

# Représentation des NURBS par Systèmes Itérés de Fonctions

Lucas Morlet, Marc Neveu, Sandrine Lanquetin, et Christian Gentil

Laboratoire Electronique, Informatique et Image  
UFR Sciences et Techniques, allée Alain Savary, 21000 Dijon, France

---

## Résumé

*Les Systèmes Itérés de Fonctions (IFS) sont un outil standard pour la génération de formes fractales. Les IFS contrôlés (CIFS) en sont une extension pour la création de formes fractales à dessein industriel. Un des avantages de cette approche est la possibilité de représenter des surfaces standards comme les surfaces de Bézier, Splines, et de subdivision. La représentation des surfaces par un unique formalisme facilite leur manipulation et la gestion des interactions comme par exemple la construction de raccords entre deux surfaces de natures différentes.*

*Dans cet article, la formulation des B-Splines Rationnelles Non-Uniformes (NURBS) dans le formalisme des CIFS est présentée. Les NURBS étant l'outil central de la plupart des systèmes CAO, leur ajout à la liste des "objets IFS" permet d'augmenter l'intérêt de ce formalisme dans le cadre d'applications industrielles.*

*En analysant la récursivité du processus de génération des fonctions de base, nous montrons que celui-ci est stationnaire. Le calcul peut alors être représenté par un automate CIFS dont les matrices de subdivision associées sont facilement déduites de la forme blossoming et exprimées en fonction du vecteur nodal initial.*

---

**Mots-clés :** NURBS, IFS, CIFS, CAO

## 1. Introduction

Créés par Hutchinson [Hut81] puis démocratisés par Barnsley [Bar14], les Systèmes Itérés de Fonctions (IFS) sont, à la base, une formulation nouvelle des fractales. Ils sont constitués d'un ensemble de transformations qui vont être appliquées récursivement sur un objet. L'objet obtenu après une infinité d'applications de l'IFS est appelé attracteur et peut être étudié grâce aux points fixes des transformations.

Bien que conçus à la base pour les fractales, elles ne sont pas les seuls objets générables par IFS : les surfaces de Bézier, B-Splines, et de subdivision [SLG05] [WW01], par exemple, peuvent être écrites sous cette forme. L'idée principale étant que si un objet peut être généré de façon itérative, il peut être décrit par un IFS et, dans le cas qui nous intéresse, par un automate CIFS.

Comme l'a montré Podkorytov [PGSL14], formaliser des objets de natures différentes sous forme d'automates CIFS permet un certain nombre d'interactions entre-eux. Par exemple, formuler le système itératif de génération de surfaces B-Splines uniformes quadratique et cubique sous la forme de deux automates CIFS permet de trouver un automate de raccord entre les deux surfaces.

Grâce à leur précision et la possibilité de les contrôler totalement, les B-Splines Rationnelles Non-Uniforme (NURBS) sont un outil incontournable des systèmes CAO. L'influence des points du polygone de contrôle sur la courbe est généralement calculée de manière récursive mais elle peut également l'être par *blossoming* [Ram87]. Les NURBS sont également générables par itération avec les algorithmes de Boehm ou d'Oslo. Elles sont donc formalisables sous forme d'IFS et d'automates CIFS et peuvent alors interagir avec les autres "objets IFS". Cet article décrit comment construire les automates CIFS associés à la construction de NURBS de degré quelconque.

Dans la [Section 2](#), le formalisme des IFS et des CIFS est présenté en utilisant comme exemple l'automate CIFS associé à l'algorithme de Chaikin [Cha74]. La formulation *blossoming* des NURBS est ensuite décrite dans la [Section 3](#). De cette formulation sont déduites les matrices de transitions des automates CIFS associés aux NURBS qui sont donnés dans la [Section 4](#). La création des automates de surface NURBS par "produit tensoriel" de deux automates de courbe NURBS est décrit en [Section 5](#). Enfin, différentes applications de cette nouvelle formulation des NURBS sont présentées en [Section 6](#).

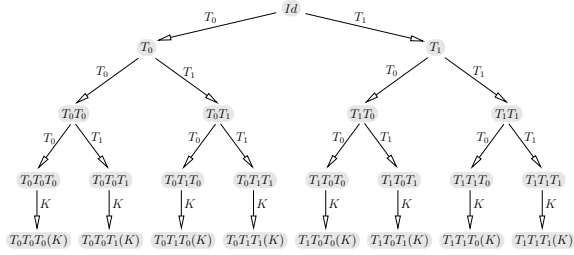


Figure 1: L'arborescence des adresses pour un IFS à deux transformations  $T_0$  et  $T_1$ .

