplacé par son successeur grâce aux règles de productions. La première règle de production spécifie que l'apex<sup>2</sup>, en vert (*figure 3.7*) , crée un embranchement de sous-structures composées de trois apices et un entre-nœud. La seconde règle de production augmente simplement la longueur de l'entre-nœud. Le modèle assure aussi que les règles de production sont invariables en ce qui concerne le redimensionnement, ce qui signifie que si la longueur du prédécesseur change (par un facteur  $r$ ), le successeur est redimensionné par le même facteur.

De manière plus formelle, un Tree L-system est défini de la manière suivante :

**Définition 3.4** *Un Tree OL-system est défini par 3 composants : un ensemble d'arêtes labellisées  $V$ , un arbre initiale  $\omega$  avec des labels de  $V$ , et un ensemble d'arbres de production  $P$ .*

*Soit un L-system  $G$ , un arbre axial  $T2$  dérive directement d'un arbre  $T1$ , noté  $T1$  implique  $T2$ , si  $T2$  est obtenu à partir de  $T1$  en remplaçant simultanément chaque arête de  $T1$  par son successeur en accord avec l'ensemble de règles de production  $P$ . Un arbre*

---

<sup>2</sup>Un apex, plu. apices : c'est le sommet de la pousse de croissance, à l'origine des organes. Il est constitué du méristème entouré d'ébauches foliaires. Il est le lieu de la croissance, de la ramification et de la différenciation entre feuilles, tiges et fleurs...

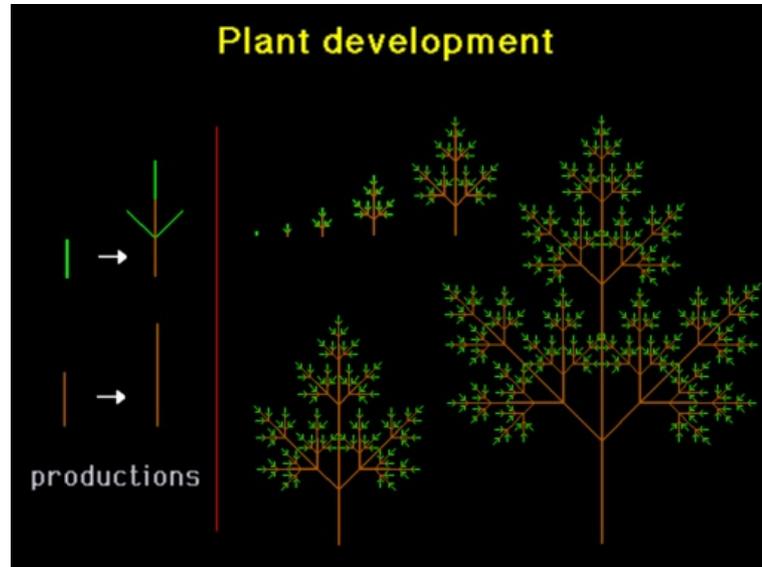


FIG. 3.7 – Un L-system relativement simple : ses deux règles de réécriture, l'axiome et l'application successive des règles à partir de l'axiome.

$T$  est généré par  $G$  dans une dérivation de longueur  $n$  si il existe une séquence d'arbres  $T_0, T_1, \dots, T_n$  telle que  $T_0 = \omega$ ,  $T_n = T$  et  $T_0$  implique  $T_1$  implique  $\dots$  implique  $T_n$ .

A partir de ces définitions, de nombreux modèles ont été développés. Ces modèles peuvent prendre en compte les propriétés physiologiques des plantes (photosynthèse, transport de la sève, ...) ou encore la concurrence entre espèces. Dans tous ces modèles, un module est associé à chaque lettre de l'alphabet. Ces modèles donnent des silhouettes d'arbres réalistes, mais posent des problèmes au niveau des jonctions entre les différents modules.

### 3.1.1.5 Interprétation graphique, les problèmes de jonctions

Pour interpréter graphiquement un L-system, il faut associer à chaque lettre du mot un objet graphique (cylindre, module, ...). La première idée est d'associer tout simplement un cylindre à chaque lettre du L-system pour modéliser les volumes du tronc et des branches (Prusinkiewicz *et al.*, 1990b). Comme le montre la (figure 3.8)<sup>3</sup>, cette technique présente quelques défauts. Par exemple, un effet d'escalier au niveau de l'empilement des cylindres (de diamètre décroissant) qui simulent l'affinement du tronc ou des branches ; ou encore, si les cylindres ne sont pas empilés dans l'axe, il se produit un chevauchement entre les cylindres au niveau de la zone de contact.

Pour palier ces défauts, (Max, 1990) propose d'utiliser des cônes-sphères. Une cône-sphère est un volume délimité par deux sphères et le tronc de cône généré par les

<sup>3</sup>Images extraites de(Maritaud, 2003)

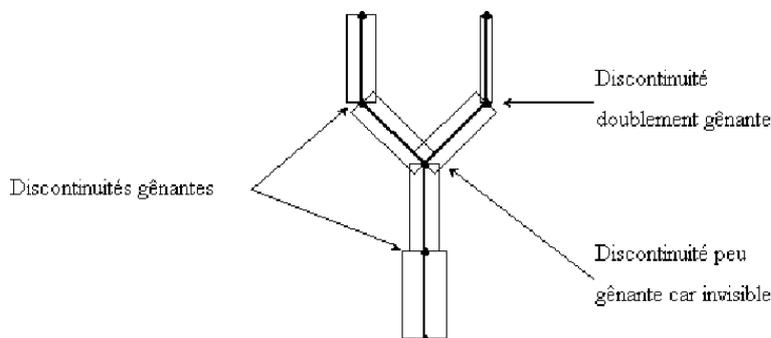


FIG. 3.8 – Schéma d'un arbre représenté avec des cylindres

tangentes aux deux sphères (Max, 1990). Bien que cette technique donne de meilleurs résultats, elle ne permet pas de faire disparaître la totalité des zones anguleuses (figure 3.9) .

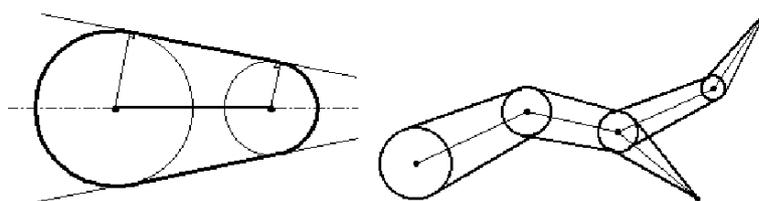


FIG. 3.9 – A gauche : un cône-sphère. A droite : branche formée de cônes-sphères

Ce modèle discret a été étendu au cas continu, le cylindre généralisé (Agin *et al.*, 1976), (Bloomenthal, 1985) (Grimm, 1999), (Lintermann *et al.*, 1999), (Prusinkiewicz *et al.*, 2001) et (Boudon *et al.*, 2003). Un cylindre généralisé peut être défini comme une suite infinie de cônes-sphères de hauteur presque nulle. De manière analogue, nous pouvons définir un cylindre comme l'ensemble de tous les points situés à une distance  $r$  ( $r \in \mathbb{R}^{+*}$ ) d'un segment défini par deux points non confondus (appelé *squelette* du cylindre). Dans le cas des cylindres généralisés, le squelette n'est plus un segment, mais une portion de courbe quelconque. De plus la distance  $r$  peut varier le long du squelette. Bien que cette définition semble très satisfaisante, d'un point de vue pratique, pour la visualisation, les cylindres généralisés sont discrétisés en tronc-cône ou cônes-sphère.

(Bloomenthal, 1997), (Mech *et al.*, 1996) proposent de généraliser ce modèle en utilisant des surfaces implicites.

Cette technique présente de nombreux avantages, en particulier elle permet d'adoucir les angles au niveau des raccords, mais les calculs aux niveaux des raccords de branches restent très complexes. Des travaux ont été développés sur ces problèmes, mais nous ne nous y attarderons pas (Galbraith *et al.*, 2004), (Maritaud, 2003).

Une dernière approche peut être de simplement utiliser des surfaces de subdivision. Cette technique lisse les zones anguleuses, mais elle lisse également les défauts du bois, en particulier au niveau des branchements (Tobler *et al.*, 2002).

### 3.1.2 Présentation des Maps L-systems

A l'origine, les L-systems n'avaient pas de concept de cellule de dimension 2. Chaque symbole dans un string L-system pouvait représenter indifféremment une structure cellulaire ou toute autre sous-unité d'un organisme, mais la structure sous-jacente de l'unité n'était pas explicitée. Pour étendre ce concept en dimension 2, Lindenmayer et Rozenberg (Lindenmayer *et al.*, 1979) ont introduit un nouveau formalisme appelé les Maps L-systems. Ce dernier est basé sur la notion de carte (graphe planaire avec des cycles). Les différences principales avec les grammaires de L-systems ordinaires se situent au niveau de :

- l'axiome primaire, souche (ou cellule mère) qui doit former une surface fermée (cette hypothèse forte assure que le système commence par une surface valide) ;
- l'interprétation des systèmes de branchaison (nous y reviendrons de manière plus détaillée un peu plus loin).

Les Maps L-systems étaient à l'origine employés pour modéliser le développement cellulaire dans la morphogenèse. Ici, les symboles prennent la représentation d'un mur de cellule. Une couche cellulaire est représentée comme une carte géométrique où chaque bord dans la carte a une étiquette et chaque cellule (région) de la carte a un type de cellules.

Avec les Maps L-systems, des branches sont reliées à leurs voisines et celles qui n'ont pas de vis à vis sont simplement enlevées. Ceci mène donc à une subdivision croissante de la surface originale. Formellement, un Map L-system est composée de :

- un ensemble de labels d'arêtes ;
- un ensemble types de cellules ;
- un ensemble de règles de production de la forme  $\alpha \rightarrow \beta$  qui, lorsqu'elles sont appliquées subdivisent chaque arête (bord) labellisée  $\alpha$  en une séquence d'arêtes labellisées  $\beta$  ;
- un ensemble de cellules de production qui opèrent sur des séquences de murs spécifiques ;
- une cellule de départ.

Une étape de dérivation se décompose en deux sous-étapes :

- les règles de production de bord des cellules sont appliquées ;
- les règles de production sur les cellules sont appliquées.

Par exemple, considérons le Map L-system suivant (*figure 3.10*)<sup>4</sup>. La première étape consiste à remplacer chaque arête de la carte par ses arêtes successeurs en suivant les règles de production. Ici, il s'agit :

- d'éclater les arêtes de type **A** en deux arêtes de type **B** et rajouter deux arêtes de type **A** au milieu (seule l'arête à l'intérieur de la cellule est retenue sur le schéma). Nous obtenons ainsi un état transitoire où il peut exister des cellules à

<sup>4</sup>Les images de cette partie sont issues de (Prusinkiewicz *et al.*, 1990b).

bord non fermé. Chaque région est ensuite analysée pour trouver des marqueurs concordants (des arêtes de même type ayant une extrémité libre). Si un tel cas se présente, les arêtes sont alors recollées et la cellule est donc divisée en deux (ici il n'y a aucune règle de production sur les cellules), sinon elles sont tout simplement supprimées du modèle ;

- de modifier les arêtes de type **B** en type **A**.

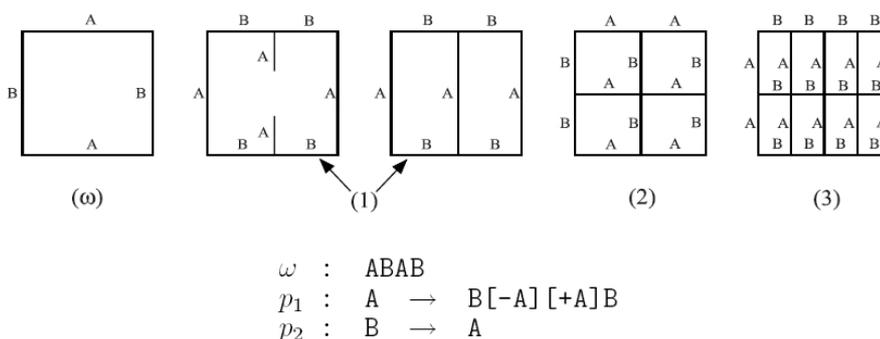


FIG. 3.10 – Exemple de Map L-system : dans le premier step, une distinction est effectuée entre la phase de réécriture des arêtes et la connexion des marqueurs concordants

### 3.1.3 Présentation des Cellworks L-systems

Les Cellwork L-systems ont été introduits afin de représenter l'aspect tri-dimensionnel des couches cellulaires. Un cellwork est un ensemble fini de cellules (volumes). Chaque cellule est délimitée par une paroi (un ensemble de faces). Chaque face constituant la paroi est délimitée par une séquence circulaire et finie d'arêtes. Formellement, un Cellwork L-system est défini par :

- un alphabet fini (de labels d'arêtes) ;
- un cellwork de départ (ensemble de cellules) ;
- un ensemble fini de règles de production.

Les labels des murs sont utilisés pour déterminer la liste des parois sur lesquelles les règles de production s'appliquent. Comme pour les Map L-systems ( *section 3.1.2* ), les règles de production sur les arêtes peuvent donner des arêtes avec un point de liberté (elles ne sont rattachées à la paroi que par un seul sommet, l'autre étant libre). C'est un mécanisme spécifique qui reconnecte ces arêtes afin de créer de nouvelles parois, puis un autre reconnecte ces parois afin de créer de nouvelles cellules.

La (*figure 3.11*) fournit un exemple de Cellwork L-system.

Cependant, il est encore plus difficile que pour les Maps L-systems, de déterminer des règles de production robustes pour un ensemble de cellules données (cellwork).

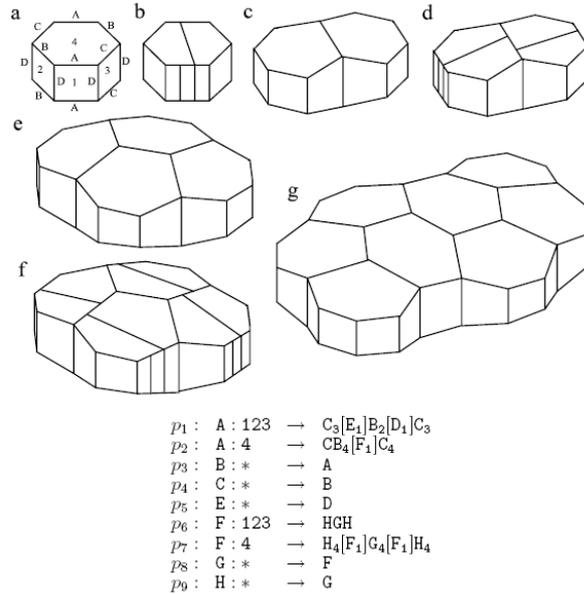


FIG. 3.11 – Séquence de développement de cellules de l'épiderme : (a) l'ensemble cellulaire de départ ; (b), (d) et (f) cellworks immédiatement après la division des cellules ; (c), (e) et (g) les cellules correspondantes une fois l'état d'équilibre atteint

## 3.2 Introduction des 3Gmap L-systems

Le système que nous proposons, les 3Gmaps L-systems, essaie de lever les limitations des Cellworks L-systems décrites dans la *sous-section 3.1.3* :

- pour les règles de production des volumes, nous utilisons directement des opérations de haut niveau sur ces derniers, ce qui simplifie les spécifications du modèle ;
- nos travaux sont basés sur l'utilisation de cartes généralisées de dimension 3 (Lienhardt, 1994), ce qui nous permet de représenter la topologie de toute subdivision de l'espace ;
- la dépendance au contexte est gérée grâce aux relations d'adjacences entre les volumes.

Dans cette partie, nous allons décrire le modèle que nous proposons ainsi que les opérations associées aux règles de production que nous avons développées. L'implémentation utilise, bien sûr, le modèle topologique basé sur les cartes généralisées de dimension 3 présentées dans le *chapitre 2*.

Cependant, afin d'être plus précis dans la définition des règles de réécritures, nous ne détaillerons pas les aspects techniques liés à l'utilisation du modèle (la plupart de ces opérations pouvant se retrouver dans le *chapitre 2*). Plus généralement, toutes les opérations topologiques que nous allons utiliser ici (création, éclatement et recollage de volumes) sont classiques et formellement décrites dans la littérature (pour plus de détails, les lecteurs pourront se référer à (Dufourd, 1991), (Bertrand *et al.*, 1993b), (Lienhardt, 1994)).

Nous commencerons par introduire des simplifications et des notations par rapport au modèle décrit dans le *chapitre 2*. Puis, dans un second temps, nous détaillerons les opérations de bases correspondantes aux règles de production.

### 3.2.1 Simplifications et notations

#### Labels des volumes et des prismes

Le but recherché ici, est de simplifier le contrôle du processus d'évolution en définissant des règles de production qui agissent directement sur les volumes et non sur les arêtes comme dans les Maps et les Cellworks L-systems. Cependant, le modèle décrit dans le *chapitre 2* n'est pas tout à fait adapté à cette nouvelle approche. Ainsi, nous choisissons d'associer un label à chaque volume. Dans la suite de ce chapitre, le symbole  $A$  désigne un volume de label  $A$ .

Pour simplifier le contrôle de l'évolution et pour pouvoir ajouter facilement une orientation aux volumes, nous pouvons conserver les volumes de type prisme. Nous utilisons cette orientation pour simplifier les spécifications de la grammaire pour l'application à la modélisation de bois car il existe des directions de croissance spécifiques dans le bois (croissance en longueur et en largeur). Cette orientation nous permet ainsi d'exploiter les L-systems traditionnels (voir la *section 3.5.3*).

Les volumes les plus utilisés sont des prismes réguliers comme pour le premier modèle. L'ordre d'un prisme est donné par l'ordre de sa base. Un prisme d'ordre  $n$  est noté  $A(n)$ . Comme nous le verrons plus tard, l'ordre d'un volume est le premier paramètre pour la création d'un volume. Les autres paramètres, qui sont surtout relatifs à la géométrie, seront introduits plus tard.

