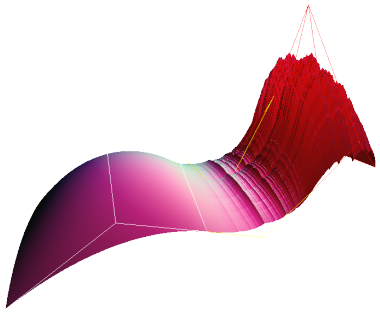
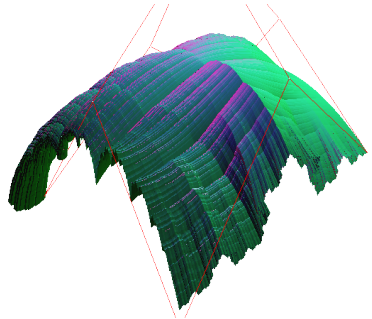
Representation of NURBS surfaces by Controlled Iterated Functions System automata

Lucas Morlet, Christian Gentil, Sandrine Lanquetin, Marc Neveu, Jean-Luc Baril

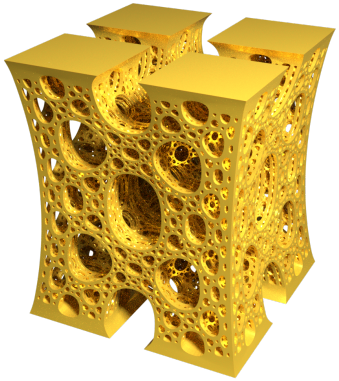


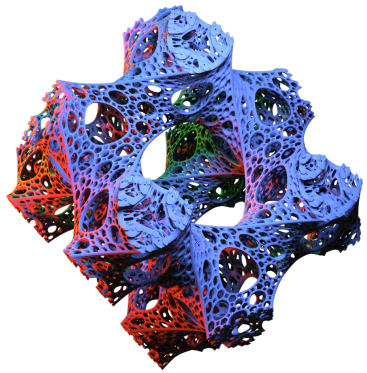
Laboratoire Informatique de Bourgogne

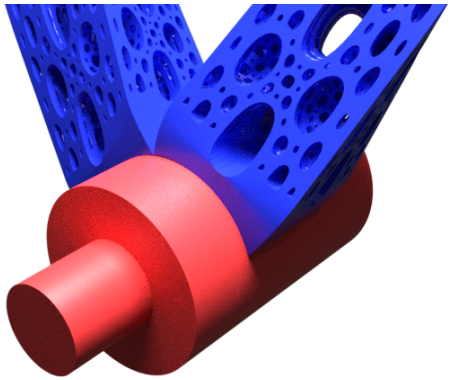
Université de Bourgogne Franche-Comté