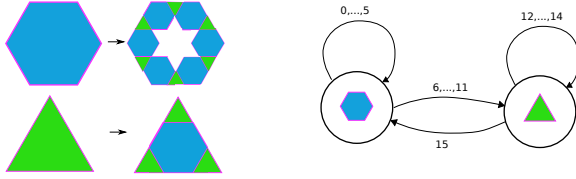


Figure 2: Exemple d'automate CIFS d'une transformation à deux états : un "état hexagone" et un "état triangle". Les transformations 0 à 5 bouclent sur l'hexagone, les transformations 12, 13 et 14 bouclent sur le triangle et les transformations 6 à 11 et 15 change l'état courant

## 2. Systèmes Itérés de Fonctions et IFS contrôlés

Les IFS permettent de formaliser la notion d'autosimilarité d'une figure géométrique  $F$  en décrivant comment elle peut être composée de copies plus petites d'elle-même :  $F = \bigcup_{i=0}^{N-1} T_i(F)$ . Un IFS est un ensemble de transformations, chaque transformation appliquée à la figure autosimilaire lui fait correspondre l'une de ses parties. D'un autre côté, si l'on donne  $N$  transformations contractantes  $T_i, i = \{0, \dots, N-1\}$  définies dans un espace métrique complet  $(E, d)$ , il existe un unique compact  $F$  de  $(E, d)$ , i.e. une figure, vérifiant  $F = \bigcup_{i=0}^{N-1} T_i(F)$  (théorème du point fixe). Le calcul de la figure  $F$  se fait alors par itération de l'opérateur de Hutchinson,  $T(K) = \bigcup_{i=0}^{N-1} T_i(K)$ , à partir d'un compact initial  $K$  quelconque.  $F$  est le point fixe de  $T$  et est appelé attracteur de l'IFS et l'on a  $F = \lim_{n \rightarrow \infty} T^n(K)$ . Pour l'évaluation d'une figure  $F$  décrite par un IFS, on se fixe un niveau d'itération  $n$ , et l'on calcule une approximation de l'attracteur  $T^n(K)$ . Cette évaluation peut être représentée par un arbre d'évaluation décrivant les combinaisons des transformations à appliquer à  $K$ . Un exemple d'arbre est donné à la Figure 1.

L'arbre d'évaluation montre clairement qu'à chaque niveau d'itération toutes les transformations sont appliquées pour calculer le niveau suivant. Ainsi, chaque partie de la figure se décompose toujours de la même façon en sous parties. En introduisant un automate, il est possible de "piloter" cette décomposition et de définir des règles de subdivision différentes pour chaque partie comme dans la Figure 2. A chaque transition de l'automate est associée une transformation. L'automate permet de savoir, à partir d'un état, quelles

transformations il faut appliquer pour opérer le processus de subdivision. Les IFS enrichis d'un automate sont appelés CIFS (Controlled IFS [SGGM15]). Pour un IFS standard, l'automate est implicite et est constitué d'un seul état avec une transition par transformation bouclant sur ce même état.

La paramétrisation des attracteurs se fait à l'aide de la notion d'adresse. Un noeud de l'arbre d'évaluation représente une partie de l'attracteur. La branche d'un noeud représente la transformation à utiliser pour obtenir une sous-partie. Ainsi, un chemin dans l'arbre représente un détail de la figure que l'on obtient en composant l'ensemble des transformations "rencontrées" le long de ce chemin. La liste des indices des transformations figurant sur un chemin est appelée adresse. Pour un arbre infini (nombre infini d'itérations) l'adresse est une suite d'indices et correspond à un point de l'attracteur.

Toutes les courbes polynomiales sont autosimilaires et peuvent être représentées par un IFS. Prenons l'exemple d'une courbe B-spline quadratique uniforme définie à partir d'un vecteur de points de contrôle  $P = [P_0, P_1, P_2]$ . L'algorithme de Chaikin montre que l'on peut exprimer la courbe comme union de deux courbes B-spline quadratique uniforme dont les points de contrôle sont  $Q_G = [Q_{1b}, Q_2, Q_{2b}]$  et  $Q_D = [Q_2, Q_{2b}, Q_3]$  où :

$$Q_i = \frac{1}{4}P_{i-1} + \frac{3}{4}P_i \text{ et } Q_{ib} = \frac{3}{4}P_i + \frac{1}{4}P_{i+1}$$

Sont alors définies les deux transformations  $\mathcal{G}$  et  $\mathcal{D}$  qui transforment le polygone de contrôle initial en deux nouveaux polygones de contrôle :

$$\mathcal{G} : P = [P_1, P_2, P_3] \mapsto Q_G = [Q_{1b}, Q_2, Q_{2b}]$$

$$\mathcal{D} : P = [P_1, P_2, P_3] \mapsto Q_D = [Q_2, Q_{2b}, Q_3]$$

Les matrices associées aux transformations  $\mathcal{G}$  et  $\mathcal{D}$  sont respectivement  $M_G$  et  $M_D$  où :

$$M_G = \begin{pmatrix} \frac{3}{4} & \frac{1}{4} & 0 \\ \frac{1}{4} & \frac{3}{4} & 0 \\ 0 & \frac{3}{4} & \frac{1}{4} \end{pmatrix} \quad M_D = \begin{pmatrix} \frac{1}{4} & \frac{3}{4} & 0 \\ 0 & \frac{3}{4} & \frac{1}{4} \\ 0 & \frac{1}{4} & \frac{3}{4} \end{pmatrix}$$