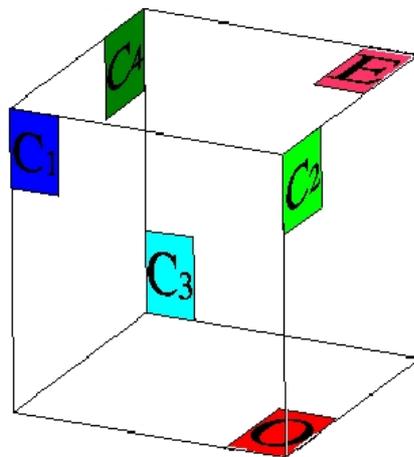


FIG. 3.12 – Labels associés aux faces d'un prisme d'ordre 4

## Labellisation des faces de prismes réguliers

Pour simplifier le contrôle des règles d'évolution, nous devons modifier la labellisation précédemment introduite dans le *chapitre 2*. En effet, sur le modèle précédent, il n'existait que 3 types de labels O, E et C (origine, extrémité et côté). Par la suite, nous aurons besoin de connaître chaque face avec plus précision. Nous associons donc un nouveau label à chaque face du prisme  $A(n)$  ( $n \in N, n \geq 3$ ).

Ces labels sont définis comme suit (*figure 3.12*) :

- la base du prisme  $A$  est appelée «face origine »et notée  $A_O$  ;
- le fin du prisme est appelée «face extrémité »et notée  $A_E$  ;
- les autres faces sont appelées «faces côté »et notées  $A_{C_1}, A_{C_2}, \dots, A_{C_n}$ . L'ordre de classification des faces côté suit les faces adjacentes des arêtes de la face origine dans un ordre trigonométrique par rapport à la normale à la face origine.

Pour pouvoir sélectionner des familles d'objets similaires, nous introduisons les symboles  $*$  et  $-$ . Ainsi, en considérant le prisme  $A(n)$ , ( $n \in N, n \geq 3$ ) :

- $A_*$  désigne toutes les faces du volume  $A$  ;
- $A_{C*}$  désigne toutes les faces côtés du volume  $A$  ;
- $A_{C*-2}$  désigne toutes les faces du volume  $A$  sauf la face  $C_2$ <sup>5</sup>.

## Relations d'adjacence et dépendance au contexte

Toutes les opérations que nous allons détailler dans la suite de ce chapitre dépendent des relations de voisinage des objets (adjacents ou non, ...)

Cependant, si la dépendance au contexte s'exprime assez simplement en dimension 1 avec le concept de prédécesseur et successeur, l'écriture de tels concepts devient rapidement très complexe quand la dimension des objets augmente. Dans notre modèle, la dépendance au contexte s'exprimera au travers des relations d'adjacence entre les différents objets topologiques (*figure 3.13*) :

- deux faces  $A_X$  et  $B_Y$  sont dites adjacentes si elles ont une arête commune ;
- deux volumes  $A$  et  $B$  sont dits voisins ou adjacents si ils ont au moins une face adjacente.

Comme pour les L-systems traditionnels, la relation d'adjacence est représentée par les symboles  $<$  et  $>$ . Comme la relation d'adjacence est une opération symétrique, nous pouvons utiliser indifféremment l'un ou l'autre de ces symboles. Le symbole est précédé et suivi par une liste de face qui vérifient la relation d'adjacence. Par exemple :

- $A_E > B_O$  ou  $A_E < B_O$  : adjacence entre la face extrémité du volume de label  $A$  avec la face origine du volume de label  $B$  ;
- $A_* > B_{C_2}$  ou  $A_* < B_{C_2}$  : adjacence entre une face non spécifiée d'un volume de label  $A$  et la face  $C_2$  d'un volume de label  $B$  ;
- $A_* > B_*$  ou  $A_* < B_*$  : adjacence entre un volume de label  $A$  et un volume de label  $B$  par une face non spécifiée.

---

<sup>5</sup>Par simple abus de notation, nous omettons le préfixe C ici, car il n'y a aucune ambiguïté, mais ce n'est pas toujours le cas.

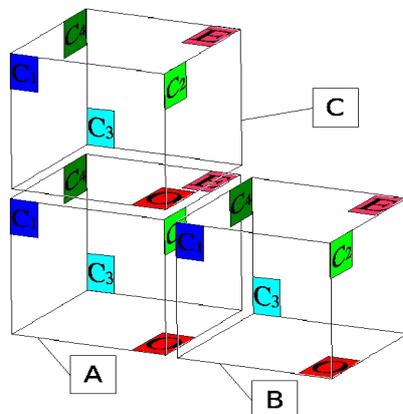


FIG. 3.13 – Faces et volumes adjacents : C est adjacent à B par les faces C2 et E (la condition  $C_{C2} > B_E$  est donc vraie)

### Introduction de la dépendance au contexte dans la grammaire

Comme le nombre de relations d'adjacence peut être important (il dépend du nombre de volumes voisins), nous introduisons la dépendance au contexte comme les autres conditions pour l'application d'une règle dans la grammaire. Plus précisément, nous utilisons le formalisme suivant (formalisme *cpfg*, voir (Prusinkiewicz *et al.*, 1999)) :

*predecesseur* :  $block_1 \text{ cond } block_2 \rightarrow \textit{successeur}$

où :

- *cond* est une condition booléenne qui contrôle l'application de la règle de production ;
- *block<sub>1</sub>* est un bloc de code toujours exécuté ;
- *block<sub>2</sub>* est un bloc de code optionnel qui n'est exécuté que si la condition *cond* est vérifiée.

Par exemple :  $A : A_O < D_E \text{ et } A_E > E_O \rightarrow BC$  signifie que la règle de production  $A \rightarrow BC$  n'est appliquée que si le volume *A* est adjacent par sa face origine avec la face extrémité du volume *D* et adjacent par sa face extrémité avec la face origine du volume *E*.

### 3.2.2 Règles de base et opérations correspondantes

#### 3.2.2.1 Eclatement d'un volume

Cette opération éclate un volume en deux parties. Pour éclater un volume *V*, nous avons besoin, d'un point de vue technique, d'un chemin fermé le long des faces de *V*. Les faces concernées de *V* sont alors éclatées le long de cette trajectoire. Ensuite les deux nouveaux volumes ainsi générés sont refermés. Dans cette forme les spécifications de l'opération sont difficiles à mettre en place. Ainsi, nous utilisons un modèle simplifié :  $A \xrightarrow{F} BC$  où *F* est une face du volume *A*. Dans cette opération, nous éclatons toutes

les faces adjacentes à  $F$  et nous refermons les deux volumes obtenus avec une nouvelle face du même degré que le nombre de faces adjacentes à  $F$ . Excepté pour le départ du processus d'éclatement à partir d'une face origine ou une face extrémité d'un prisme, les volumes résultants n'ont pas le même nombre de faces que le volume de départ voir (figure 3.14). Durant cette opération nous réorganisons les labels de manière automatique (nous détaillerons cette opération par la suite).

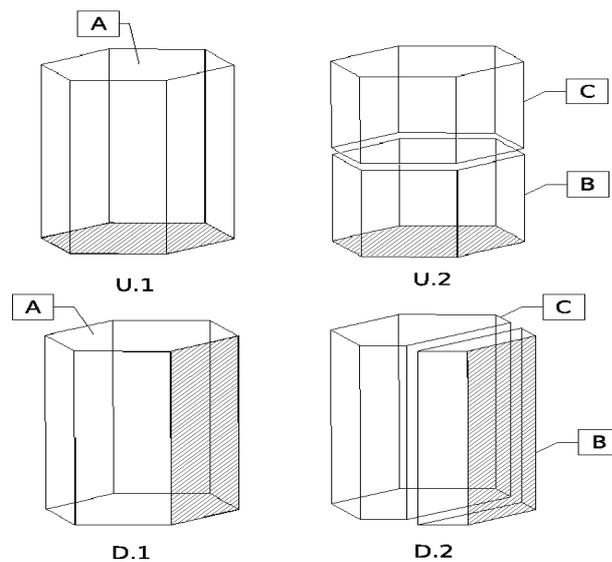


FIG. 3.14 – L'opération d'éclatement de volume : (HAUT) U.1 Volume initial (face masquée : O), U.2 Résultat de la règle  $A \rightarrow^O BC$ , (BAS) D.1 Volume initial (face masquée : C1), D.2 Résultat de la règle  $A \rightarrow^{C1} BC$

**Adjacence :** Afin de garder une bonne cohérence avec le modèle topologique utilisé, l'opération d'éclatement prend en compte les relations de voisinage de la manière suivante :

- toutes les faces voisines sont éclatées simultanément ;
- toutes les faces voisines directes sont refermées simultanément ;
- les labels des faces ne sont mis à jours que lorsque toutes les opérations d'éclatement sont terminées.

**Propagation de l'éclatement :** Quand plusieurs volumes voisins sont concernés par l'éclatement, le modèle distingue deux cas :

- **Eclatement en série** : dans ce cas, le premier volume est éclaté, ses labels sont réagencés, et ensuite le second volume est éclaté. Cette opération crée des arêtes et des faces dans la zone d'adjacence (figure 3.15.b). Cette opération est notée :  $A \rightsquigarrow^O BC$

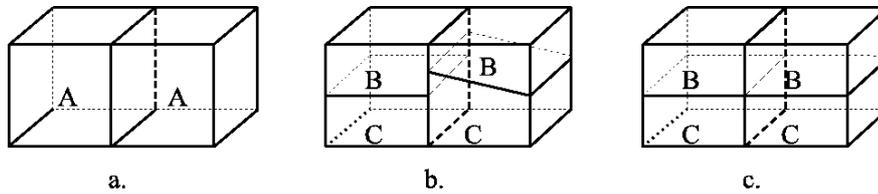


FIG. 3.15 – Propagation de l'éclatement de volumes : a. Volumes initiaux. b. Résultat d'un éclatement en série. c. Résultat d'un éclatement en parallèle

- **Eclatement en parallèle** : dans ce cas les volumes voisins sont éclatés simultanément (ce qui permet d'éviter la création de faces supplémentaires dans la zone d'adjacence) (*figure 3.15.c*). Nous notons cette opération :  $A \xrightarrow{O} BC$

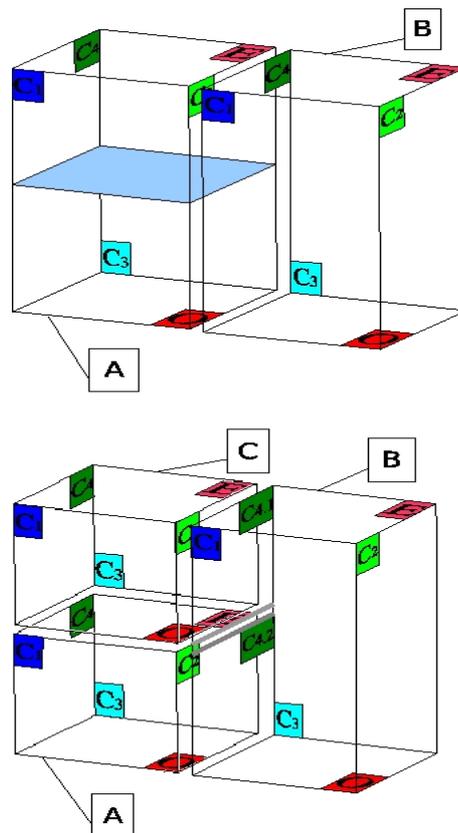


FIG. 3.16 – Réorganisation des labels : en haut, les volumes avant l'éclatement et en bas, le résultat après l'éclatement ( $A \xrightarrow{O} AC$ ), et la réorganisation des labels

**Réorganisation automatique des labels :** Quand toutes les opérations décrites précédemment sont terminées, les labels des faces sont réorganisés afin de préserver une bonne cohérence du modèle, et ainsi permettre une interprétation géométrique. Lors de cette opération plusieurs cas de figures se présentent (*figure 3.16*) :

- **Chacune des deux faces résultantes de l'éclatement d'une face se trouve sur deux volumes différents :** les faces préservent alors leur label initial (voir les faces côté du volume  $A$  (*figure 3.16*)) ;
- **Chacune des deux faces résultantes de l'éclatement d'une face se retrouve sur le même volume :** les deux faces doivent donc être relabellisées avec un index additionnel (voir la face  $B_{C_4}$  (*figure 3.16*)) ;
- Les faces nouvellement créées reçoivent, pour l'une le label de la face  $F$ , pour l'autre l'opposé de  $F$ . La face opposée à la face  $O$  est la face  $E$  (et réciproquement), la face opposée à la face  $C_i$  est  $C[i+n/2] \bmod [n]$ , où  $n$  est l'ordre du prisme (voir  $A_E$  ou  $C_O$  dans la (*figure 3.16*)).

### 3.2.2.2 Ajout d'un volume sur une face

Cette opération permet de créer un nouveau volume et de le coller sur une face que nous appelons *face support*. Durant cette opération, la face support située sur le volume précédent et la face origine du nouveau volume sont identifiées (gluées). Cette opération est notée :

$A \rightarrow A[B(n)]_F$  où  $F$  est la face support du volume  $A$  et  $n$  est le degré du nouveau volume  $B$ .

Il convient de remarquer que cette opération ne s'applique que si la face  $F$  est libre (elle n'est pas déjà collée avec une autre face). Si la face support et la face origine du nouveau volume ne sont pas isomorphes, alors de nouvelles arêtes sont ajoutées comme indiqué dans la *sous-sous-section 3.2.2.3*. Si le degré du nouveau volume n'est pas spécifié, alors le degré est égal au degré de la face support. La (*figure 3.17*) illustre la différence entre l'opération que nous venons de développer et l'opération d'éclatement, et plus particulièrement quand des volumes adjacents existent.

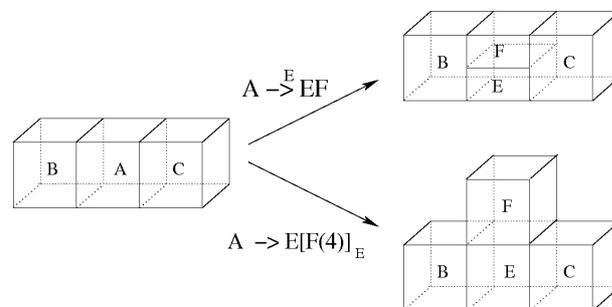


FIG. 3.17 – Différence entre les opérations d'ajout et d'éclatement

### 3.2.2.3 Identification de faces

L'identification de deux faces permet de créer des relations d'adjacences entre deux volumes. Cette opération se déroule comme suit :

- nous commençons par tester que les faces qui doivent être collées ne sont pas déjà collées à une autre face (dans ce cas, notre modèle interdit l'opération) ;
- nous vérifions les degrés des deux faces. Si celles-ci ne sont pas isomorphes, les bords des faces sont modifiés par éclatement successif des arêtes ;
- puis, les faces sont collées.

Pour pouvoir reconstruire les relations d'adjacences, il est nécessaire d'avoir une arête de départ et un sens de rotation sur chaque face. En pratique, nous utilisons cette opération seulement si deux volumes sont déjà adjacents par une arête, ce qui nous donne au moins une arête de départ et le sens de rotation. Ainsi, cette opération a la forme suivante :

$$A : A_{C2} < B_{C4} \rightarrow A_{C2} | B_{C4}.$$

Dans le cas présent, nous identifions les faces  $C2$  des volumes de label  $A$  avec les faces  $C4$  des volumes de label  $B$ , sous réserve que ces faces soient déjà adjacentes par une arête.

### 3.2.2.4 Autres opérations

Pour que notre modèle soit complet, nous avons besoin d'introduire quelques opérations supplémentaires :

- **opérations sur les faces** (éclatement de face) : par exemple, l'éclatement d'une face  $F$  d'un volume  $A$  en deux faces  $F1$  et  $F2$  est notée :  
 $A_F \rightarrow A_{F1}A_{F2}$  ;
- **opérations sur les arêtes** (éclatement d'arête) : ces opérations supposent qu'un label est associé à chaque arête. Dans un tel cas, nous pouvons définir l'opération d'éclatement d'arête avec un formalisme similaire à celui sur l'éclatement de face.

Nous ne redétaillerons pas ces opérations ici, car elles sont déjà décrites dans la *sous-section 2.3.2*. Cependant, il ne faut pas oublier que ces opérations nous permettent d'obtenir une très grande variété de subdivision de volumes.

## 3.3 Interprétation des 3Gmaps L-systems

### 3.3.1 Interprétation géométrique

Dans notre méthode, un volume topologique de type prisme est associé à chaque lettre de notre grammaire. Ces volumes sont ensuite plongés dans  $R^3$ . L'interprétation géométrique d'un volume est décrite dans la *sous-sous-section 2.3.3.1*.

### 3.3.2 Introduction des attributs dans les L-systems

Les attributs topologiques (ordre du prisme :  $O$ ) et géométriques (angles de la tortue :  $atx, aty, atz$ , dimensions du volume  $H, L, W$ ) sont introduits dans les 3Gmaps L-systems comme des paramètres. Plus précisément, l'ordre des paramètres est le suivant :  $A(O, atx, aty, atz, H, L, W)$ .

Dans la pratique, nous pouvons spécifier les valeurs des paramètres directement dans les règles de réécriture ou définir ces valeurs en utilisant l'instruction `#define`. Si les paramètres ne sont pas explicités, alors le système prend des valeurs par défaut (4 pour l'ordre, 0 pour les angles et 1 pour les dimensions). L'évolution du contrôle des paramètres dans le temps est effectuée de manière classique en se basant sur le formalisme utilisé dans `cpfg` et les blocs d'instructions sont écrits en `C`<sup>6</sup>. Nous pouvons introduire des fonctions grâce au mot-clé `#defunc`. L'utilisation de paramètres dans la partie gauche de la règle de réécriture nous permet de réutiliser d'anciennes valeurs pour les opérations (par exemple, voir la règle `p1` dans la *(table 3.1)*). Nous utilisons également une variable globale `stage` qui représente le nombre d'itérations sur les règles de production appliquées.