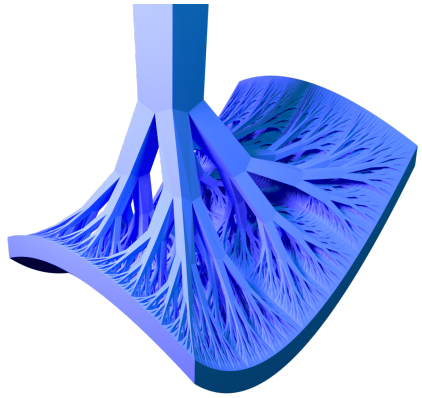


SMI 2019 - Vancouver

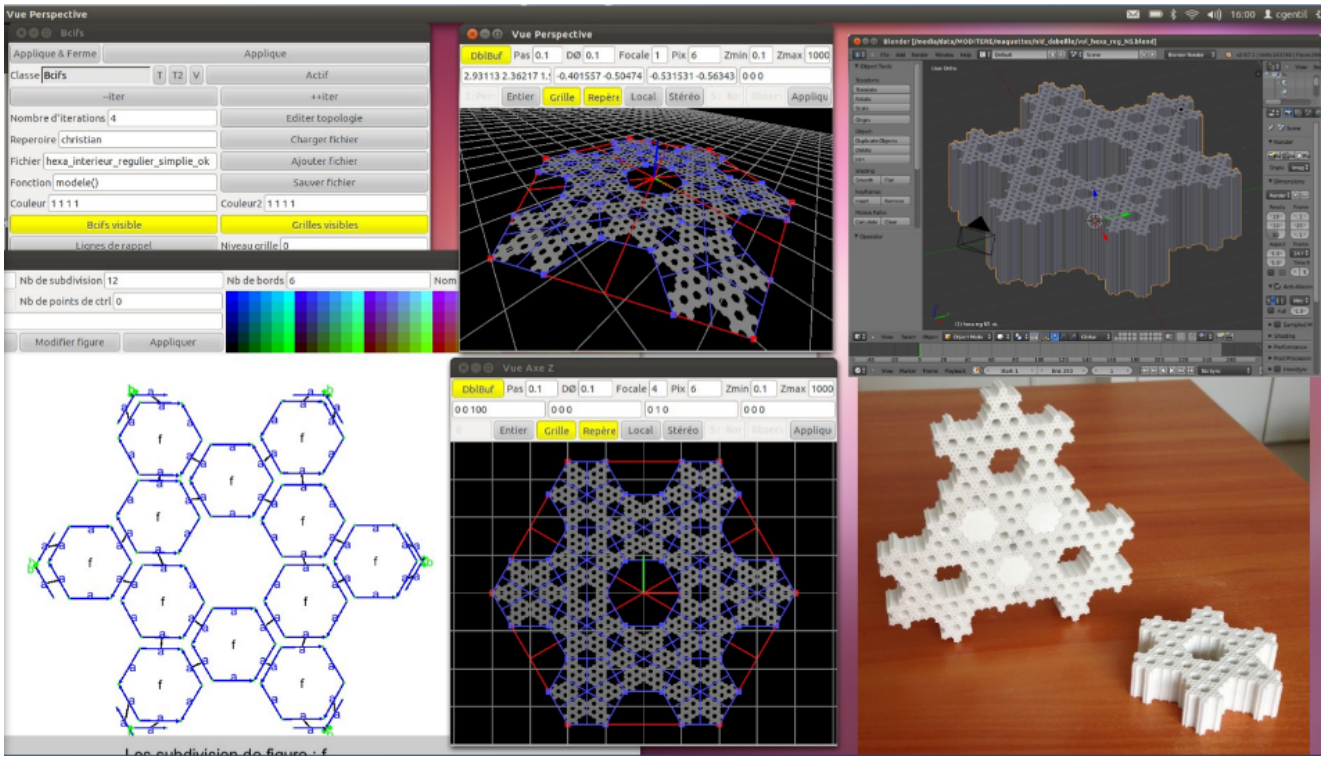








Our iterative geometric modeller



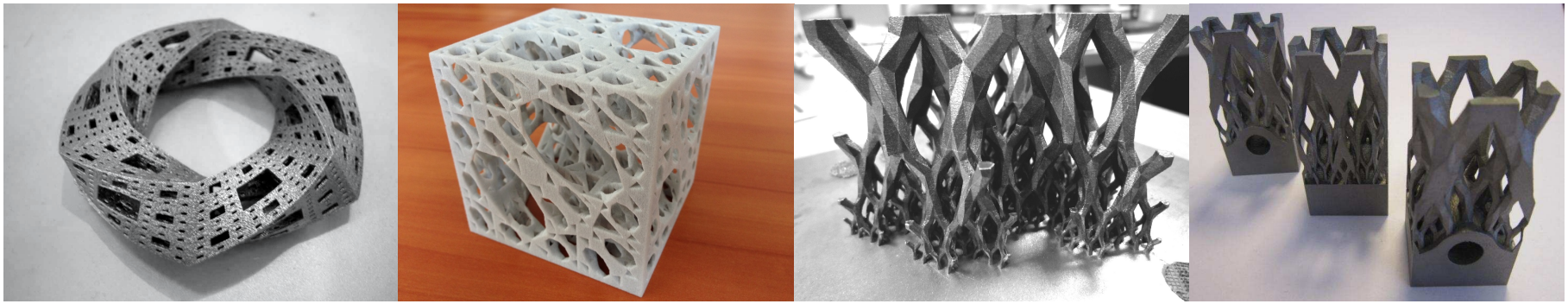


Table of contents

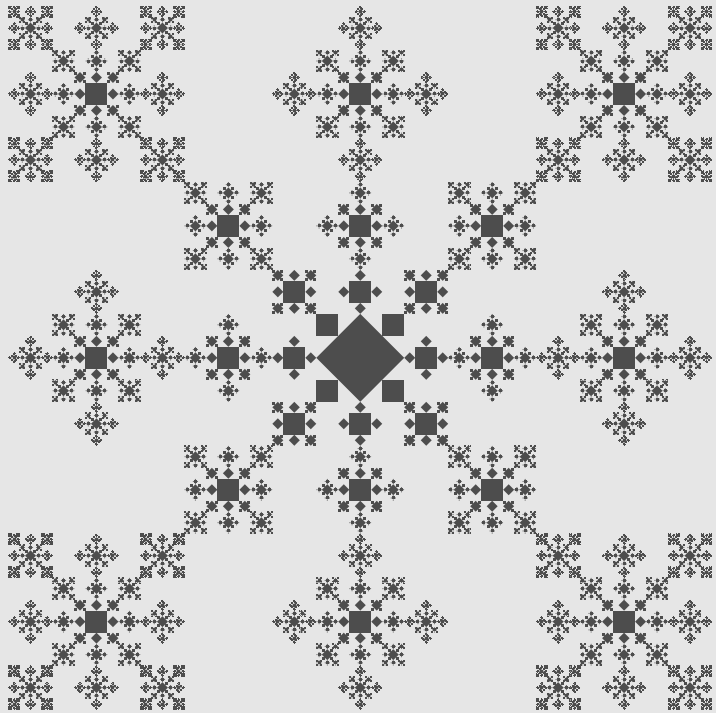
IFS - Iterated Function System

Hutchinson operator (1981)

$$\mathcal{H}(K) = \bigcup_{i \in \Sigma} T_i(K).$$

- T_i a contraction ($i \in \Sigma$)
- K a compact / an object (list of points representing a polygon, a mesh,...)