Ainsi,  $PN([0, 1]) = P\mathcal{G}(N[0, 1]) \cup P\mathcal{D}(N[0, 1])$  où  $N(\cdot)$  représente le vecteur de fonctions de base. Ceci étant vrai quel que soit le vecteur de points de contrôle, on a  $N([0, 1]) = \mathcal{G}(N[0, 1]) \cup \mathcal{D}(N[0, 1])$ . Ce qui montre que les fonctions de base  $N(\cdot)$  sont autosimilaires relativement aux transformations  $\mathcal{G}$  et  $\mathcal{D}$ . Ces dernières étant contractantes et par unicité du point fixe, les fonctions de base  $N(\cdot)$  sont l'attracteur de l'IFS  $\{\mathcal{G}, \mathcal{D}\}$ . L'automate CIFS correspondant à l'algorithme de Chaikin est présenté dans la Figure 3.

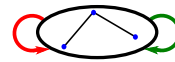


Figure 3: L'automate CIFS de l'algorithme de Chaikin. La transition rouge correspond à la transformation  $\mathcal{G}$  et la transition verte à la transformations  $\mathcal{D}$ .

### 3. Le Blossoming

Les NURBS sont traditionnellement générées à partir de la formule de Cox-De Boor qui est une généralisation de l'algorithme de De Casteljau utilisé pour les courbes de Bézier. Cette formule permet de calculer les fonctions de bases qui serviront à connaître l'influence de chaque point du polygone de contrôle sur la courbe finale.

Introduit par Ramshaw [Ram87], le *blossoming* (appelé également forme polaire ou floraison) est une manière de représenter la formule de Cox-De Boor grâce à de nouvelles fonctions.

Ces fonctions ont plusieurs variables appelées arguments dont le nombre est égal au degré de la courbe désirée. La liste de ces arguments est l'étiquette du point. Ces arguments correspondent à des valeurs du vecteur nodal  $T = [t_0 \dots t_n]$ .

**Symétrie :** l'ordre des arguments n'a aucune importance. Par souci de simplicité, ils seront donc systématiquement notés dans l'ordre du vecteur nodal :

$$\{\dots t_i \dots t_j \dots\} = \{\dots t_j \dots t_i \dots\}$$

**Diagonale :** tout point de la courbe a tous ses arguments identiques. La valeur de cet argument multiple est le paramètre de la courbe associé à ce point :

$$\mathcal{C}(t) = \{t \dots t\}$$

**Multi-affinité :** la fonction est affine pour chaque argument, ce qui veut dire que des interpolations linéaires peuvent être appliquées sur la fonction :

$$\{\dots t \dots\} = \frac{b-t}{b-a} \{\dots a \dots\} + \frac{t-a}{b-a} \{\dots b \dots\}$$

**Consécutivité :** par définition, les points de contrôle sont définis par la floraison  $s$  en  $d$  valeurs consécutives du vecteur nodal  $P_i = s(t_i \dots t_{i+d})$ ; autrement dit tout point du polygone de contrôle est associé à une liste d'arguments consécutifs sur le vecteur nodal :

$$P_i : \{t_i \dots t_{i+d}\}$$

L'objectif étant de calculer un point de la courbe en fonction du polygone de contrôle, il nous faut exprimer  $\mathcal{C}(t) = \{t \dots t\}$  en fonction de  $P_i : \{t_i \dots t_{i+d}\}$ . Pour cela, il faut utiliser la propriété de multi-affinité pour "découper" les arguments les uns après les autres jusqu'à obtenir uniquement des étiquettes correspondant à des points du polygone de contrôle. La structure ainsi créée est de forme arborescente (d'où le nom de *blossoming*) où la racine est le point de la courbe calculé, les branches sont les différents coefficients d'influence et les feuilles sont les points du polygone de contrôle. L'influence d'un chemin qui relie la racine à l'une des feuilles est le produit de l'influence des branches traversées; celui d'un point du polygone de contrôle est la somme

de l'influence des chemins menant à la feuille associée au point. Un exemple de calcul d'un point dans le cas d'une B-Spline quadratique uniforme est donné à la Figure 4.

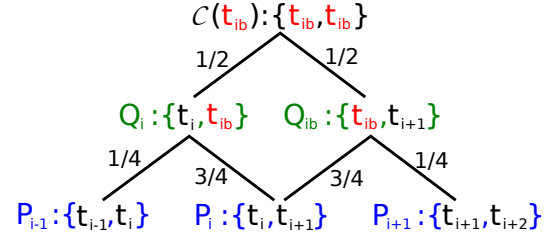


Figure 4: Calcul d'un point d'une courbe quadratique uniforme par *blossoming* dans le cas où la valeur nodale associée au point est le milieu d'un intervalle nodal ( $t_{ib} = \frac{t_i + t_{i+1}}{2}$ ).