## 3.4 Premiers résultats

Nous allons illustrer ce système par un premier exemple. L'ensemble de règles décrites dans la *(table 3.1)* (*3Gmap L-system 1*) crée l'enveloppe d'une spirale (*figure 3.18*).

TAB. 3.1 – 3Gmap L-system 1 : Spirale

---

<code>#define B(4, 0., 0., 0., stage/3.+5, 0.4, 1.)</code>
<code>#defunc F1(H) return(H + 0.5);</code>
Axiom : <code>A(16, 0., 0., 0., 5., 5., 5.)</code>
(p1) <code>A(O, x, y, z, H, L, W) → A[A(O, 20., y, z, F1(H), L + 0.5, W + 0.5)]<sub>E</sub></code>
(p2) <code>A → A[B]<sub>C2,C4,C6,C8,C10,C12,C14,C16</sub></code>
(p3) <code>B : B<sub>C1</sub> &lt; B<sub>C3</sub> → B<sub>C1</sub> B<sub>C3</sub></code>

---

Dans le 3Gmap L-system 1, les volumes du centre sont de label A et ceux du bord, de label B. Dans la grammaire :

- nous commençons par définir des outils pour les règles de production (variables, constantes et fonctions) :
  - les propriétés des volumes de type B : ce sont des volumes d'ordre 4 dont la hauteur est proportionnelle à l'itération pendant laquelle ils sont générés (*stage*) ;
  - une fonction (*F1*) qui renvoie un paramètre d'entrée incrémenté d'un demi-pas.

---

<sup>6</sup>Une nouvelle version de cet interpréteur en python est en cours de finalisation. L'utilisation d'un langage interprété permet une plus grande souplesse pour la définition de nouvelles fonctionnalités.

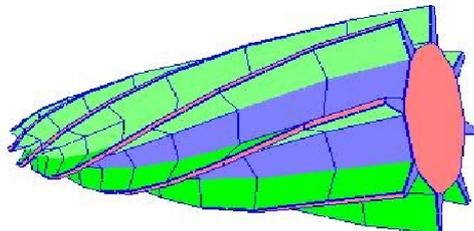


FIG. 3.18 – Spirale générée avec le 3Gmap L-system 1

- ensuite nous définissons notre volume de départ (l'axiome) : ici, un volume de label  $A$  et d'ordre 16 ;
- et finalement nous décrivons les règles de production :
  - (p1) ajoute un volume de label  $A$  à chaque face extrémité ( $E$ ) libre des volumes de label  $A$ . Les dimensions du nouveau volume sont incrémentées de 0.5 à chaque itération <sup>7</sup> ;
  - (p2) ajoute un nouveau volume de label  $B$  à chaque face libre ayant le label recherché ;
  - (p3) colle les faces  $C1$  et  $C3$  des volumes  $B$  si elles ont une relation d'adjacence.

### 3.5 Application à la croissance du bois

Dans cette partie, nous allons présenter une application importante des 3Gmaps L-systems : la simulation de la modélisation de structure de bois en simulant sa croissance. L'idée est d'obtenir une représentation correcte et complète de structure de bois, comme dans le *chapitre 2*, mais de manière plus simple. Avant de présenter les grammaires que nous avons développées, il semble important de reconsidérer le concept d'ordre d'application des règles de réécritures : dans les L-systems classiques, la plupart du temps, les règles de réécritures peuvent être appliquées en parallèle sans avoir d'impact sur le résultat final, ce qui est dû à l'indépendance des opérations entre elles.

Cependant, ici, la plupart des opérations dépendent du contexte (c'est même systématiquement le cas pour les opérations de jointure). Il est ainsi difficile d'anticiper le résultat d'une application parallèle des règles. Ainsi les exemples suivant supposent une application séquentielle des règles, nous plaçant ainsi dans le cadre des grammaires de Chomsky (Chomsky *et al.*, 1958) comme décrit dans le langage étendu *cpfg* (Prusinkiewicz *et al.*, 2001).

Nous commencerons par construire un tronc d'arbre sans branche, puis avec une branche. Pour conclure, nous montrerons comment nos travaux peuvent se connecter avec les L-systems traditionnels.

---

<sup>7</sup>Nous pouvons remarquer que nous ne sommes pas obligé de définir systématiquement des fonctions pour modifier les paramètres, la formule peut être passée directement au constructeur du volume.

### 3.5.1 Tronc sans branche

#### Partie structure

La croissance interne d'un tronc d'arbre sans branche peut être vue comme un empilement de cônes plus ou moins déformés. Dans notre cas, les cônes sont subdivisés en *volumes cubiques*. Une telle subdivision des cônes nous permet d'offrir un contrôle plus fin de l'enveloppe du tronc grâce au contrôle de ces volumes (par exemple voir le bois de réaction (figure 4.18)).

TAB. 3.2 – 3Gmap L-system 2 : tronc sans branche

	Axiom : A(16)	
(p1)	$F \rightarrow F[A]_E$	<i>Les internœuds existant dans l'arbre produisent un nœud</i>
(p2)	$S \rightarrow S[R1(4)]_E$	<i>Les cernes d'été de l'année précédente donnent placent à la production de bois (cernes de printemps)</i>
(p3)	$A \rightarrow A[F]_E$	<i>Le nœud créé en p1 produit un nouvel inter-nœud</i>
	$[R1(4)]_{C^*}$	<i>et il est entouré par un cerne de printemps</i>
(p4)	$F \rightarrow F[R2(3)]_{C^*}$	<i>un inter-nœud est entouré par un cerne de printemps</i>
(p5)	$R1 \rightarrow R[S(4)]_E$	<i>Production des cernes d'été</i>
(p6)	$R2 \rightarrow R[S(3)]_E$	<i>Idem</i>
(p7)	$R : R_{C2} < R_{C4} \rightarrow R_{C2} R_{C4}$	<i>Jointure des cernes</i>
(p8)	$R : R_{C1} < R_{C3} \rightarrow R_{C1} R_{C3}$	<i>Idem</i>
(p9)	$S : S_{C2} < S_{C4} \rightarrow S_{C2} S_{C4}$	<i>Idem</i>
(p10)	$S : S_{C1} < S_{C3} \rightarrow S_{C1} S_{C3}$	<i>Idem</i>

Ainsi, nous pouvons définir la croissance interne d'un tronc d'arbre avec :

- un ensemble de symboles  $\{A, F, R, R1, R2, S\}$  où  $A$  représente les nœuds et  $F$  les inter-nœuds de l'axe central du tronc.  $Ri$  (respectivement  $S$ ) représente les cernes qui apparaissent le printemps (respectivement l'été).  $R2$  sont des prismes d'ordre 3 et qui permettent d'assurer la connexion entre l'année  $y$  et l'année  $y+1$  ;
- un axiome : A(16), un prisme d'ordre 16 qui représente la nervure centrale du tronc ;
- et d'un ensemble de règles de production : voir la (table 3.2) .

Ce 3Gmap L-system est illustré dans la (figure 3.19) et la (figure 3.20) .

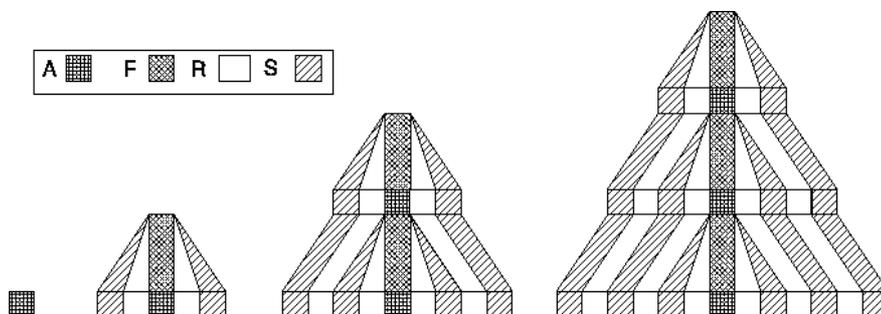


FIG. 3.19 – Tronc sans branche, après les étapes 1,2 et 3

### Contrôle de la géométrie

Le modèle ainsi développé nous permet de conserver le contrôle de la forme du tronc à plusieurs niveaux <sup>8</sup>. Par exemple :

- la silhouette du tronc ;
- la taille moyenne des cernes de printemps et d'été en fonction des espèces ;
- ...

La (table 3.3) nous illustre l'introduction de ces deux contrôles dans notre L-system.

TAB. 3.3 – 3Gmap L-system 3 : Exemple d'insertion du contrôle géométrique

```

#Define iT =3. ; // Slope of the trunk
#Define HN =2. ; LN =2. ; WN =2. ; // dimension des nœuds
#Define HI =10. ; LI =2. ; WI =2. ; // dimension des inter-nœuds
// Extremum de variation de la hauteur des cernes de printemps
#Define tMCP = 1.4 ; tmCP = 1. ;
#Defunc F1() { return alea(tMCP, tmCP) ; }
// Extremum de variation de la hauteur des cernes d'été
#Define tMCE = 0.4 ; tmCE = 0.1 ;
#Defunc F2() return alea(tMCE, tmCE) ;
Axiom : A(16, 0., 0., 0., HN, LN, WN)
(p1) F → F[A(16, 0., iT, 0., HN, LN, WN)]E
(p2) S → S[R1(4, 0., 0., 0., F1(), 1., 1.)]E
(p3) A → A[F(16, 0., iT, 0., HI, LI, WI)]E
      [R1(4, 0., 0., 0., F1(), 1., 1.)]C*
(p4) F → F[R2(3, 0., 0., 0., F1(), 1., 1.)]C*
(p5) R1 → R[S(4, 0., 0., 0., F2(), 1., 1.)]E
(p6) R2 → R[S(3, 0., 0., 0., F2(), 1., 1.)]E
(p7) ... (p10) Joining

```

<sup>8</sup>Cette avantage bien connu des 3Gcartes est repris tout au long du chapitre 2.

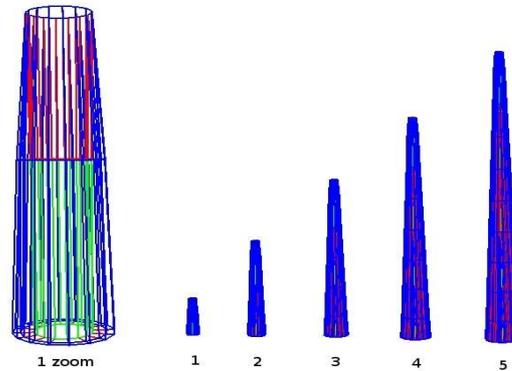


FIG. 3.20 – Tronc sans branche : étape 1 zoom et résultat après les étapes 1,2,3,4. volume de label A : vert, F : rose, R : rouge, S bleu

### 3.5.2 Tronc avec branche

#### Partie structure

Les branches sont relativement simples à ajouter à notre modèle. Cependant, il est nécessaire de prendre en compte l'insertion de la branche sur le tronc, mais également une bonne jonction entre les volumes de la branche et ceux du tronc. Cette cohérence spatiale est gérée par le modèle de carte généralisée utilisé et les volumes joints. La croissance du centre du tronc suit un processus similaire à celui décrit précédemment, excepté qu'il est nécessaire de prendre en compte la création des bourgeons qui sont à l'origine de la branche. Cette étape est obtenue simplement en modifiant la règle 3 du 3Gmap L-system 1. La grammaire du 3Gmap L-system est définie dans la (table 3.4) et illustrée dans les (figure 3.21) , (figure 3.22) et (figure 3.23) .

TAB. 3.4 – 3Gmap-L-system 4 : Tronc avec branche

(p1)	...
(p3)	$A \rightarrow A[F(16)]_E[B(16)]_{C_i}[R1(4)]_{C_{*-i}}$
(p3')	$B \rightarrow B[R1(4)]_{C_*}$
(p4)	...

#### Modification du 3Gmap L-system 3 :

- (p3)  $A \rightarrow A[F(16)]_E[B(16)]_{C_i}[R1(4)]_{C_{*-i}}$  : création d'une branche sur un nœud de l'axe principal. Cette branche aura le même comportement que l'axe principal (en particulier pour la création d'un axe supérieur). Ici la valeur de  $i$  dépend de la phyllotaxie, si au niveau d'un nœud, il y a production de plusieurs branches, la règle prendra la forme suivante :

$$A \rightarrow A[F(16)]_E[B(16)]_{C_i} \dots [B(16)]_{C_k} [R1(4)]_{C_{*-i-\dots-k}}$$

L'espérance de création d'une branche, ainsi que les positions et angles d'insertion

- peuvent être trouvés dans des articles ou des livres traitant de botanique (par exemple (Vigue, 2002) ou (Barry-Lenger *et al.*, 1999)) ;
- (p3')  $B \rightarrow B[R1(4)]_{C^*}$  : le nœud de départ de la branche est entouré par un cerne de printemps.

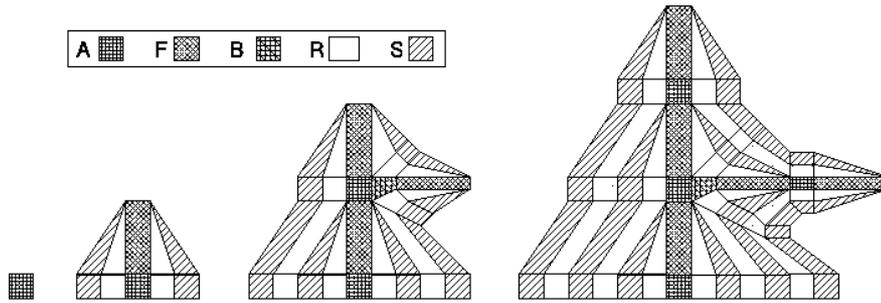


FIG. 3.21 – Tronc avec branche, après les étapes 1,2 et 3

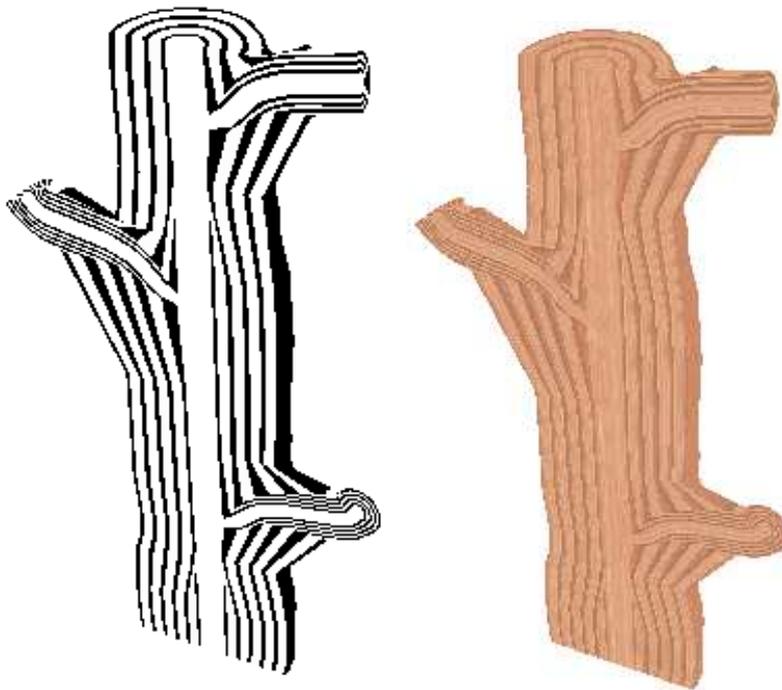


FIG. 3.22 – Tronc avec branches : à gauche le bois d'été en noir et le bois de printemps en blanc ; à droite le bois avec des textures



FIG. 3.23 – Détail de la figure 3.22

### 3.5.3 Relation avec les L-systems *traditionnels*

Comme pour le tronc, nous pouvons contrôler la forme des volumes et les cernes des branches. Pour la branche, nous pouvons, par exemple, contrôler l'angle d'insertion ainsi que la phyllotaxie (fréquence de création, loi de distribution sur la circonférence du tronc). Toutes ces données botaniques peuvent être retrouvées dans des livres spécialisés. De plus, nous pouvons réutiliser les très nombreux travaux déjà effectués sur les L-systems *traditionnels*. Pour illustrer cette relation entre le modèle et les L-systems classiques, nous utiliserons la (table 3.5) donnée à la page 56 de (Prusinkiewicz *et al.*, 1990b). La table. 3.6 montre ce L-system réécrit avec le formalisme des 3Gmaps L-systems.

TAB. 3.5 – L-system 1 : Issu de [PL90], page 56

---

<code>#define r1 0.9 /* ratio de contraction pour le tronc */</code>
<code>#define r2 0.8 /* ration de contraction pour les branches */</code>
<code>#define a0 45 /* angle d'insertion depuis le tronc */</code>
<code>#define a2 45 /* angle d'insertion pour les axes latéraux */</code>
<code>#define d 137.5 /* angle de divergence */</code>
<code>#define wr 0.707 /* taux de diminution de l'épaisseur */</code>
<code><math>\omega</math> : A(1, 10)</code>
<code>p1 : A(l, w) : * <math>\rightarrow</math>!(w)F(l)[&amp;(a0)B(l * r2, w * wr)]/(d)A(l * r1, w * wr)</code>
<code>p2 : B(l, w) : * <math>\rightarrow</math>!(w)F(l)[- (a2)\$C(l * r2, w * wr)]C(l * r1, w * wr)</code>
<code>p3 : C(l, w) : * <math>\rightarrow</math>!(w)F(l)[+ (a2)\$B(l * r2, w * wr)]B(l * r1, w * wr)</code>

---

TAB. 3.6 – 3Gmap-L-system 5 : traduction du Lsystem 1

---

// Les mêmes définitions que L-system 1  
// Définition de la dimension des cernes : voir 3Gmap L-system 3  
Axiom : A(16, 0., 0., 0., 10., 1., 1.)

(p1)  $C(O, x, y, z, H, L, W) \rightarrow F(O, x, y, z, H * r2, L * wr, W * wr)$   
 $[C(O, x, y, z, H * r2, L * wr, W * wr)]_E$

(p2)  $C(O, x, y, z, H, L, W) \rightarrow$   
 $F[B(16, 0., a2, 0., H * r2, L * wr, W * wr)]_{C1}$   
 $[B(16, 0., 0., 0., H * r2, L * wr, W * wr)]_E$

(p3)  $B(O, x, y, z, H, L, W) \rightarrow F(O, x, y, z, H * r2, L * wr, W * wr)$   
 $[B(O, x, y, z, H * r2, L * wr, W * wr)]_E$

(p4)  $B(O, x, y, z, H, L, W) \rightarrow$   
 $F[C(16, 0., -a2, 0., HI, LI, LI)]_{C1}$   
 $[C(16, 0., 0., 0., HN, LN, lN)]_E$

(p5)  $A(O, x, y, z, H, L, W) \rightarrow F(O, x, y, z, H * r1, L * wr, W * wr)$   
 $[A(O, x, y, z, H * r1, L * wr, W * wr)]_E$

(p6)  $A(O, x, y, z, H, L, W) : \{i = 16 * stage * 137, 5/360; \}$   
 $\rightarrow F[B(16, 0., a0, 0., H * r2, L * wr, W * wr)]_{Ci}$   
 $[A(16, 0., 0., 0., H * r1, L * wr, W * wr)]_E$

(p7) ... Ajout des cernes et recollement

---

### 3.6 Conclusion

Ces travaux ont donné lieu à une publication dans la revue *The Visual Computer* (Terraz *et al.*, 2008). Nous venons de présenter une nouvelle catégorie de L-systems qui permettent de générer des subdivisions de volumes. Cette approche permet de contourner certaines limitations des Cellworks L-systems, en particulier :

- grâce à l'utilisation d'opérations qui sont définies directement sur les volumes et à l'ajout d'une gestion automatisée des labels, l'expression des règles de production s'en trouve largement facilitée. Bien que ces simplifications soient faites au détriment d'un modèle plus général (utilisation de prismes orientés), elles permettent de réutiliser simplement les L-systems traditionnels ;
- les conditions sur les volumes adjacents sont faciles à exprimer.

L'application à la génération de structures internes de bois donne des résultats très similaires à la méthode décrite dans le *chapitre 2*, tout en simplifiant énormément la description du processus de croissance. Il serait intéressant d'essayer de connecter ce modèle avec des modèles de croissance d'arbres qui intègrent des données botaniques (proximité des autres arbres, conditions climatiques, comme dans (Mech *et al.*, 1996) ou

(Deussen *et al.*, 1998)... et/ou mécaniques (poids des branches (Hart *et al.*, 2003)....). Cette approche ouvrirait de nouvelles perspectives, plus particulièrement dans l'étude de la qualité du bois en fonction de conditions de croissance.

Pour finir, cette approche a également inspiré un article sur la modélisation et le vieillissement de feuilles (Peyrat *et al.*, 2008).



## Chapitre 4

# Exploitation du modèle

Au cours des précédents chapitres, nous avons vu différentes techniques pour simuler la croissance de l'arbre afin de produire une structure interne de bois en accord avec les règles botaniques de croissance. Nous allons maintenant voir dans quelles mesures nous pouvons exploiter notre modèle.

Dans ce chapitre, nous commencerons par développer la première technique de rendu utilisée pour améliorer sensiblement l'aspect visuel de notre modèle. Nous illustrerons cette méthode en proposant des essences de bois variées, mais également des défauts de structure que notre modèle permet de modéliser. Puis, nous montrerons comment notre modèle permet de générer des textures de bois plus réalistes afin de pouvoir les utiliser dans des logiciels spécialisés dans le rendu. Nous finirons par voir comment la structure particulière du modèle nous permet de simuler quelques aspects du vieillissement du bois.

### 4.1 Rendu primaire

Dans cette section, nous présenterons la technique que nous avons développée pour appliquer des textures sur notre modèle. Bien sûr, cette technique ne constitue qu'un rendu partiel car l'éclairage n'est pas géré complètement. Nous proposons une amélioration de ce modèle dans la *section 4.2*.

#### 4.1.1 Texturage du modèle

Pour pouvoir mieux valider visuellement notre modèle, nous avons décidé de texturer le bois modélisé.

Nous avons donc développé une technique d'amélioration visuelle du modèle qui s'appuie très largement sur notre méthode de modélisation. Cette technique se base sur une approche de type texture 3D.

Dans cette technique, nous distinguons les deux composantes principales du bois : « le bois de printemps » et « le bois d'été » avec différentes propriétés visuelles, que nous détaillerons plus loin dans cette partie.

Cette distinction est imposée par la méthode de modélisation.

La première étape consiste à obtenir un petit élément de texture 3D correspondant à une composante du bois (l'information est donnée par le modèle lui-même<sup>1</sup>) à partir d'une photo de bois. Nous proposons d'extraire des éléments de texture 2D à partir de vraies photographies suivant deux règles (figure 4.1) :

- aucun éclairage spéculaire ne doit être présent : cela signifie que la photographie doit montrer seulement un comportement lambertien (le bois poli ou quelques parties du bois brut peuvent se comporter de manière plus ou moins spéculaire) ;
- la photographie doit montrer les deux constituants spécifiques du bois, et si nécessaire, l'aspect granuleux du bois (présence de fibres de grosse taille).

Les éléments de texture 3D sont générés en extrudant les éléments de texture 2D ainsi obtenus. La petite taille des modèles 2D extraits et l'aspect « bruité » induit par la microstructure du bois nous permettent d'être sûr d'éviter des discontinuités visibles entre deux textures de volumes voisins. Qui plus est, les déformations des unités de texture 3D (incluant les coordonnées de texture et leur adaptation aux volumes) pendant le rendu permettent de fortement limiter les inconvénients induits par le processus d'extrusion (voir figure 4.1).

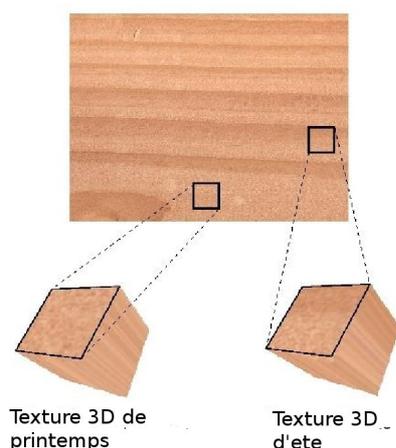


FIG. 4.1 – Extraction et extrusion de deux modèles 2D naturels

L'application des éléments de texture 3D au modèle géométrique est triviale dans notre cas. Chaque volume est associé à un des deux éléments de texture 3D. Nous associons des coordonnées de texture 3D à chaque brin, permettant ainsi de lier l'espace géométrique 3D avec les espaces des deux petits éléments de texture 3D. Cette opération est effectuée durant la génération du volume :

- les coordonnées de texture d'un brin sont calculées classiquement en fonction du type de la face et des faces voisines : nous associons à chaque brin une position sur le bord du bloc de texture 3D correspondant (bois de printemps ou d'été). Cela conduit à des déformations de texture 3D, permettant ainsi d'éviter les régularités dues à la phase d'extrusion (voir le paragraphe suivant) ;

<sup>1</sup>En effet, comme nous l'avons vu dans la sous-section 2.3.1, chaque volume possède un marqueur pour connaître sa saison de création.

- les coordonnées de texture des brins sont réajustées dans la phase de modification topologique du volume : un volume est modifié par exemple durant l'éclatement (arête, face et volume) ou la coupe. Les nouveaux brins générés reçoivent des nouvelles coordonnées de texture interpolées à partir du rapport entre les deux points extrémité de l'arête éclatée et le nouveau point généré.

L'utilisation de deux textures 3D différentes au lieu d'un simple plaquage de texture permet d'employer une structure dans laquelle nous pouvons facilement couper suivant la direction voulue par l'utilisateur. L'extrusion de modèles 2D naturels est suffisante dans notre rendu de bois temps réel, car :

- les déformations des volumes durant le processus de modélisation sont toutes différentes et appliquées sur des volumes élémentaires de couleur déjà bruités ;
- les unités de texture 3D sont petites par rapport à l'image finale rendue.

La direction privilégiée de coupe, toujours induite par l'extrusion convient bien à la structure de la fibre du bois. De plus, le modèle 3D crée des volumes orientés permettant de donner à nos textures 3D la même orientation autour du tronc. La *(figure 4.2)* montre une vue de côté de l'intérieur du bois avec les déformations des volumes modélisés ainsi que les deux textures 3D différentes appliquées.

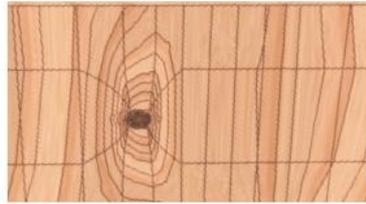


FIG. 4.2 – Image de bois synthétisée montrant les déformations des volumes

Afin d'augmenter le réalisme de nos images nous devons tenir compte de l'influence des variations de densité du bois sur l'éclairage ((Marschner *et al.*, 2005) et (Mérillou *et al.*, 2000)). Nous utilisons une technique similaire à celle utilisée dans (Mérillou *et al.*, 2000). Le principe est de prendre en compte analytiquement l'influence des pores (modélisés sous la forme de cylindres perturbés) dans la micro-géométrie des surfaces. Employer un tel modèle d'éclairage local dans n'importe quel moteur de rendu réaliste est aisé. Cependant, afin de tenir compte de ces phénomènes en temps réel (pour différencier facilement les différentes sortes de bois), nous proposons d'extraire les principes phénoménologiques de l'éclairage de surfaces poreuses. Ces principes sont utilisés sous la forme de techniques empiriques pour accroître le réalisme de nos images. La présence de pores induit les comportements suivants :

- quand la porosité augmente, la densité diminue et le comportement diffus augmente grâce aux multiples réflexions de la lumière à l'intérieur des pores (la specularité décroît) ;
- quand la porosité diminue, la densité augmente et le comportement diffus diminue (la specularité augmente).

Nous adaptons tout simplement ce comportement en employant le canal alpha

d'OpenGL comme une carte spéculaire. Dans les régions de forte densité, un fort niveau de gris est appliqué (bois d'été : moins de pluie ce qui implique une plus grande densité) ; nous gardons la texture originelle dans les régions de faible densité. Cette technique empirique permet d'obtenir des résultats de rendu en temps réel suffisamment plausibles pour valider notre modèle (*figure 4.4*). La (*figure 4.3*) montre une comparaison entre une photographie réelle et les résultats obtenus en combinant tous ces éléments. Le but de cette comparaison n'est pas d'obtenir des images entièrement identiques mais d'évaluer le réalisme de nos images.

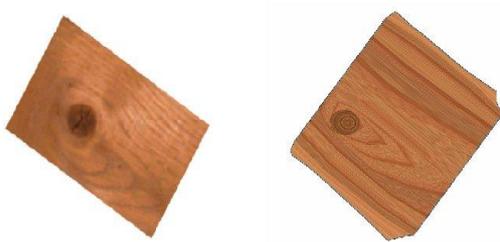


FIG. 4.3 – Comparaison entre une photo réelle (gauche) et une image synthétisée (droite)



FIG. 4.4 – Morceau de bois rendu en temps réel

En fournissant une structure interne de bois plus réaliste, notre méthode nous permet d'améliorer sensiblement le rendu du bois. Cependant, nous verrons dans la *section 4.2* que ce rendu peut encore être nettement amélioré en utilisant des moteurs de rendu beaucoup plus complets.

Après avoir présenté notre modèle de texturage, nous pouvons maintenant illustrer notre technique par des exemples d'essences de bois.

#### 4.1.2 Essence de bois

Grâce à l'utilisation de la technique d'application de texture présentée dans la *sous-section 4.1.1*, nous pouvons mieux visualiser les améliorations apportées par le modèle.

Pour valider nos résultats, nous avons choisi trois espèces différentes : le pin, le mélèze et l'orme. Il est intéressant de comprendre que notre modèle n'est contrôlé que par très peu de paramètres :

- un interval de variation de l'épaisseur des cernes de printemps et d'été qui dépend de l'âge de l'arbre ;
- la fréquence d'apparition des branches, la distribution de ces branches sur la circonférence du tronc et l'angle d'insertion des branches ;
- les textures des cernes pour le printemps et l'été, spécifiques à chaque essence.

Toutes ces données (taille des cernes, couleur du bois...) peuvent être trouvées dans des ouvrages spécialisés tels que (Vigue, 2002) et (Barry-Lenger *et al.*, 1999). Afin d'accélérer les tests, nous avons souvent restreint les processus de croissance à un nombre limité de

volumes sur les axes principaux. Nous avons aussi limité le nombre de branches. Pour de telles images, le temps de génération des subdivisions 3D est d'environ 5 secondes sur un pentiumIV 2.66 GHz. La coupe est effectuée en temps réel. Sur la (figure 4.5), nous pouvons voir une modélisation de pin : le plan de coupe dans le tronc est localisé sur la (figure 4.5) a, alors que la (figure 4.5) b est une photographie réelle de pin et que la (figure 4.5) c montre le résultat de la coupe. Sur la (figure 4.6), nous pouvons apercevoir trois nœuds dans une planche de pin.

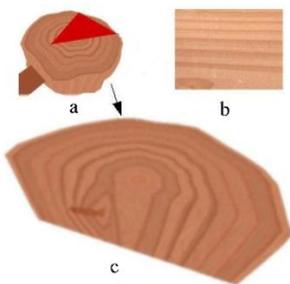


FIG. 4.5 – Pin. a : plan de coupe dans le tronc ; b : photographie réelle de pin ; c : résultat de la coupe

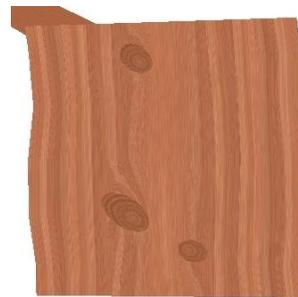


FIG. 4.6 – Planche de pin avec 3 nœuds

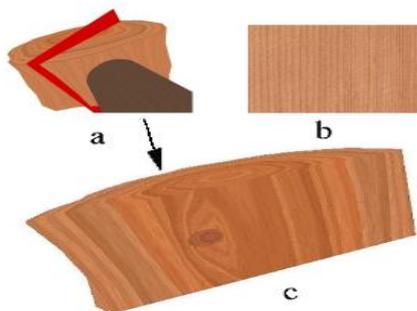


FIG. 4.7 – Mélèze. a : plan de coupe dans le tronc ; b : photographie réelle de mélèze ; c : résultat de la coupe

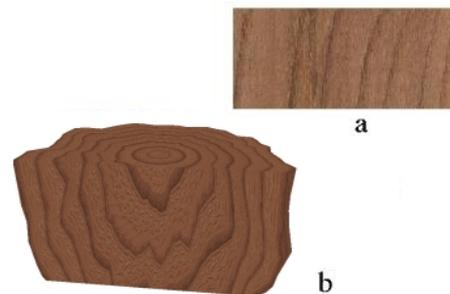


FIG. 4.8 – Orme. a : photographie réelle ; b : résultat de la coupe

La (figure 4.7) représente une modélisation de mélèze : la (figure 4.7) a permet de localiser le plan de coupe dans le tronc, la (figure 4.7) b est une photographie réelle et la (figure 4.7) c montre le résultat obtenu après la coupe. Sur la (figure 4.8) a, nous pouvons voir une photographie réelle d'orme et sur la (figure 4.8) b, le résultat d'une

coupe qui montre que l'orme peut avoir un fil assez ondulé. Cependant notre modèle, dans sa forme actuelle, n'est pas adapté à des déformations trop importantes des cernes.

Comme nous venons de le voir, nous obtenons une bonne diversification des espèces de bois.

## 4.2 Génération de texture

### 4.2.1 Positionnement du modèle

Une texture en synthèse d'image peut être définie comme une image quelconque collée sur une surface plane ou gauche, comme par exemple un papier peint sur un mur. Les premières textures utilisées en synthèse d'images ont été introduites par (Catmull, 1974) et (Blinn *et al.*, 1976). Catmull propose un simple plaquage de textures naturelles (images, photos) sur des surfaces, tandis que Blinn et Newel proposent de générer des textures synthétiques.

Si nous nous basons sur la dimension des textures manipulées, deux grandes catégories se distinguent :

- **les textures planes conventionnelles**, appelées également textures 2D, qui sont définies dans un espace à deux dimensions (ce sont les premières textures à avoir été utilisées) ;
- **les textures volumiques**, appelées aussi textures 3D, qui sont définies dans un espace à trois dimensions.

Générer une texture revient à calculer ses variations de couleurs dans le plan (pour les textures 2D) ou dans l'espace (pour les textures 3D). En utilisant une définition plus mathématique, cela revient à définir une fonction texture appelée souvent  $T$ , qui à tout point du plan (respectivement de l'espace) de coordonnées  $(x,y)$  (resp.  $(x,y,z)$ ) associe un point du plan (respectivement de l'espace) de texture, ce qui donne une couleur. Exemple :  $C = T(x,y)$  où  $C$  représente la couleur.

Dans le modèle que nous proposons, le but étant d'appliquer ces textures sur des volumes, nous ne nous attarderons pas sur les textures 2D qui présentent plusieurs défauts majeurs :

- **le problème de la fonction de transformation** : il n'est pas toujours évident de trouver une bonne fonction qui à tout point de l'objet associe une couleur. Ceci peut introduire des déformations de la texture appliquée, comme par exemple sur la sphère (*figure 4.9*) où la texture à appliquer est de type damier ;
- **le problème de discontinuité** : dans certaines régions de l'objet, comme par exemple pour le cube, une discontinuité apparaît sur les arêtes (*figure 4.9*) .