Le résultat ainsi obtenu est  $\mathcal{C}(t_{ib}) = \frac{1}{8}P_{i-1} + \frac{6}{8}P_i + \frac{1}{8}P_{i+1}$

Le *blossoming* permet de calculer un point de la courbe  $\mathcal{C}(t)$  mais permet également de calculer le polygone de contrôle  $Q$  associé à un nouveau vecteur nodal qui correspond exactement à la même courbe que celle définie par le couple  $(P; T)$ .

La deuxième ligne de la Figure 4 indique l'équivalence suivante :

$$\begin{aligned} \left\{ \begin{array}{l} [P_{i-1}, P_i, P_{i+1}] \\ [t_{i-1}, t_i, t_{i+1}, t_{i+2}] \end{array} \right\} &\Leftrightarrow \left\{ \begin{array}{l} [P_{i-1}, Q_i, Q_{ib}, P_{i+1}] \\ [t_{i-1}, t_i, t_{ib}, t_{i+1}, t_{i+2}] \end{array} \right\} \\ Q_i &= \frac{1}{4}P_{i-1} + \frac{3}{4}P_i \\ Q_{ib} &= \frac{3}{4}P_i + \frac{1}{4}P_{i+1} \end{aligned}$$

Les coefficients de Chaikin apparaissent : l'insertion au milieu d'un intervalle nodal dans le cas d'une courbe quadratique uniforme correspond à l'algorithme de Chaikin. La forme *blossoming* de la Figure 4 permet la génération de l'automate de la Figure 3. Dans la section suivante, la généralisation de la méthode de construction des automates CIFS pour des cas non-uniforme est présentée.

## 4. Automate CIFS de NURBS

L'objectif est de trouver l'automate CIFS qui correspond au tracé d'une courbe B-Spline Rationnelle Non-Uniforme. Les cas quadratique et cubique seront présentés en détail puis un algorithme de généralisation pour les degrés supérieurs est proposé.

### 4.1. Les NURBS quadratiques

Un morceau de NURBS quadratique est toujours défini par un couple composé d'un polygone de contrôle  $P =$

$[P_1, P_2, P_3]$  et d'un vecteur nodal  $T = [t_1, t_2, t_3, t_4]$ . Les étiquettes des points de contrôle sont dans l'ordre  $\{t_1, t_2\}$ ,  $\{t_2, t_3\}$ , et  $\{t_3, t_4\}$ . Il faut d'abord trouver le polygone de contrôle  $[Q_{1b}, Q_2, Q_{2b}, Q_3]$  associé au vecteur nodal  $[t_{1b} = \frac{t_1+t_2}{2}, t_2, t_{2b} = \frac{t_2+t_3}{2}, t_3, t_{3b} = \frac{t_3+t_4}{2}]$ . Les étiquettes des nouveaux points sont alors  $\{t_{1b}, t_2\}$ ,  $\{t_2, t_{2b}\}$ ,  $\{t_{2b}, t_3\}$ , et  $\{t_3, t_{3b}\}$  et leurs positions sont calculables par *blossoming*.

Les influences des points de  $P$  sur les points de  $Q$  ne dépendant pas de la valeur des noeuds mais des rapports de proportionnalité entre les intervalles nodaux, le vecteur nodal est généralement transformé en vecteur inter-noeud  $U = [u_1, u_2, u_3]$  où  $u_i = t_{i+1} - t_i$ . Ces influences ne sont alors plus exprimées sous la forme d'une différence de valeurs nodales mais comme une somme de longueurs inter-noeuds. Ce vecteur inter-noeud devient avec l'insertion au milieu de chaque noeud le nouveau vecteur inter-noeud  $V = [\frac{u_1}{2}, \frac{u_2}{2}, \frac{u_2}{2}, \frac{u_3}{2}]$  qui est équivalent à  $V = [u_1, u_2, u_2, u_3]$  en terme de rapport de proportionnalité. Cette équivalence est la propriété qui rend le processus stationnaire. Le calcul de l'influence de  $P$  sur  $Q$  en fonction du vecteur inter-noeud par *blossoming* est présenté à la Figure 5.

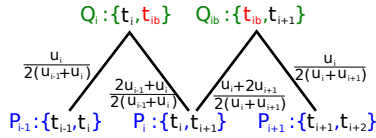


Figure 5: Le calcul par *blossoming* des positions des points du nouveau polygone de contrôle d'une courbe quadratique non-uniforme en fonction des points de l'ancien polygone lors d'une insertion au milieu de chaque noeud

Sont alors définies les deux transformations  $\mathcal{G}$  et  $\mathcal{D}$  qui transforment le couple  $(P; U)$  en deux nouveaux couples  $(Q_{\mathcal{G}}; V_{\mathcal{G}})$  et  $(Q_{\mathcal{D}}; V_{\mathcal{D}})$  :

$$\begin{aligned} \mathcal{G} : \begin{cases} P = [P_1, P_2, P_3] \\ U = [u_1, u_2, u_3] \end{cases} &\mapsto \begin{cases} Q_{\mathcal{G}} = [Q_{1b}, Q_2, Q_{2b}] \\ V_{\mathcal{G}} = [u_1, u_2, u_2] \end{cases} \\ \mathcal{D} : \begin{cases} P = [P_1, P_2, P_3] \\ U = [u_1, u_2, u_3] \end{cases} &\mapsto \begin{cases} Q_{\mathcal{D}} = [Q_2, Q_{2b}, Q_3] \\ V_{\mathcal{D}} = [u_2, u_2, u_3] \end{cases} \end{aligned}$$