En ce qui concerne les textures de bois en 2D, le lecteur pourra se référer à (Mandelbrot, 1983), (Fournier *et al.*, 1982), (Gardner, 1985), (Dischler *et al.*, 1995). La solution générale pour éviter ces problèmes de déformation et de discontinuité, est l'utilisation de textures 3D.

Les textures 3D ou textures solides ont été introduites en 1984 par Gardner, puis par Peachey et Perlin en 1985. L'idée principale est de se donner directement une fonction

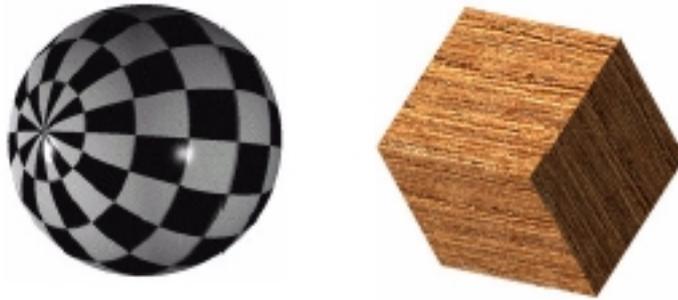


FIG. 4.9 – Problèmes de discontinuité et de déformations liées au placage de textures planes

texture dans l'espace 3D, d'où une utilisation très simple par rapport aux textures 2D. En effet, appliquer une texture 3D revient à ne faire qu'un simple changement de base. Cependant, contrairement à la génération de texture en 2D, il est presque impossible de numériser des échantillons de texture directement en 3D, d'où la nécessité de synthétiser ces textures, ce qui est le principal inconvénient de cette méthode.

Dans la suite de cette section, nous ne proposerons pas volontairement un panorama exhaustif des méthodes de génération de texture 3D (le lecteur trouvera une étude approfondie dans (Dischler *et al.*, 2001)). Nous présenterons plus particulièrement les méthodes qui peuvent avoir pour application la génération de texture de bois.

### Texture volumique procédurale expérimentale

Générer une texture volumique procédurale consiste à calculer ses variations de couleurs dans l'espace. Cela revient à définir une fonction qui à tout triplet de coordonnées de l'espace associe une couleur.

Exemple :  $C=T(x,y,z)$  où  $C$  est la couleur de coordonnées  $(x,y,z)$  et  $T$  la fonction de texture.

Cette utilisation procédurale permet d'éviter le plaquage de textures comme en 2D. Elle revient en fait à tailler l'objet dans la texture au lieu de le recouvrir. Cependant la fonction de texture  $T$  reste toujours difficile à déterminer. Plusieurs approches ont été développées, nous allons en détailler quelques unes.

L'idée générale de ces méthodes est de rendre un aspect naturel par manque de régularité et par introduction d'un grand nombre de détails distribués aléatoirement.

Une première approche consiste à utiliser des fonctions périodiques pour représenter les surfaces naturelles. La fonction de texture est constituée d'une sommation de sinusoïdes d'amplitudes, de fréquences et de phases variées ainsi que de fonctions de déphasage. Les différents travaux ( (Gardner, 1984) et (Gardner, 1985) par exemple) qui s'inspirent de cette idée ont permis de modéliser efficacement des arbres, des montagnes, des nuages et d'autres textures périodiques. Les résultats sont réalistes, mais cette méthode présente un défaut majeur : du fait de la complexité de la formule, la

détermination des paramètres est souvent délicate pour obtenir un type précis de structure. Peachey (Peachey, 1985) propose une autre approche en utilisant des combinaisons de projections pour générer ses textures à partir de textures 2D, mais les défauts sont identiques à la méthode citée ci-dessus.

Une autre approche consiste à utiliser des fonctions aléatoires pour générer des textures ayant une certaine irrégularité et beaucoup de détails distribués de façon aléatoire comme le marbre ou le bois par exemple. Le point de départ de ces méthodes est l'utilisation d'une fonction de base T (i.e. une fonction représentant de manière simplifiée la texture à produire) qui est perturbée par une ou plusieurs fonctions aléatoires pour lui donner un aspect moins régulier et donc plus naturel. Par exemple, Perlin (Perlin, 1985), utilise des fonctions de bruit et turbulence où le bruit est une fonction  $b(x,y,z)$  continue, dérivable et invariante par isométrie retournant une valeur pseudo-aléatoire et la turbulence est une sommation de bruits sur différentes fréquences.

Exemple : Génération de texture de bois,  $T(x,y,z)=\sin(\sqrt{x^2 + y^2} + \text{bruit}(x,y,z))$ .

Le défaut de cette méthode est analogue à celui des approches citées précédemment : comment déterminer les paramètres des fonctions de texture, de bruit, et de turbulence pour obtenir un type précis de texture ?

Les différentes approches empiriques qui viennent d'être présentées nécessitent des recherches expérimentales afin de déterminer les différents paramètres des fonctions. Par conséquent, les modèles des textures 3D générés sont très hasardeux.

### Textures naturelles : méthodes de génération par évolution

Ces méthodes utilisent les théories de la biologie ou de la physique pour reproduire des phénomènes existants. Ces méthodes introduites par (Sims, 1991) utilisent des théories biologiques comme la sélection, la mutation, ou la reproduction pour générer des textures à partir d'autres textures définies précédemment. Les textures de type procédural sont représentées sous forme d'un arbre où les opérations utilisées (+, \*, sin, cos, bruit, ...) sont les nœuds et les opérandes les feuilles. La création de nouvelles textures se fait à partir de mutations aléatoires ou par croisement entre deux textures initiales (*figure 4.10*)<sup>2</sup>.

Nous retrouvons le même défaut : l'absence de nœuds dans les textures. Il existe d'autres approches, mais celles-ci ne donnant pas de texture de bois (textures cellulaires (Fleischer *et al.*, 1995),...), nous ne les détaillerons pas.

### Textures analytiques

Ce sont des méthodes qui permettent de générer des textures volumiques le plus souvent à partir de photographies. Elles permettent également d'éviter l'aspect expérimental des autres méthodes. Les principales méthodes sont l'approche spectrale ((Dischler *et al.*, 1995) et (Ghazanfarpour *et al.*, 1996)) qui concerne plutôt les textures ayant une

---

<sup>2</sup>Images extraites de (Sims, 1991).



FIG. 4.10 – Exemples de textures naturelles : marbre et bois

certaine périodicité, et l'approche pyramidale (Heeger *et al.*, 1995) pour des textures plutôt aléatoires.

Le principe des méthodes de synthèse de textures solides est d'aider l'utilisateur à décrire autrement le motif de l'espace 3D en fournissant une ou plusieurs images 2D, ce qui revient à faire une extension de l'analyse spectrale.

Principe de la méthode de base :

- fournir une texture (modèle) naturelle 2D (par exemple une image numérisée) du même type que la texture 3D à générer ;
- calculer la transformée de Fourier de cette image et faire une sommation de cosinus comme dans le cas des textures 2D ;
- utiliser comme fonction de base simplement cette sommation de cosinus et la perturber avec une fonction de perturbation.

Exemple :  $T(x,y,z)=f(x+p(x,y,z),y+p(x,y,z),z+p(x,y,z))$  où  $p$  est la fonction de perturbation.

Comme pour les méthodes expérimentales, le choix de la fonction de perturbation est délicat. Choisir un bruit blanc ou une turbulence pour perturber ne conduit pas toujours à de bons résultats car ces signaux sont indépendants du motif et le détériorent. La solution est de filtrer le bruit blanc pour l'adapter au moins selon les axes  $X$  et  $Y$  sur lesquels l'utilisateur dispose d'informations. Le motif ainsi obtenu est contrôlé pour les axes  $X$  et  $Y$  mais reste incontrôlé le long de  $Z$  (*figure 4.11*) .

Ces techniques comportent plusieurs avantages :

- **une simplicité d'utilisation** : il n'y a pas de nécessité de décrire la texture mathématiquement par une recherche au hasard (le résultat est obtenu automatiquement) ;
- **des textures de très bonne qualité** pourvu que la texture naturelle soit suffisamment régulière.

Cependant, nous pouvons noter une forte baisse de la qualité de la texture avec des

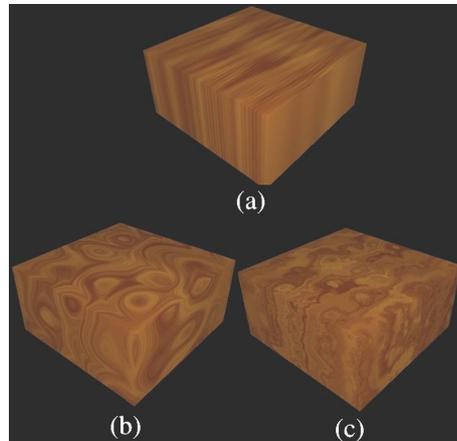


FIG. 4.11 – Exemple de génération de textures avec différentes perturbations et bruits

motifs très aléatoires ou très bruyants. De plus, cette technique ne permet pas de générer par exemple des nœuds ou des singularités dans le cas de texture d'arbre. A ma connaissance, la plupart des textures géométriques (hypertextures (Perlin *et al.*, 1989) et (Dischler *et al.*, 1995), texels (Kajiya *et al.*, 1989) et (Dischler *et al.*, 1995), et textures cellulaires (Fleischer *et al.*, 1995)) n'ont pas pour application de simuler des textures de bois. Nous détaillerons donc une autre approche, la simulation de texture de bois par insertion de macro-structures.

### Textures solides avec macro-structures

L'idée est d'améliorer le rendu d'une texture 3D en incorporant des macro-structures sur la texture (Dischler *et al.*, 1999) pour enrichir la texture. L'insertion d'une macro-structure déforme localement la texture (on ajoute une distorsion de la texture sur la zone d'influence de la macro-structure) pour un meilleur rendu. La macro-structure est souvent une photo (d'un nœud ici) qui est numérisée et traitée avant d'être appliqué sur une texture (*figure 4.12*).

Le rendu est ainsi amélioré. Cependant il manque encore certains détails. Par exemple, la répartition des cernes autour des nœuds (particulièrement des petits) ne paraît pas très naturelle. Nous pouvons remarquer également que la distribution des nœuds elle-même ne semble pas naturelle.

En conclusion, nous pouvons affirmer que les différentes méthodes de génération de textures que nous venons de voir présentent toutes de manière plus ou moins importante les mêmes défauts : les textures ne respectent pas la structure interne d'un arbre *sous-section 1.2.2*. Une solution serait de générer directement l'objet naturel, par exemple la structure interne de l'arbre, et ensuite de l'utiliser comme une texture.

Tous les travaux mentionnés ci-dessus ne concernent qu'une simulation de la structure du bois, grâce à un processus de croissance. Il existe d'autres méthodes qui se basent sur une approche différente. Leur objectif global n'est pas la simulation de bois à travers un processus de croissance, mais seulement la génération de textures de bois

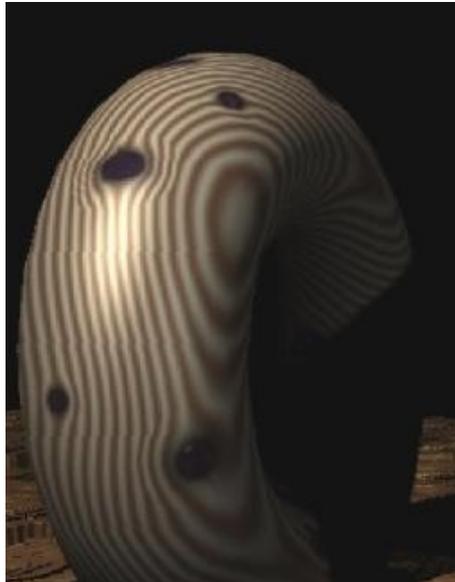


FIG. 4.12 – Insertion de macrostructure, ici un nœud

visuellement réalistes ((Dischler *et al.*, 1995), (Ghazanfarpour *et al.*, 1996)). Concernant l'aspect des nœuds, une approche utilisant des textures macro-structurées a été proposée dans (Dischler *et al.*, 1999). Les nœuds sont inclus comme des modèles spécifiques (macro-structures) dans la texture de bois qui est déformée autour de ces inclusions. Cependant, le manque de cohérence spatiale dans la répartition des nœuds conduit à un aspect artificiel des cernes autour des nœuds.

## 4.2.2 Modèle de génération de texture

### 4.2.2.1 Introduction

Afin de mieux nous rendre compte des possibilités de notre modèle, nous avons décidé de l'exporter vers un logiciel qui nous a semblé mieux adapté pour effectuer le rendu. Nous avons hésité entre développer notre propre moteur de rendu et utiliser un moteur déjà existant. Cependant, comme nous ne connaissions pas encore toutes les propriétés du modèle que nous pourrions exploiter, nous avons donc décidé de ne pas développer notre propre moteur de rendu, mais de passer par un modèle déjà existant.

Nous avons développé deux types de modèle d'exportation :

- le premier consiste simplement à exporter la super-texture générée par notre modèle (c'est celle que nous détaillerons dans cette section) ;
- comme nous le verrons dans la *section 4.4*, pour les fentes, nous avons dû développer un autre modèle où la géométrie de l'objet est également exportée.

#### 4.2.2.2 Blender

Il existe de très nombreux logiciels de modélisation et de rendu. Cependant, pour des raisons techniques, et pratiques, nous avons choisi d'utiliser le logiciel Blender car :

- c'est une suite libre et gratuite d'animation, modélisation et rendu 3D performante ;
- il possède une documentation très fournie ;
- les forums de discussions sur Blender sont plutôt nombreux et très réactifs ;
- pour la création de scripts et de plugins, il existe de nombreux sites français d'excellente qualité. <sup>3</sup> ;

Cependant, comme nous le verrons dans la suite de cette partie, le format d'échange que nous avons utilisé est classique, et donc le porter sur un autre logiciel ne devrait pas poser de problèmes particuliers.

L'un des principaux atouts de Blender, pour notre cas, est de posséder un système de greffons (plugins) en Python (il est également possible d'écrire ces plugins en C).

#### 4.2.2.3 Modèle développé

Pour pouvoir générer notre texture, nous proposons l'approche suivante (d'autres approches peuvent être envisagées, mais cette dernière a l'avantage d'être relativement simple) : le bloc de texture 3D de bois est formé par l'accumulation de captures de coupes successives.

Par exemple, pour une texture voulue de résolution  $L*P*H$ , nous opérons  $L$  coupes radiales avec un pas de  $L$ /longueur de la boîte englobante de la grume de bois. Pour chaque coupe, nous calculons l'intersection entre la grume de bois et le plan de coupe courant. Le résultat est affiché avec une résolution  $P*H$ , puis est stocké dans le bloc matriciel.

Cette matrice est ensuite récupérée sous blender pour être appliquée comme un bloc de texture 3D. Pour calculer la coordonnée texture d'un point de l'espace, nous plongeons le point dans le bloc texture. Si le point ne tombe pas exactement sur un pixel de la matrice, nous l'interpolons avec les 4 plus proches voisins dans la matrice.

#### 4.2.3 Résultats

La première illustration est une planche légèrement biseautée dans un coin sur laquelle nous avons appliqué une texture 3D de bois générée à partir de la technique décrite dans la *sous-sous-section 4.2.2* (*figure 4.13*) . Nous pouvons apercevoir le centre de la grume ainsi que deux branches qui traversent de part et d'autre le bloc de texture.

La deuxième illustration est une horloge en bois (*figure 4.14*) . Nous pouvons voir sur l'image suivante un détail de la scène précédente (*figure 4.15*) . L'aspect fibreux est bien rendu.

La dernière illustration est une texture de bois avec des noeuds appliquée sur un tore (*figure 4.16*) .

---

<sup>3</sup>Par exemple <http://jmsoler.free.fr/didacticiel/blender/tutor/index.htm>

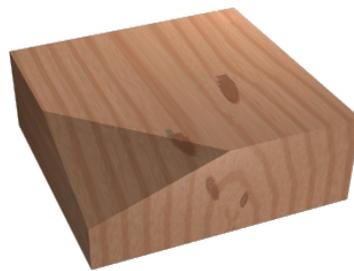


FIG. 4.13 – Planche de bois rendue avec le logiciel Blender



FIG. 4.14 – Horloge générée avec une texture de bois sans nœud



FIG. 4.15 – Zoom sur un détail de l'horloge



FIG. 4.16 – Tore en bois

D'après les illustrations précédentes, nous pouvons en conclure que le modèle est plus performant qu'un simple affichage sous OpenGL. Cependant, pour obtenir des bonnes images, il est nécessaire de générer des textures3D avec une grosse résolution, ce qui implique une occupation mémoire assez importante (de l'ordre de 700Mo à 2Go). Pour pallier à ce défaut, il pourrait être plus judicieux d'incorporer le modèle de croissance au complet sous forme de plugin dans blender.

### 4.3 Défauts du bois

Nous avons également travaillé sur la modélisation de quelques phénomènes particuliers : duramen (bois de coeur), bois de réaction, courbure des branches et rayons médullaires.

#### 4.3.1 Duramen

La sève ne passe plus dans le duramen, ce qui a pour conséquence une modification de sa couleur. Pour le chêne, le bois de coeur prend une couleur blanchâtre, pour les conifères la couleur est, au contraire, plus foncée. La (figure 4.17) montre ce résultat pour du pin.



FIG. 4.17 – Duramen ou bois de coeur



FIG. 4.18 – Bois de réaction

### 4.3.2 Bois de réaction

Le bois de réaction est produit lors de la croissance d'un tronc non-rectiligne (l'arbre croit sur un sol incliné ou il est gêné par un autre). Ceci a comme conséquence une plus grande production en bois dans certaines zones pour rectifier le tronc. Les cernes sont ainsi plus épais d'un côté du tronc que de l'autre. Ce phénomène est aisé à modéliser avec notre méthode : nous devons seulement modifier la taille des éléments des axes radiaux du tronc dans une direction. La *(figure 4.18)* montre une modélisation du bois de réaction.

### 4.3.3 Rayons médullaires

Les rayons médullaires ou ligneux sont des lames de tissus cellulaires qui relient la moëlle (centre de l'arbre) et l'écorce. Ces rayons recoupent les cernes en un grand nombre de secteurs. Ils constituent des éléments de faible résistance qui favorisent les fentes de séchage et la fente du bois. Ces rayons sont plus ou moins visibles en fonction des espèces : chez tous les résineux ils sont invisibles car très petits, tandis que dans d'autres bois ils sont visibles car beaucoup plus gros, c'est le cas chez le chêne. Actuellement, notre modèle ne nous permet pas de simuler des rayons médullaires botaniquement réalistes, car nous intégrons ces rayons directement à nos blocs de textures 3D. Cependant, nous obtenons des résultats visuellement acceptables *(figure 4.19)*.

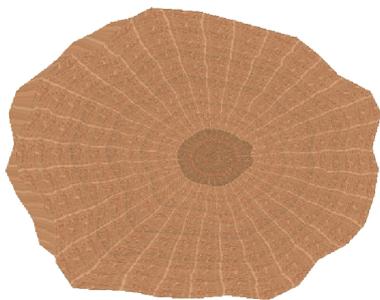


FIG. 4.19 – Chêne, rayons médullaires



FIG. 4.20 – Détail de la courbure d'une branche

### 4.3.4 Noeuds

La morphologie d'une branche (courbure en particulier) dépend de son espèce (propriétés mécaniques) et des conditions de vie de celle-ci (climat, environnement...). Avec la croissance de l'arbre, cette déformation locale de la branche, ou courbure, se retrouve dans le tronc *(figure 4.20)*. Sur cette figure, nous pouvons distinguer les cernes de la branche et les déformations locales occasionnées par la courbure de celle-ci sur le tronc, en particulier le long de la jonction entre la branche et le tronc. Pour simuler de tels phénomènes, nous modifions tout simplement le nombre et les angles d'insertion des

différents volumes cylindriques constituant le cœur de la branche, et grâce au modèle de plongement développé, ces modifications se propagent naturellement le long des cernes.

La (figure 4.2) montre une vue de côté de l'intérieur du bois. Nous pouvons, en particulier apercevoir les déformations des volumes modélisés autour de l'axe de la branche.

## 4.4 Vieillessement

Dans cette partie, nous nous intéresserons à quelques avantages que peut apporter notre modèle sur la simulation de phénomènes qui apparaissent avec le temps.

Le bois est un matériau vivant. Ainsi, il peut subir des modifications lorsqu'il est exposé à des contraintes (usure, altération des couleurs, fissures, ...). Dans notre étude, nous distinguerons deux types de contraintes (bien que d'autres critères de découpages puissent être définis) :

- le premier est lié aux modifications que peut subir la structure en fonction des conditions d'utilisation. Ces défauts sont principalement des déformations de la pièce de bois étudiée (fissure, fracture, gauchissement, ...). Nous appellerons ces modifications des déformations de structure ;
- le second est plutôt lié aux modifications de l'aspect visuel du bois (ternissement du bois au contact du soleil, ...). Nous appellerons ces modifications des déformations d'aspect.

Naturellement ces phénomènes sont liés. Par exemple, lorsqu'une pièce de bois sèche, de l'eau s'évapore, ce qui se traduit par un ternissement sensible des couleurs et une rétraction structurelle du bois. Cependant, ces phénomènes peuvent être expliqués indépendamment les uns des autres.

### 4.4.1 Simulation de fissures et de fractures

Dans cette partie, nous allons développer un modèle physique de simulation de fissures et de fractures. De nombreux travaux ont déjà été publiés sur les fractures. O'brian (Iben *et al.*, 2006) utilise une approche basée sur les éléments finis. L'objet est décomposé en un maillage tétraédrale avec des tenseurs de stress. Quand l'effort est trop important, des fractures apparaissent et se propagent. Les valeurs pour les tenseurs de stress sont obtenues à partir de données réelles, mais le matériau est considéré comme homogène. (X.Yin *et al.*, 2004) applique ce modèle sur le bois, mais les auteurs considèrent également le bois comme un matériau homogène. (Gobron *et al.*, 2000) simule la propagation de craquelures sur la surface d'objets avec un automate cellulaire de dimension 3. (Lefebvre *et al.*, 2002) utilise un système masse-ressort pour simuler des craquelures sur l'écorce des troncs.

Cependant, même si les images sont réalistes d'un point de vue visuel, toutes les méthodes que nous venons de voir se basent sur le fait que le matériau est homogène. Ainsi les simulations de craquelures et de fractures sur le bois n'utilisent pas les propriétés hétérogènes du bois.

C'est dans ce contexte que nous proposons une approche originale, basée sur notre structure interne d'arbre (développée dans les premiers chapitres). Cependant, bien que notre méthode ne soit pas basée sur des mesures physiques comme pour les éléments finis avec des tenseurs de stress, elle suit un chemin naturel en accord avec des rapports botaniques.

#### 4.4.1.1 Considérations botaniques

Le bois est un matériau hétérogène, hygroscopique et anisotropique. Le bois est composé jusqu'à 90% de fibres. Ces fibres ont, pour la plupart, une forme allongée, et sont situées dans le sens longitudinal de l'arbre. Elles ont la forme d'une paille (cellules longues de 3-4mm, et on les appelle trachéides longitudinales (*figure 4.21*)). Les trachéides donnent aux arbres leur support structurel et servent de conduit pour l'acheminement vertical de l'eau et des nutriments (lorsqu'elles ne sont pas mortes).

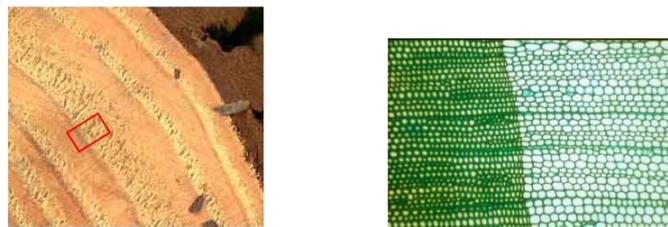


FIG. 4.21 – Vue macroscopique (à gauche) et microscopique (à droite) d'une coupe transversale d'un morceau de bois

Le bois vivant, ou bois vert contient une grosse proportion d'eau : l'eau représente entre 40 et 45% de la masse totale du bois vert, la partie restante correspondant au bois sec (les cellules déshydratées).

Lorsque la fibre absorbe de l'eau, celle-ci est d'abord retenue par les parois cellulaires proprement dites. Lorsque les parois cellulaires regorgent d'eau, toute l'eau additionnelle absorbée par le bois vient remplir les cavités de ces cellules tubulaires. La saturation de la fibre est le degré d'humidité où les cellules retiennent autant d'eau qu'elles le peuvent. L'eau retenu dans la paroi cellulaire est appelée eau liée et l'eau dans les cavités cellulaires eau libre. Le point de saturation de la fibre représente la limite de retrait du bois. En effet, le bois rétrécit ou gonfle selon les changements de sa teneur en humidité, mais seulement à mesure que l'eau dans les parois cellulaires est absorbée ou libérée. Les changements de teneur en humidité dans la cavité cellulaire n'auront pas d'effet sur le bois. Par conséquent, le bois ne rétrécit ou ne gonfle qu'au moment où la teneur en humidité descend sous le point de saturation de la fibre.

D'un point de vue technique il existe 3 types d'eaux dans le bois :

- **l'eau de constitution** : c'est l'eau qui entre dans la composition chimique des molécules du bois. Son élimination entraîne la destruction du bois (par le feu par exemple) ;

- **l'eau liée** : c'est l'eau qui entre dans la composition des fibres. Quand les fibres absorbent de l'eau, elle est d'abord retenue par les parois des cellules elles-mêmes, ce qui provoque une dilatation de la dimension des parois. L'évaporation de cette eau liée provoque le retrait du bois lors du séchage (et l'apparition éventuelle de fissures)<sup>4</sup>. Par exemple, pour une éponge, il s'agit de l'eau qui est contenue dans la matière mais que nous ne pouvons pas essorer ;
- **l'eau libre** : c'est l'eau emprisonnée dans les cavités des cellules. L'eau libre s'évapore facilement sans provoquer de changement de structure.

Le départ de l'eau liée s'accompagne de déformations importantes (jusqu'à 10 % dans la direction tangentielle). Les déformations axiales sont considérées comme négligeables. Le point de saturation d'une fibre représente la limite de déformation du bois.

Ainsi le bois se contracte ou se dilate en fonction des variations de la teneur en humidité ambiante du milieu dans lequel il se trouve.

Ce comportement naturel du bois est à l'origine de certains problèmes rencontrés lors du séchage du bois. Lorsque la pièce de bois sèche, il se produit un gradient d'humidité de part et d'autre de la pièce (sec à l'extérieur, humide à l'intérieur). Les parties externes de la pièce veulent se rétrécir à mesure qu'elles sèchent sous le point de saturation de la fibre, mais le centre, plus humide contraint les parties latérales extérieures. Ce phénomène peut entraîner la formation de craquelures à la surface ou une déformation de la pièce. A ce moment là, les parties latérales prennent leur dimensions finales et ce, même si le coeur continue de sécher et voudra lui aussi rétrécir. Mais les parties latérales, qui ont acquis leur dimension contraignent l'âme, ce qui peut amener la formation de fissures dans le centre de la pièce.

#### 4.4.1.2 Modèle proposé

Nous supposons que, lors de la croissance de l'arbre, le taux d'humidité ambiant reste constant. L'orientation des fibres du bois est donnée par le repère calculé pour l'application de texture. De plus, nous associons à chaque volume un coefficient de saturation en eau ( $\mathbf{H}_{sat}$ ). Ce coefficient permet de détecter, en dessous de quel taux d'humidité le volume se déforme. Ce coefficient est estimé en fonction des paramètres suivants :

- la localisation du volume dans l'arbre lors de sa création (les volumes du tronc et des branches n'ont pas les mêmes indices) ;
- le voisinage du volume (propriétés des volumes voisins, distance au centre, ... ) ;
- la différence entre les dimensions réelles du volume et les dimensions théoriques parfaites (cela nous permet de prendre en compte le bois de compression par exemple) ;
- la saison à laquelle est apparu le volume (le bois d'été est moins spongieux) ;
- un facteur aléatoire qui simule les défauts locaux de structure.

Nous associons également un indice d'humidité à chaque volume, qui représente le degré d'humidité ambiant du volume.

---

<sup>4</sup>L'eau liée correspond à l'eau qui s'évapore lors du séchage, entre 30% et 0% d'humidité.

Ces indices sont ensuite modifiés lorsque les cellules qui constituent le volume meurent (le bois devient du bois dur alors l'indice diminue).

Tous ces paramètres nous permettent de proposer un modèle le plus proche possible de la réalité.

Pour simuler les fractures nous diminuons progressivement le taux d'humidité ambiant. Dans notre modèle, le départ de l'humidité commence à partir des volumes du bord et se propage vers le centre de la pièce de bois.

#### **Etape 0 : initialisation**

Nous commençons par marquer tous les volumes du bord comme volume à traiter. Ce marquage est effectué en affectant le taux d'humidité ambiant à une variable propre au volume ( $H_c$ ).

#### **Etape 1 : calcul du nouveau taux d'humidité**

Pour chaque volume marqué, nous calculons son nouveau taux d'humidité par la formule suivante :

Si  $H_c > H_{sat}$

rien ne se passe (au dessus du seuil de saturation du bois,  
les cellules ne changent pas de taille)

Sinon

$$H_{temp} = \text{abs}(H_{vol} - H_c)$$

Si  $H_{temp} > 0.03$

$$H_{vol} = H_{temp}$$

Sinon

rien ne se passe (la variation du taux d'humidité n'est pas assez importante pour qu'elle ait une influence).

#### **Etape 2 : propagation et éclatement**

Pour chaque Volume marqué ( $V_m$ ), si le taux d'humidité courant a changé, nous recherchons son volume père (dans l'arbre de filiation). Puis nous regardons l'écart entre leur taux d'humidité respectif :

- l'écart est trop grand, les volumes se décollent proportionnellement à cet écart ;
- dans l'autre cas, le volume père est marqué avec pour taux d'humidité courant le taux d'humidité du volume en cours de traitement.

Puis, si de nouveaux volumes ont été marqués, nous réitérons le processus à partir de l'étape 2.

Pour simplifier notre modèle, nous avons ajouté les restrictions suivantes <sup>5</sup> :

- le taux d'humidité ambiant (taux externe) est uniforme ;
- nous supposons, à juste titre, que la propagation ne se fait que dans le sens radial de l'arbre de l'extérieur vers l'intérieur (comme nous l'avons vu précédemment le taux de rétractation axial est presque nul) ;
- nous attendons que l'état d'équilibre soit atteint avant de re-modifier le taux d'humidité ambiant (d'un point de vue technique cela revient à attendre que le modèle ait atteint son point d'équilibre avant de recharger le taux d'humidité ambiant).

---

<sup>5</sup>Toutes ces restrictions peuvent être levées avec une approche plus fine du phénomène.

#### 4.4.1.3 Opération de décollement progressif

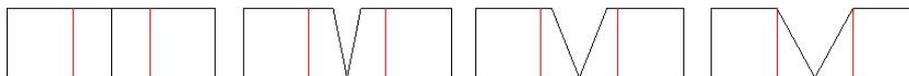


FIG. 4.22 – Séparation progressive des volumes

La première approche pour simuler une fissure entre deux volumes, serait tout simplement de les décoller par leurs faces de contact respectives (liaisons  $\alpha_3$ ). Cependant, cette granularité primaire, s'est vite avérée insuffisante.

Nous rajoutons donc un coefficient de décollement complémentaire. Ce coefficient va de 0 à 1 (0 quand les volumes sont collés, 1 quand les volumes sont complètement décollés (*figure 4.22*)). Cette technique nous permet de discrétiser les étapes intermédiaires pour la simulation des fractures et des fissures.

#### 4.4.1.4 Illustrations

Nous illustrons ce modèle par la (*figure 4.23*). Sur ce modèle des fissures commencent à apparaître, puis se transforment en fractures.

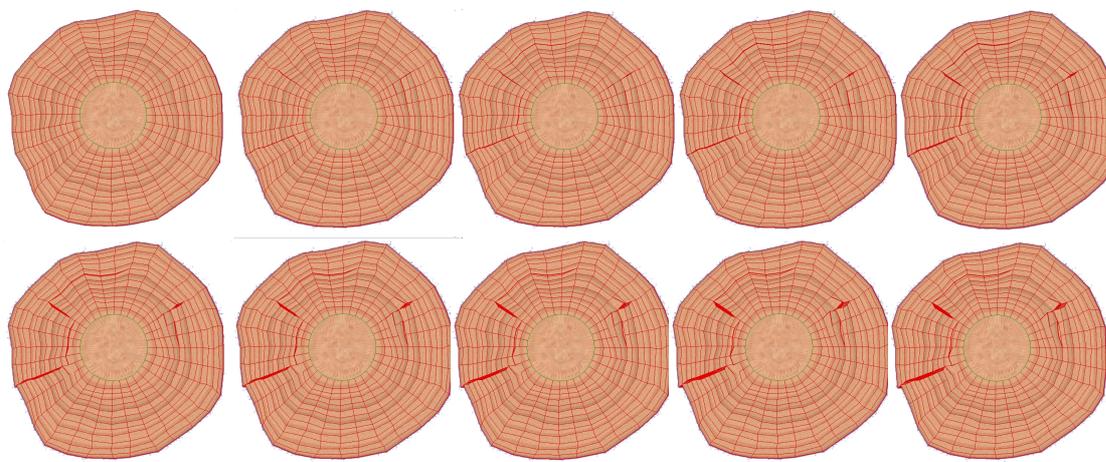


FIG. 4.23 – Exemple d'évolution de fractures

Cependant, avec l'introduction de fissures et de fractures, nous ne pouvons plus utiliser la technique de génération de texture 3D. En effet, cette approche ne permet pas de gérer convenablement l'absence de matière dans certaines zones.

Nous avons développé un outil qui permet d'exporter une plus grande partie du modèle (maillage complet et texture par volume). Pour visualiser les résultats, nous réalisons des intersections entre le maillage et la pièce à texturer.

Nous donnons un exemple primaire sur la (*figure 4.24*). Il s'agit d'une poutre qui commence à se fissurer. Nous pouvons constater que les fissures sont trop rectilignes, et qu'il faudrait un plus grand niveau de détails pour que l'aspect visuel soit plus agréable.



FIG. 4.24 – Exemple d'apparition de fente sur une poutre

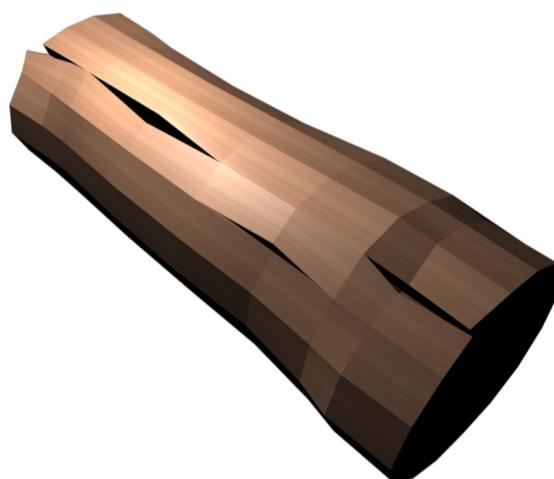


FIG. 4.25 – Apparition de fissures sur un tronc



FIG. 4.26 – Fentes axiale sur un plateau de bois

Cette approche particulière permet de voir les limites du modèle actuel ( *figure 4.25* et *figure 4.26* ). Nous avons effectué quelques travaux en subdivisant les faces pour apporter un plus grand niveau de détails, mais les résultats n'ont pas été concluants. Une des pistes de développement envisageable serait d'associer, par exemple, un système masse-ressort à la structure. Les difficultés se situent surtout au niveau de l'estimation des paramètres pour le système, et de la conservation de l'aspect hétérogène apporté par notre modèle.