Les matrices associées aux transformations  $\mathcal{G}$  et  $\mathcal{D}$  sont respectivement  $M_{\mathcal{G}}$  et  $M_{\mathcal{D}}$ . Les coefficients de celles-ci dépendent du vecteur nodal en entrée; il faut donc générer les matrices en fonction de celui-ci :

$$M_{\mathcal{G}}(u_1, u_2, u_3) = \begin{pmatrix} \frac{u_1+2u_2}{2(u_1+u_2)} & \frac{u_1}{2(u_1+u_2)} & 0 \\ \frac{u_2}{2(u_1+u_2)} & \frac{2u_1+u_2}{2(u_1+u_2)} & 0 \\ 0 & \frac{u_2+2u_3}{2(u_2+u_3)} & \frac{u_2}{2(u_2+u_3)} \end{pmatrix}$$

$$M_{\mathcal{D}}(u_1, u_2, u_3) = \begin{pmatrix} \frac{u_2}{2(u_1+u_2)} & \frac{2u_1+u_2}{2(u_1+u_2)} & 0 \\ 0 & \frac{u_2+2u_3}{2(u_2+u_3)} & \frac{u_2}{2(u_2+u_3)} \\ 0 & \frac{u_3}{2(u_2+u_3)} & \frac{2u_2+u_3}{2(u_2+u_3)} \end{pmatrix}$$

Remarquons que si le vecteur nodal d'entrée est uniforme, les matrices de transformations sont celles de Chaikin. Tout comme pour Chaikin, ces deux transformations sont les transformations de l'IFS mais, dans le cas présent, elles sont dépendantes du vecteur inter-noeud courant. L'automate CIFS a donc plusieurs états qui correspondent aux différentes configurations possible du vecteur inter-noeud. Les coefficients des matrices de transitions dépendent alors du vecteur inter-noeud associé à l'état de départ de la transition. L'automate ainsi obtenu est donné à la Figure 6.

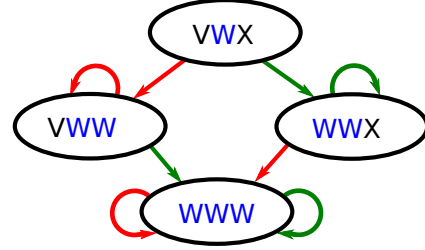


Figure 6: Schéma de l'automate CIFS pour le cas quadratique non-uniforme. Les transitions rouges correspondent aux transformations  $\mathcal{G}$  et les transitions vertes aux transformations  $\mathcal{D}$ . Le vecteur inter-noeud associé à l'état initial est  $[V, W, X]$ .

Le nombre d'état de l'automate dans le cas quadratique est limité à 4 :

- 1 état initial;
- 2 états stationnaires qui bouclent sur eux-mêmes avec toujours la même matrice de transition;
- 1 état uniforme qui est à lui seul l'automate de Chaikin.

#### 4.2. Le cas cubique

Un morceau de NURBS cubique est toujours défini par un couple composé d'un polygone de contrôle  $P = [P_0, P_1, P_2, P_3]$  et d'un vecteur nodal  $T = [t_0, t_1, t_2, t_3, t_4, t_5]$ . Les étiquettes des points sont composées de trois arguments, ce qui donne les deux cas de blossoming présentés à la Figure 7.

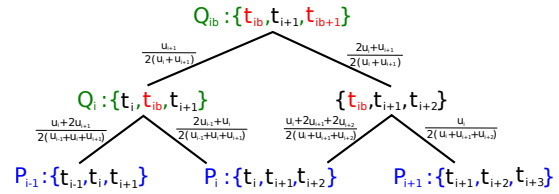


Figure 7: Le calcul par *blossoming* des positions des points du nouveau polygone de contrôle d'une courbe cubique non-uniforme en fonction des points de l'ancien polygone lors d'une insertion au milieu de chaque noeud

Sont alors définies les deux transformations  $\mathcal{G}$  et  $\mathcal{D}$  qui transforment le couple  $(P; U)$  en deux nouveaux couples  $(Q_{\mathcal{G}}; V_{\mathcal{G}})$  et  $(Q_{\mathcal{D}}; V_{\mathcal{D}})$  :

$$\mathcal{G} : \begin{cases} P = [P_0, P_1, P_2, P_3] \\ U = [u_0, u_1, u_2, u_3, u_4] \end{cases} \mapsto \begin{cases} Q_{\mathcal{G}} = [Q_{1b}, Q_2, Q_{2b}, Q_3] \\ V_{\mathcal{G}} = [u_1, u_1, u_2, u_2, u_3] \end{cases}$$

$$\mathcal{D} : \begin{cases} P = [P_0, P_1, P_2, P_3] \\ U = [u_0, u_1, u_2, u_3, u_4] \end{cases} \mapsto \begin{cases} Q_{\mathcal{D}} = [Q_2, Q_{2b}, Q_3, Q_{3b}] \\ V_{\mathcal{D}} = [u_1, u_2, u_2, u_3, u_3] \end{cases}$$

Ces deux transformations correspondent à deux matrices de transitions qui dépendent du vecteur inter-noeud associé à l'état de départ de la transition. L'automate ainsi obtenu est donné à la Figure 8.