#### 4.4.2 Altérations de structure et d'aspect

Nous avons également réalisé un modèle primitif de vieillissement de structure qui agit sur l'aspect visuel de la pièce modélisée (altération des couleurs), ainsi que sur l'usure (simulation de l'érosion). Pour cela, nous nous appuyons, sur le modèle topologique pour connaître les parties tendres du bois (celles qui s'érodent le plus facilement).

Une illustration du ternissement des couleurs est donnée sur la (*figure 4.27*) .



FIG. 4.27 – Une planche dont les couleurs ont terni avec le temps

La (figure 4.28) montre une évolution combinée de l'érosion et du patinage.



FIG. 4.28 – Exemple de vieillissement de planche

## 4.5 Conclusion

Tout au long de ce chapitre, nous venons de voir différents aspects liés à l'exploitation de notre modèle. Nous obtenons des essences de bois variées, et nous sommes capables de simuler certains défauts du bois (bois de coeur, bois de compression,...). Même si nous n'avons pu aborder que partiellement les phénomènes liés au vieillissement (fissure, fracture, érosion), il n'en demeure pas moins que le modèle développé donne des résultats intéressants.



# Conclusion

Dans ce mémoire, nous avons introduit un nouveau modèle de génération de structures internes d'arbres par simulation de croissance. Le contrôle de l'évolution est facilité grâce à la décomposition de la structure en axes principaux et radiaux couplée à des opérations topologiques (volumes se dédoublant dans le voisinage des frontières). Les paramètres de contrôle permettent de représenter une grande variété d'espèces de bois, en utilisant seulement de simples variations telles que des valeurs minimales et maximales de taille de cernes.

Nous avons également présenté une nouvelle catégorie de L-systems, les 3Gmaps L-systems, qui permet de manipuler des subdivisions de volumes. Cette approche permet de contourner les limitations des Cellworks L-systems :

- grâce à l'utilisation d'opérations définies sur les volumes et à une gestion automatique des labels. L'expression des règles de production en est largement facilitée. Cette simplification est faite au détriment d'une certaine généralité (utilisation de prismes orientés), mais elle permet une réutilisation simple et directe des L-systems traditionnels ;
- grâce à une simplification de l'expression des conditions d'adjacences sur les volumes.

Nous avons exploré quelques pistes pour évaluer le potentiel du modèle. Grâce à ce modèle, nous fournissons une bonne cohérence spatiale de l'emplacement des nœuds (cohérence spatiale inhérente au modèle), et une modélisation de quelques autres caractéristiques particulières telles que le bois de réaction, le bois de cœur ou les rayons médullaires. Nous proposons également une première approche de la simulation de phénomènes liés au vieillissement du bois (fissures, altérations des couleurs).

Pour pouvoir aborder de manière plus approfondie les aspects liés au vieillissement, nous pourrions proposer un modèle physique basé sur les éléments finis, et généré à partir de notre structure. Ensuite, il serait plus simple de simuler des fissures plus évoluées, des fractures, et des déformations.

Enfin, notre modèle peut également être employé dans d'autres champs d'applications tels que l'optimisation de découpe pour l'industrie du bois. La localisation des nœuds et la connaissance de la structure interne permettent d'optimiser de manière qualitative et quantitative la coupe pour chaque tronc et de simuler des découpes d'objets dans des pièces de bois, comme le montre la *(figure 4.29)* .



FIG. 4.29 – Théière en bois obtenue à partir de notre modèle

# Annexe : GGToPLSystem

## Modeleur

Dans cette annexe, nous allons nous intéresser plus particulièrement aux solutions techniques retenues pour l'implémentation.

Pour la partie modeleur proprement dite (3-Gcartes), nous avons choisi de nous appuyer partiellement sur ToPLSystem, car celui-ci a été développé et utilisé pour des travaux analogues par l'un de mes co-directeurs de thèse, Mr Olivier Terraz. Il existe bien sûr d'autres modeleurs topologiques, le plus similaire étant à notre connaissance MOKA<sup>6</sup>.

## Historique

ToPLSystem a été originellement écrit en SMALLTALK par Olivier Terraz dans le cadre de ses travaux de recherches au début des années 90. Un nouveau noyau a été écrit en C++ avec une interface sous XForms<sup>7</sup> au tout début des années 2000. Puis, à partir de 2003, lors de mon stage de DEA, j'ai repris ce projet. J'ai commencé par une refonte complète de l'interface en GTKmm (la librairie XForms étant trop restrictive pour nos besoins). Au fil des années cette dernière n'a cessée d'évoluer et de s'étoffer (*figure 4.30*), (*figure 4.31*). Puis, j'ai enrichi considérablement le noyau en y incorporant des nouvelles opérations topologiques (opération de coupe, éclatement de volumes, . . .)

Dans sa première version, ce modeleur a été développé uniquement pour générer des arbres. Cette structure monolithique a rapidement montré ses limites pour la modélisation d'autres objets. J'ai donc dû repenser l'ensemble sous une forme plus modulaire (noyau distinct de l'interface et des outils spécifiques à la croissance). Cette nouvelle structuration modulaire présente plusieurs avantages :

- une maintenance plus simple (les différents modules unitaires sont séparés) ;
- la possibilité d'avoir plusieurs développements parallèles (modules sur les arbres, L-systems, poissons, feuilles, . . .).

Il existe trois modules principaux, ou modules de base :

- un noyau qui contient l'implémentation des cartes et des principaux plongements ;
- un module d'affichage des cartes écrit en OpenGL ;
- une interface de base écrite en GTKmm.

---

<sup>6</sup>Ce logiciel peut être trouvé à l'adresse suivante : <http://www.sic.sp2mi.univ-poitiers.fr/moka/index.html>

<sup>7</sup><http://savannah.nongnu.org/projects/xforms>



plus grande souplesse dans la déclaration de nouvelles fonctionnalités avec des règles de réécritures qui peuvent être définies à la volée ;

- un module sur la génération de poissons : pour ce dernier, j’ai adapté les travaux d’un projet d’étudiants en maîtrise sur la modélisation déclarative de poissons. Ce module a ensuite été complété par un stage de maîtrise sur des techniques d’application d’images 2D de poissons sur des structures des poissons en 3D. Cet ensemble a servi dans le cadre d’un stage de DEA pour l’animation de poissons en fonds marins ;
- un module sur la reconstruction 3D de grume de bois (non achevé) : à partir de photos d’une grume de bois prises sous divers angles, et à l’aide de marqueurs, des données caractéristiques de la grume sont extraites (nombre de cernes, position des branches, . . .). A partir de ces dernières et en fonction de l’espèce, une grume de bois virtuelle devait être reconstruite afin de simuler des découpes qualitatives de bois (le modèle permet de positionner les noeuds). Ce travail qui s’appuie sur notre modéleur a été réalisé par deux étudiants en école d’ingénieur. Cependant des difficultés ont été rencontrées pour détecter convenablement la position des cernes et les angles des branches sur le pourtour de la grume.

Nous venons d’avoir un bref aperçu des principaux travaux liés directement avec notre modéleur, auxquels j’ai participé. Une partie de ces travaux est illustrée dans les (*figure 4.32*) et (*figure 4.33*) .

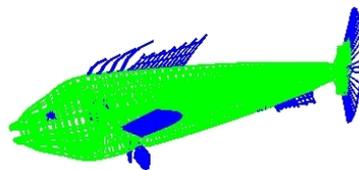


FIG. 4.32 – La première structure de poisson obtenue

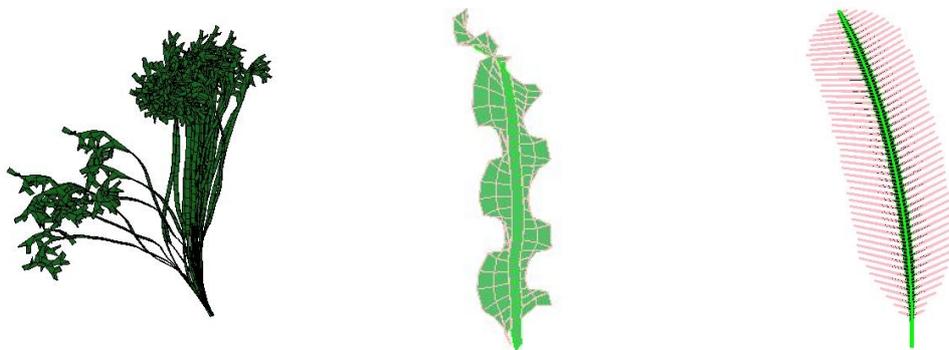


FIG. 4.33 – Différentes modélisations d’algues et de plantes



# Bibliographie

- Abelson H., diSessa A. A., *Turtle Geometry : The Computer as a Medium for Exploring Mathematics*, MIT Press, Cambridge, MA, 1981.
- Agin G. J., Binford T. O., « Computer Description of Curved Objects », *IEEE Trans. Computers*, vol. 25, n° 4, 1976, p. 439-449.
- Aono M. K. L., « Botanical Image Generation », *IEEE Computer Graphics and Applications*, vol. 4, IEEE Computer Society, 1984, p. 10-34.
- Barry-Lenger A., Pierson J., Poncelet J., *Transformation, utilisation, et industrie du bois en Europe*, Editons du Perron, 1999.
- Bertrand Y., Dufourd J.-F., Françon J., Lienhardt P., « 3-Dimensional Manifold Modeling using 3-Dimensional Generalized Maps », Technical report, Université Louis Pasteur, Strasbourg, 1993a.
- Bertrand Y., Dufourd J.-F., Françon J., Lienhardt P., « Algebraic Specification and Development in Geometric Modeling », *TAPSOFT*, 1993b, p. 75-89.
- Blinn J. F., Newell M. E., « Texture and reflection in computer generated images », *Commun. ACM*, vol. 19, n° 10, 1976, p. 542-547.
- Bloomenthal J., « Modeling the mighty maple », in P. Cole, R. Heilman, B. A. Barsky (eds), *SIGGRAPH*, ACM, 1985, p. 305-311.
- Bloomenthal J., « Bulge Elimination in Convolution Surfaces », *Comput. Graph. Forum*, vol. 16, n° 1, 1997, p. 31-41.
- Boudon F., Prusinkiewicz P., Federl P., Godin C., Karwowski R., « Interactive design of bonsai tree models », *Computer Graphics Forum. Proceedings of Eurographics*, vol. 22, n° 3, 2003, p. 591-599.
- Buchanan J. W., « Simulating Wood Using a Voxel Approach », *Comput. Graph. Forum*, vol. 17, n° 3, 1998, p. 105-112.
- Catmull E. E., « A subdivision algorithm for computer display of curved surfaces », PhD thesis, 1974.
- Chen X., Lienhardt P., « Modelling and Programming Evolutions of Surfaces », *Comput. Graph. Forum*, vol. 11, n° 5, 1992, p. 323-341.
- Chomsky N., « Three Models for the Description of Language », *IRA Transactions on Information Theory*, vol. 2, n° 3, 1956, p. 113-124.

- Chomsky N., Miller G. A., « Finite State Languages », *Information and Control*, vol. 1, n° 2, 1958, p. 91-112.
- Deussen O., Hanrahan P., Lintermann B., Mech R., Pharr M., Prusinkiewicz P., « Realistic Modeling and Rendering of Plant Ecosystems », *SIGGRAPH*, 1998, p. 275-286.
- Deussen O., Lintermann B., *Digital Design of Nature : Computer Generated Plants and Organics*, X.media.publishing, Springer-Verlag, 2005.
- Dischler J.-M., Ghazanfarpour D., « A Geometrical Based Method for Highly Complex Structured Textures Generation », *Comput. Graph. Forum*, vol. 14, n° 4, 1995, p. 203-216.
- Dischler J.-M., Ghazanfarpour D., « Interactive Image-based Modeling of Macrostructured Textures », *IEEE Comput. Graph. Appl.*, vol. 19, n° 1, 1999, p. 66-74.
- Dischler J.-M., Ghazanfarpour D., « A survey of 3D texturing », *Computers & Graphics*, vol. 25, n° 1, 2001, p. 135-151.
- Dufourd J.-F., « An OBJ3 functional specification for boundary representation », *Symposium on Solid Modeling and Applications*, 1991, p. 61-72.
- Fleischer K. W., Laidlaw D. H., Currin B. L., Barr A. H., « Cellular texture generation », *SIGGRAPH '95 : Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 1995, p. 239-248.
- Fournier A., Fussell D., Carpenter L., « Computer rendering of stochastic models », *Commun. ACM*, vol. 25, n° 6, 1982, p. 371-384.
- Françon J., Lienhardt P., « Basic principles of topology-based methods for simulating metamorphoses of natural objects », in N. Thalmann, D. Thalmann (eds), *Artificial Life and Virtual Reality*, John Wiley & Sons Ltd., 1994, p. 23-44.
- Galbraith C., MacMurphy P., Wyvill B., « BlobTree Trees », *CGI '04 : Proceedings of the Computer Graphics International*, IEEE Computer Society, Washington, DC, USA, 2004, p. 78-85.
- Gardner G. Y., « Simulation of natural scenes using textured quadric surfaces », *SIGGRAPH '84 : Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 1984, p. 11-20.
- Gardner G. Y., « Visual simulation of clouds », *SIGGRAPH '85 : Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 1985, p. 297-304.
- Ghazanfarpour D., Dischler J.-M., « Generation of 3D Texture Using Multiple 2D Models Analysis », *Comput. Graph. Forum*, vol. 15, n° 3, 1996, p. 311-323.
- Gobron S., Chiba N., « Crack Pattern Simulation Based on 3D Surface Cellular Automaton », *CGI '00 : Proceedings of the International Conference on Computer Graphics*, IEEE Computer Society, Washington, DC, USA, 2000, p. 153.
- Golden J., Yoon H., « Heterocyst formation in *Anabaena* », , vol. 1, n° 6, 1998, p. 623-629.

- Grimm C., « Implicit generalized cylinders using profile curves », *Proc. of Implicit Surfaces*, 1999, p. 33-41.
- Guiard N., « Construction de modèles géologiques 3D par co-raffinement de surfaces », PhD thesis, Ecole des Mines de Paris, Paris, Mai, 2006.
- Guimberteau G., Terraz O., Mérillou S., « Simulation de la croissance interne de bois basée sur des subdivisions de volumes », *Journées AFIG*, 2004. Conférence nationale.
- Guimberteau G., Terraz O., Mérillou S., « Modélisation volumique de bois par simulation de croissance », *Technique et Science Informatiques (TSI)*, vol. 25, n° 6, 2006, p. 735-757.
- Hanan J., « Parametric L-systems and their application to the modelling and visualization of plants », PhD thesis, University of Regina, June, 1992.
- Hart J. C., Baker B., Michaelraj J., « Structural simulation of tree growth and response », *The Visual Computer*, vol. 19, n° 2-3, 2003, p. 151-163.
- Heeger D. J., Bergen J. R., « Pyramid-based texture analysis/synthesis », *SIGGRAPH '95 : Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 1995, p. 229-238.
- Iben H. N., O'Brien J. F., « Generating surface crack patterns », *SCA '06 : Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2006, p. 177-185.
- Jones H., Kaandorp J., « Modelling of growing natural forms », *Tutorial T2, Eurographics 2000*, vol. T2, 2000, p. 1-32.
- Kajiya J. T., Kay T. L., « Rendering fur with three dimensional textures », *SIGGRAPH Comput. Graph.*, vol. 23, n° 3, 1989, p. 271-280.
- Kang M., Yan H., Reffye P., Jaeger M., Hu B., Houllier F., « A fast algorithm for calculating stem and branch radial growth of a tree with substructure approach », *Fourth Workshop on : Connexion between Forest ressources and wood quality : Modelling approaches and simulation software*, September, 2002.
- Kurth W., « Morphological models of plant growth : Possibilities and ecological relevance », *Ecological Modelling*, vol. 75/76, Elsevier Science B.V., 1994, p. 299-308.
- Lefebvre S., Neyret F., « Synthesizing bark », *EGRW '02 : Proceedings of the 13th Eurographics workshop on Rendering*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2002, p. 105-116.
- Lienhardt P., « Free-form surfaces modeling by evolution simulation », *Eurographics Conference Proceedings*, 1988, p. 327-341.
- Lienhardt P., « N-Dimensional Generalized Topological Maps », Technical report, Mars, 1989.
- Lienhardt P., « N-dimensional generalized combinatorial maps and cellular quasi-manifolds », *International Journal on Computational Geometry and Applications*, vol. 4, n° 3, 1994, p. 275-324.