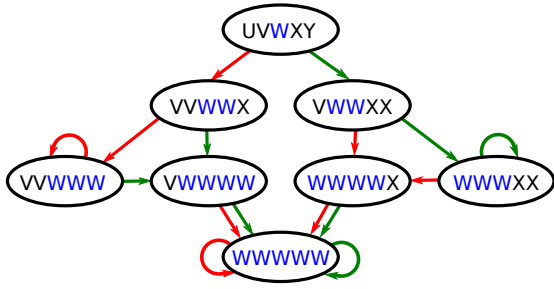


Figure 8: Schéma de l'automate CIFS pour le cas cubique non-uniforme. Les transitions rouges correspondent aux transformations  $\mathcal{G}$  et les transitions vertes aux transformations  $\mathcal{D}$ . Le vecteur inter-noeud associé à l'état initial est  $[U, V, W, X, Y]$ .

Le nombre d'état de l'automate dans le cas cubique est plus conséquent que dans le cas quadratique :

- 1 état initial ;
- 2 états stationnaires qui bouclent sur eux-même avec toujours la même matrice de transition ;
- 1 état uniforme ;
- 4 états de transition.

Il est important de noter que la seule différence avec le cas quadratique est le nombre d'états de transition (4 au lieu de 0) ; le nombre d'états stationnaires et uniformes reste identique.

#### 4.3. Généralisation au degré supérieur

Itérer sur le polygone de contrôle correspond à une insertion au milieu de chaque intervalle nodal ce qui revient à dédoubler chaque valeur du vecteur inter-noeud. Les nouveaux vecteur inter-noeud  $V_{\mathcal{G}}$  et  $V_{\mathcal{D}}$  des transformations  $\mathcal{G}$  et  $\mathcal{D}$  doivent être de même longueur que le vecteur inter-noeud  $U$ , c'est à dire  $2d - 1$  où  $d$  est le degré de la courbe. Ils doivent également être centrés sur le même inter-noeud que  $U$  ; celui-ci étant doublé,  $V_{\mathcal{G}}$  est centré sur la partie gauche

et  $V_{\mathcal{D}}$  sur la partie droite. Pour les degrés de 1 à 4, les résultats sont les suivants (en rouge l'inter-noeud exclusif à  $V_{\mathcal{G}}$ , en vert celui exclusif à  $V_{\mathcal{D}}$  et en bleu les communs) :

$$\begin{array}{lll} \text{degré 1 :} & W & \mapsto \text{WW} \\ \text{degré 2 :} & VWX & \mapsto \text{VVWWXX} \\ \text{degré 3 :} & UVWXY & \mapsto \text{UUUVWWXXYY} \\ \text{degré 4 :} & TUVWXYZ & \mapsto \text{TTUUVVWWXXYYZZ} \end{array}$$

La récurrence qui apparaît est la suivante : chaque inter-noeud est dédoublé, les  $d - 1$  inter-noeuds extérieurs sont éliminés et les deux nouveaux vecteurs ont un inter-noeud qui leur est exclusif (le premier à gauche et à droite) et tous les autres inter-noeuds sont communs aux deux nouveaux vecteurs. En itérant sur cette règle, les différents états de l'automate sont décrits. De plus, si un état est obtenu à partir d'un autre, cela signifie qu'il y a une transition entre les deux états.

Une autre méthode de génération des états uniquement basée sur le dédoublement des inter-noeuds est possible. Soit  $d$  le degré de la courbe,  $n = \lceil \log_2(2d - 1) \rceil$  est le nombre maximal d'itérations nécessaires pour générer tous les états. Pour chaque itération  $i \in [0; n]$  un vecteur contenant chaque inter-noeud  $2^i$  fois consécutives est créé. Dans ce vecteur, toutes les séquences de  $2d - 1$  inter-noeuds consécutifs qui ont le même inter-noeud central que l'état de départ sont des états valables. Le nombre d'états par itération est donc de  $\min(2d - 1; 2^i)$ . Pour finir, les doublons sont supprimés.

Par exemple, pour le cas d'une NURBS de degré 4, les 12 états distincts du vecteur inter-noeud (centré sur  $\underline{W}$ ) sont les suivants :

- $i = 0 \Rightarrow 1$  état
  - TUVWXYZ
- $i = 1 \Rightarrow 2$  états
  - UVVWXX
  - VVWXXY
- $i = 2 \Rightarrow 4$  états
  - VVVWWW
  - VVWWWX
  - VWWWXX
  - WWWXXX
- $i = 3 \Rightarrow 7$  états
  - VVVWWWW (doublon)
  - VVWWWW
  - VWWWW
  - WWWW
  - WWWWXX
  - WWWWXXX (doublon)

Une fois l'automate CIFS constitué, il faut calculer les matrices de transition. Les états de l'automate à l'itération  $i = 1$  sont les vecteurs  $V_{\mathcal{G}}$  et  $V_{\mathcal{D}}$ . En exprimant les différentes étiquettes de ces vecteurs en fonction des étiquettes de  $U$  par *blossoming*, les influences des points de  $P$  sur  $Q_{\mathcal{G}}$  et  $Q_{\mathcal{D}}$  sont calculées.

#### 5. Automate CIFS de surfaces NURBS

Les surfaces NURBS générées par produit-tensoriel de NURBS peuvent également être définies par un automate CIFS. L'automate  $S$  associé à la surface est déduit des automates  $A$  et  $B$  associés aux deux courbes et par abus de langage, sera désigné comme le "produit tensoriel" des automates de courbe [GHG12].



L'idée principale est que chaque état de  $S$  correspond à la fois à un état de  $A$  et à un état de  $B$ . L'état initial de  $S$  est donc l'état initial de  $A$  et l'état initial de  $B$ . Chaque transition  $T_S$  de  $S$  correspond à des transitions  $T_A$  et  $T_B$  parallèles sur les automates  $A$  et  $B$ , la matrice de transition  $M_S$  est le produit des matrices de transitions  $M_A$  et de  $M_B$ . L'état d'arrivée de chaque transition  $T_S$  est également l'état d'arrivée de  $T_A$  et celui de  $T_B$ . Un exemple est donné en Figure 9

La méthode est répétable pour générer des volumes ou autres objets de dimension supérieure : tant qu'un objet est généré par produit tensoriel de NURBS, il peut l'être par le "produit-tensoriel" des automates CIFS associés aux courbes.

## 6. Applications

Le formalisme des CIFS permet de représenter les NURBS ainsi que tous les objets générés par produit tensoriel de celles-ci. Comme l'a montré [PGSL14], des objets de natures différentes sont combinables s'ils sont tous représentés sous forme de CIFS. Les surfaces de subdivision étant générables par un automate CIFS, il est donc envisageable de créer des maillages CAO où elles cohabiteraient avec les NURBS. L'intérêt de ce type de maillage hybride est de pouvoir profiter de la capacité des surfaces de subdivision à gérer les topologies arbitraires pour compléter la surface générée par les NURBS sur les grilles régulières sans passer par une étape de *tuning*. Les raccords entre les deux types de surfaces seraient alors assurés continus par la méthode de [PGSL14].

[GHG12] a présenté des structures arborescentes permettant d'assurer le support d'un objet avec une structure lacunaire, solide mais légère. Pour cela, il fallait que la surface de l'objet soit représentée sous forme d'automate CIFS. Avec la formulation des surfaces NURBS sous forme d'automate CIFS, ce genre de structure peut être généré pour des objets conçu avec des NURBS dans un système CAO.

## 7. Conclusion

Dans cet article nous avons présenté la méthode permettant de formuler les NURBS, ainsi que tous les objets générés par produit tensoriel de celles-ci, sous forme d'automate CIFS. Les objets générés par NURBS étant courants en CAO, cette reformulation permet de les combiner avec des objets générables par CIFS plus inhabituels dans ce milieu comme les structures fractales.

La principale limite de la méthode présentée est l'absence de la gestion des surfaces NURBS autour des sommets de valence irrégulière. L'intégration des surfaces définies par [CADSO9] à la liste des objets IFS fera l'objet de futurs travaux.

## Références

- [Bar14] BARNSEY M. F. : *Fractals everywhere*. Academic press, 2014.
- [CADSO9] CASHMAN T. J., AUGSDÖRFER U. H., DODGSON N. A., SABIN M. A. : Nurbs with extraordinary points : high-degree, non-uniform, rational subdivision schemes. In *ACM Transactions on Graphics (TOG)* (2009), vol. 28, ACM, p. 46.
- [Cha74] CHAIKIN G. M. : An algorithm for high-speed curve generation. *Computer graphics and image processing*. Vol. 3, Num. 4 (1974), 346–349.
- [GHG12] GOUATY G., HOUSAM H., GENTIL C. : Modélisation à base de règles et de combinaisons de règles topologiques. *Journées du Groupe de Travail en Modélisation Géométrique 2012, Strasbourg* (2012), 29–38.
- [Hut81] HUTCHINSON J. E. : Fractals and self similarity. *Indiana University Mathematics Journal*. Vol. 30, Num. 5 (1981), 713–747.
- [PGSL14] PODKORYTOV S., GENTIL C., SOKOLOV D., LANQUETIN S. : *Joining Primal/Dual Subdivision Surfaces*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 403–424.
- [Ram87] RAMSHAW L. : *Blossoming : A connect-the-dots approach to splines*. Digital Equipment Corporation Palo Alto, 1987.
- [SGGM15] SOKOLOV D., GOUATY G., GENTIL C., MISHKINIS A. : *Boundary Controlled Iterated Function Systems*. Springer International Publishing, Cham, 2015, pp. 414–432.
- [SLG05] SCHAEFER S., LEVIN D., GOLDMAN R. : Subdivision schemes and attractors. In *Proceedings of the Third Eurographics Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2005), SGP '05, Eurographics Association.
- [WW01] WARREN J., WEIMER H. : *Subdivision Methods for Geometric Design : A Constructive Approach*, 1st ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001.