- Lienhardt P., « A study of basic tools for simulating metamorphoses of subdivided three-dimensional objects », Technical report, May, 1995.
- Lienhardt P., Françon J., « Synthèse d'images de feuilles végétales », *Actes du 3ème colloque Image*, Paris, France, 1987.
- Lindenmayer A., « Mathematical Models for Cellular Interactions in Development, I & II », *Journal of Theoretical Biology*, vol. 18, 1968, p. 280-315.  
*Lindenmayer's original articles on L-Systems.*
- Lindenmayer A., Rozenberg G., « Parallel Generation of Maps : Developmental Systems for Cell Layers », *Proceedings of the International Workshop on Graph-Grammars and Their Application to Computer Science and Biology*, Springer-Verlag, London, UK, 1979, p. 301-316.
- Lintermann B., Deussen O., « Interactive Modeling of Plants », *IEEE Comput. Graph. Appl.*, vol. 19, n° 1, 1999, p. 56-65.
- Longuetaud F., « Détection et analyse non destructive de caractéristiques internes de billons d'Epicéa commun ((L.) Karst) par tomographie à rayons X », PhD thesis, Ecole Nationale du Génie Rural des Eaux et Forêts, Unité associée INRA/ENGREF en Sciences Forestières, Octobre, 2005.
- Mandelbrot B. B., *The Fractal Geometry of Nature*, W. H. Freedman and Co., New York, 1983.
- Maritaud K., « Rendu réaliste d'arbres vus de près en images de synthèses », PhD thesis, Laboratoire MSI, Faculté des Sciences et Techniques, Université de Limoges, decembre, 2003.
- Marschner S. R., Westin S. H., Arbree A., Moon J. T., « Measuring and modeling the appearance of finished wood », *ACM Trans. Graph.*, vol. 24, n° 3, 2005, p. 727-734.
- Max N., « Cone-spheres », *SIGGRAPH '90 : Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 1990, p. 59-62.
- Mech R., Prusinkiewicz P., « Visual Models of Plants Interacting with Their Environment », *SIGGRAPH*, 1996, p. 397-410.
- Mérillou S., Dischler J.-M., Ghazanfarpour D., « A BRDF Postprocess to Integrate Porosity on Rendered Surfaces », *IEEE Trans. Vis. Comput. Graph.*, vol. 6, n° 4, 2000, p. 306-318.
- Peachey D. R., « Solid texturing of complex surfaces », *SIGGRAPH '85 : Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, ACM Press, 1985, p. 279-286.
- Perlin K., « An image synthesizer », *SIGGRAPH '85 : Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, ACM Press, 1985, p. 287-296.
- Perlin K., Hoffert E. M., « Hypertexture », *SIGGRAPH '89 : Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, ACM Press, 1989, p. 253-262.

- Perttunen J., Sievänen R., Nikinmaa E., Salminen H., Saarenmaa H., Väkevä J., « LI-GNUM : a tree model based on simple structural units », *Annals of Botany*, vol. 77, n° 1, 1996, p. 87-98.
- Peyrat A., Terraz O., Merillou S., Galin E., « Generating vast varieties of realistic leaves with parametric 2Gmap L-systems », *Vis. Comput.*, vol. 24, n° 7, 2008, p. 807-816.
- Prusinkiewicz P., Hanan J., « Visualization of botanical structures and processes using parametric L-systems », *Scientific visualization and graphics simulation*, vol. 1, 1990a, p. 183-201.
- Prusinkiewicz P., Hanan J., Mech R., « An L-system-based plant modeling language », *CInternational workshop AGTIVE'99*, 1999, p. 395-410.
- Prusinkiewicz P., Lindenmayer A., *The algorithmic beauty of plants*, Springer-Verlag, New York, 1990b.
- Prusinkiewicz P., Mündermann L., Karwowski R., Lane B., « The use of positional information in the modeling of plants », *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM Press, 2001, p. 289-300.
- Reeves W. T., « Particle systems A technique for modeling a class of fuzzy objects », *SIGGRAPH '83 : Proceedings of the 10th annual conference on Computer graphics and interactive techniques*, ACM Press, 1983, p. 359-375.
- Reeves W. T., Blau R., « Approximate and probabilistic algorithms for shading and rendering structured particle systems », *SIGGRAPH '85 : Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, ACM Press, 1985, p. 313-322.
- Reffye P., Edelin C., Françon J., Jaeger M., Puech C., « Plant models faithful to botanical structure and development », *SIGGRAPH '88 : Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, ACM Press, 1988, p. 151-158.
- Reffye P., Fourcaud T., Blaise F., Barthélémy D., Houllier F., « A functional model of tree growth and tree architecture », *Silva Fennica*, vol. 31, n° 3, 1997a, p. 297-311.
- Reffye P., Houllier F., Blaise F., Fourcaud T., « Essai sur les relations entre l'architecture d'un arbre et la grosseur de ses axes végétatifs », in J. Bouchon, P. Reffye, D. Barthélémy (eds), *Modélisation et simulation de l'architecture des végétaux*, INRA, Paris-France, 1997b, p. 255-423. Collection : Science Update.
- Reynolds C. W., « Flocks, herds and schools : A distributed behavioral model », *SIGGRAPH '87 : Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 1987, p. 25-34.
- Sims K., « Artificial evolution for computer graphics », *SIGGRAPH '91 : Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 1991, p. 319-328.
- Szafran N., Despreaux S., Biard L., Blaise F., « Sawing of Logs in Virtual Trees using 3D-Intersection Algorithms », *PMA'03 : International Symposium on plant growth Modeling, simulation, visualisation and their application*, Octobre 2003, p. 372-383.

- Szilard A., Quinton R., « An interpretation for DOL systems by computer graphics », , vol. 4, 1979, p. 8-13.
- Terraz O., « Programmation de métamorphoses d'objets surfaciques et volumiques », PhD thesis, Université Louis Pasteur, Strasbourg, Septembre, 1994. ordre 1874.
- Terraz O., Guimberteau G., Mérillou S., Plemenos D., Ghazanfarpour D., « 3Gmap L-systems : an application to the modelling of wood », *The Visual Computer*, vol. ?, 2008, p. ?- ?
- Terraz O., Lienhardt P., « A study of basic tools for simulating metamorphoses of subdivided 2D and 3D objects. Applications to the internal growing of wood and to the simulation of growing fishes », *Act of Sixth Eurographics Workshop on Animation and Simulation*, 1995, p. 104-129.
- Tobler R. F., Maierhofer S., Wilkie A., « Mesh-Based Parametrized L-Systems and Generalized Subdivision for Generating Complex Geometry », *International Journal of Shape Modeling*, vol. 8, n<sup>o</sup> 2, 2002, p. 173-191.
- Vigue J., *Le grand livre du Bois*, Editions Place des Victoires, 2002.
- X.Yin T., N.Chiba, « CG Representation of Wood Aging with Distortion, Cracking and Erosion », *The Journal of the Society for Art and Science*, vol. 3, n<sup>o</sup> 4, 2004, p. 216-223.
- Y. R., P. C., S. S., C. L., « Particle Systems for Plant Modeling », *Int. Symposium on Plant growth Modeling, simulation, visualization and their Applications*, 2003.

# Table des figures

1	Etude de la structure interne des gourmands de chêne : image issue du tomographe à rayon X (à gauche), et reconstruction 3D à droite. . . . .	9
1.1	De gauche à droite : arbre (chêne), arbuste (néflier), arbrisseau (laurier)	14
1.2	Coupe macroscopique du chêne (de gauche à droite) : chêne sec, chêne mouillé, et coupe zoomée . . . . .	17
1.3	Divers essences de bois : noyer, merisier, chêne, pin . . . . .	17
1.4	Les différentes parties d'un arbre . . . . .	18
1.5	Noeud, internœud et bourgeons . . . . .	19
1.6	Les différents aspects du tronc pour des arbres résineux, feuillus et monocotylédones résineux . . . . .	20
1.7	Pin parasol déformé par le vent . . . . .	20
1.8	Coupe transversale . . . . .	21
1.9	Croissance de la tige d'un arbre . . . . .	22
1.10	Coupe radiale et transversale d'un arbre . . . . .	24
1.11	Structuration principale . . . . .	24
2.1	Image issue de l'article de Reev85 . . . . .	29
2.2	Diagramme du comportement des particules (à gauche) et modélisation de l'arbre (à droite) . . . . .	29
2.3	Développement de l'arbre . . . . .	30
2.4	Structure d'un arbre naturel . . . . .	31
2.5	Exemple d'arbre obtenu à partir du logiciel NatFX . . . . .	31
2.6	Structure d'une feuille . . . . .	32
2.7	Dégradation des feuilles d'une branche de rosier (de la gauche vers la droite, les feuilles sont affectées par une altération des couleurs, des attaques d'insectes, et des flétrissements) . . . . .	33
2.8	Détermination des voxels voisins dans le plan radial . . . . .	34
2.9	Vue 2D du choix de sélection du voxel (A, B, C sont trois voxels cibles pour le voxel pointé par la flèche) . . . . .	34
2.10	Planche de bois coupé dans le sens axial . . . . .	35
2.11	Application de la texture sur des objets . . . . .	35
2.12	Modèle de croissance . . . . .	36
2.13	Gestion de la branche . . . . .	36

2.14	Calcul de l'intersection . . . . .	37
2.15	Planches obtenues à partir du premier modèle . . . . .	37
2.16	Paramètres des troncs de cônes . . . . .	39
2.17	Planches obtenues à partir du second modèle . . . . .	39
2.18	Images de la méthode de Ter94 . . . . .	40
2.19	Exemple de plantes modélisées avec OnyxGarden 6 . . . . .	41
2.20	Arbre produit avec le logiciel XFROG de la société Greenworks . . . . .	42
2.21	Interface d'XFROG : à gauche hiérarchie de l'arbre et à droite interprétation visuelle du graphe . . . . .	43
2.22	Exemple d'arbo produit avec le logiciel natFx de la société bionatics . . . . .	44
2.23	Détails des bijections $\alpha_0, \alpha_1, \alpha_2$ de deux faces recollées le long d'une arête . . . . .	46
2.24	Détails des bijections $\alpha_3$ permettant de recoller deux volumes le long d'une face . . . . .	46
2.25	Exemple de dépendance : la classe volume . . . . .	50
2.26	Création d'un volume cylindrique de degré 6 . . . . .	53
2.27	Identification de deux faces non isomorphes . . . . .	54
2.28	Éclatement d'une face . . . . .	55
2.29	Éclatement d'un volume . . . . .	55
2.30	Les principaux éléments du plongement : repère du volume, point représentatif des faces (marron) et sommets (verts) . . . . .	57
2.31	Calcul final des coordonnées des sommets en tenant compte des relations de voisinage . . . . .	57
2.32	Une UCL avec l'emplacement pour une branche . . . . .	62
2.33	Une UB de deux ans . . . . .	64
2.34	Labellisation des volumes cylindriques d'ordre 3 : modèle d'origine (à gauche), modèle utilisé (à droite) . . . . .	65
2.35	Demi-unité de jointure sur son support (de la gauche vers la droite) : vue de côté, de dessous, de dessus et coupe de trois-quarts . . . . .	66
2.36	Différent points de vue d'une coupe transversale d'une unité de jointure sur son support . . . . .	66
2.37	Croissance en épaisseur par propagation de l'éclatement des volumes des couches externes . . . . .	67
2.38	Co-raffinement de deux maillages 2D . . . . .	69
2.39	Marquage des objets pour la découpe (de gauche à droite) : marquage des sommets, éclatement et marquage des arêtes, éclatement et marquage des faces . . . . .	70
2.40	Quelques coupes transversales issues du premier modèle . . . . .	72
3.1	Bactéries <i>Anabaena catenula</i> . . . . .	74
3.2	Les 4 types de bactérie . . . . .	75
3.3	Processus de croissance . . . . .	75
3.4	interprétation du mot F-F+F+f-FF . . . . .	78
3.5	Les notions de lacet, roulis, tangage dans un avion . . . . .	79
3.6	Ramifications d'un arbre . . . . .	80

3.7	Un L-system relativement simple : ses deux règles de réécriture, l'axiome et l'application successive des règles à partir de l'axiome. . . . .	81
3.8	Schéma d'un arbre représenté avec des cylindres . . . . .	82
3.9	A gauche : un cône-sphère. A droite : branche formée de cônes-sphères . . . . .	82
3.10	Exemple de Map L-system : dans le premier step, une distinction est effectuée entre la phase de réécriture des arêtes et la connexion des marqueurs concordants . . . . .	84
3.11	Séquence de développement de cellules de l'épiderme : (a) l'ensemble cellulaire de départ ; (b), (d) et (f) cellworks immédiatement après la division des cellules ; (c), (e) et (g) les cellules correspondantes une fois l'état d'équilibre atteint . . . . .	85
3.12	Labels associés aux faces d'un prisme d'ordre 4 . . . . .	86
3.13	Faces et volumes adjacents : C est adjacent à B par les faces C2 et E (la condition $C_{C2} > B_E$ est donc vraie) . . . . .	88
3.14	L'opération d'éclatement de volume : (HAUT) U.1 Volume initial (face masquée : O), U.2 Résultat de la règle $A \rightarrow^O BC$ , (BAS) D.1 Volume initial (face masquée : C1), D.2 Résultat de la règle $A \rightarrow^{C1} BC$ . . . . .	89
3.15	Propagation de l'éclatement de volumes : a. Volumes initiaux. b. Résultat d'un éclatement en série. c. Résultat d'un éclatement en parallèle . . . . .	90
3.16	Réorganisation des labels : en haut, les volumes avant l'éclatement et en bas, le résultat après l'éclatement ( $A \rightarrow^O AC$ ), et la réorganisation des labels . . . . .	90
3.17	Différence entre les opérations d'ajout et d'éclatement . . . . .	91
3.18	Spirale générée avec le 3Gmap L-system 1 . . . . .	94
3.19	Tronc sans branche, après les étapes 1,2 et 3 . . . . .	96
3.20	Tronc sans branche : étape 1 zoom et résultat après les étapes 1,2,3,4. volume de label A : vert, F : rose, R : rouge, S bleu . . . . .	97
3.21	Tronc avec branche, après les étapes 1,2 et 3 . . . . .	98
3.22	Tronc avec branches : à gauche le bois d'été en noir et le bois de printemps en blanc ; à droite le bois avec des textures . . . . .	98
3.23	Détail de la figure 3.22 . . . . .	99
4.1	Extraction et extrusion de deux modèles 2D naturels . . . . .	104
4.2	Image de bois synthétisée montrant les déformations des volumes . . . . .	105
4.3	Comparaison entre une photo réelle (gauche) et une image synthétisée (droite) . . . . .	106
4.4	Morceau de bois rendu en temps réel . . . . .	106
4.5	Pin. a : plan de coupe dans le tronc ; b : photographie réelle de pin ; c : résultat de la coupe . . . . .	107
4.6	Planche de pin avec 3 nœuds . . . . .	107
4.7	Mélèze. a : plan de coupe dans le tronc ; b : photographie réelle de mélèze ; c : résultat de la coupe . . . . .	107
4.8	Orme. a : photographie réelle ; b : résultat de la coupe . . . . .	107

4.9	Problèmes de discontinuité et de déformations liées au placage de textures planes . . . . .	109
4.10	Exemples de textures naturelles : marbre et bois . . . . .	111
4.11	Exemple de génération de textures avec différentes perturbations et bruits	112
4.12	Insertion de macrostructure, ici un nœud . . . . .	113
4.13	Planche de bois rendue avec le logiciel Blender . . . . .	115
4.14	Horloge générée avec une texture de bois sans nœud . . . . .	115
4.15	Zoom sur un détail de l'horloge . . . . .	115
4.16	Tore en bois . . . . .	116
4.17	Duramen ou bois de coeur . . . . .	116
4.18	Bois de réaction . . . . .	116
4.19	Chêne, rayons médullaires . . . . .	117
4.20	Détail de la courbure d'une branche . . . . .	117
4.21	Vue macroscopique (à gauche) et microscopique (à droite) d'une coupe transversale d'un morceau de bois . . . . .	119
4.22	Séparation progressive des volumes . . . . .	122
4.23	Exemple d'évolution de fractures . . . . .	122
4.24	Exemple d'apparition de fente sur une poutre . . . . .	123
4.25	Apparition de fissures sur un tronc . . . . .	123
4.26	Fentes axiale sur un plateau de bois . . . . .	124
4.27	Une planche dont les couleurs ont terni avec le temps . . . . .	124
4.28	Exemple de vieillissement de planche . . . . .	125
4.29	Théière en bois obtenue à partir de notre modèle . . . . .	128
4.30	L'interface historique . . . . .	130
4.31	Dernière interface de développement . . . . .	130
4.32	La première structure de poisson obtenue . . . . .	131
4.33	Différentes modélisations d'algues et de plantes . . . . .	131



# Modélisation volumique de bois

## Résumé

Dans ce mémoire, nous présentons les résultats de nos travaux sur la génération de structures internes d'arbres basée sur la simulation de la croissance de l'arbre. Pour la modélisation de ces structures, nous utilisons un modèle permettant la représentation de subdivisions d'objets volumiques : les 3Gcartes. Notre approche permet, grâce à une structuration d'objets à différents niveaux (cerne, volume), de simplifier le contrôle de l'évolution. De plus, nous proposons une formalisation du contrôle de la croissance à l'aide d'une extension des L-systems en dimension 3 : les 3Gmap L-systems.

Munis de cette modélisation et de ce formalisme, nous sommes capables de générer des structures internes d'arbres plausibles d'un point de vue botanique. Pour valider notre modèle, nous proposons quelques pistes d'exploitations : génération de textures de bois avec gestion des nœuds, simulation de défauts du bois, du vieillissement, de fissures, . . .

## Mots clés

Simulation de la croissance de bois, Phénomène naturel, Subdivision géométrique, Modélisation géométrique, L-systems, Rendu de bois

# Volumic modelling of wood

## Abstract

In this thesis, we present the results of our research about the generation of internal structure of trees based on the simulation of the tree growth. For the modelling of these structures, we use a model which allows the presentation of subdivisions of 3D objects : the 3Gmaps. Thanks to a structuring of objects at different levels (ring, volume), our approach allow simplifying the control of the evolution. Moreover, we add a formalization of the growth management with an extension of L-systems in dimension 3 : 3Gmap L-systems.

Provided with this modelling and with this formalism, we are able of generating internal wood structures and credible from a botanical point of view. To validate our model, we offer some lanes of farms : generation of wooden textures with management of knots, wood defects, weathering, . . .

## Keywords

Wood growth simulation, Natural phenomena, Geometric subdivision, Geometric modelling, L-systems, Natural phenomena, Wood rendering