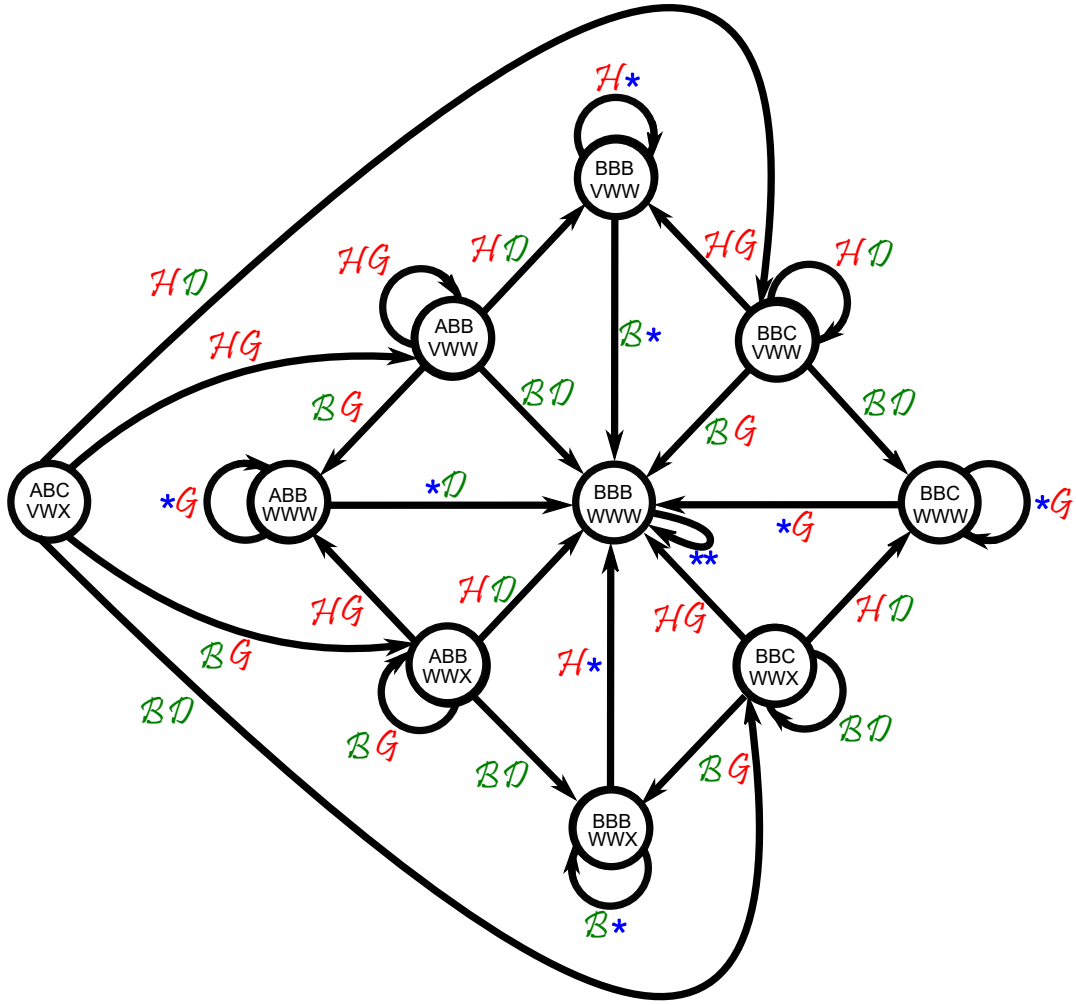


Figure 9: L'automate CIFS d'une surface NURBS quadratique dont les vecteurs nodaux de départ sont  $[A, B, C]$  et  $[V, W, X]$ . Les transformations  $\mathcal{G}$  et  $\mathcal{D}$  agissent sur le vecteur  $[A, B, C]$  et les transformations  $\mathcal{H}$  et  $\mathcal{B}$  sont leur équivalent pour le vecteur  $[V, W, X]$ . Chaque transition correspond à deux transformations : une sur chaque vecteur. La transformation  $\star$  représente une transformation quelconque. L'état final  $[B, B, B] - [W, W, W]$  est uniforme et boucle sur lui-même.