We iterate: $\mathcal{H}^n(K)$



IFS - Iterated Function System

Hutchinson operator (1981)

$$\mathcal{H}(K) = \bigcup_{i \in \Sigma} T_i(K).$$

- T_i a contraction ($i \in \Sigma$)
- K a compact / an object (list of points representing a polygon, a mesh,...)

We iterate: $\mathcal{H}^n(K)$

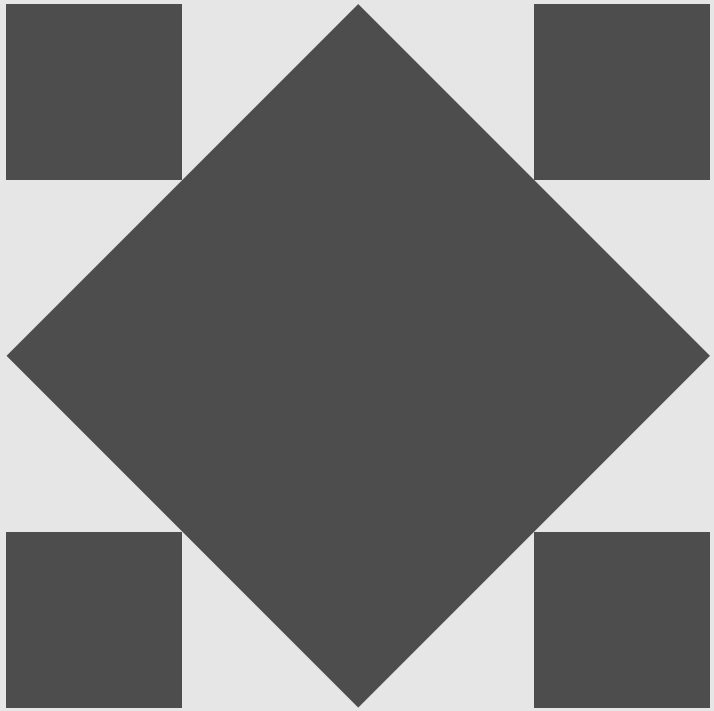
IFS - Iterated Function System

Hutchinson operator (1981)

$$\mathcal{H}(K) = \bigcup_{i \in \Sigma} T_i(K).$$

- T_i a contraction ($i \in \Sigma$)
- K a compact / an object (list of points representing a polygon, a mesh,...)

We iterate: $\mathcal{H}^n(K)$



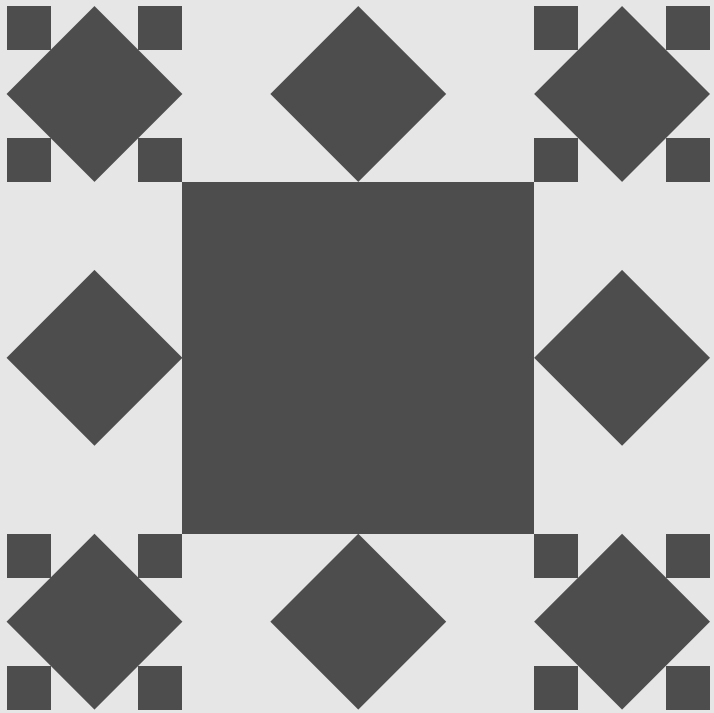
IFS - Iterated Function System

Hutchinson operator (1981)

$$\mathcal{H}(K) = \bigcup_{i \in \Sigma} T_i(K).$$

- T_i a contraction ($i \in \Sigma$)
- K a compact / an object (list of points representing a polygon, a mesh,...)

We iterate: $\mathcal{H}^n(K)$



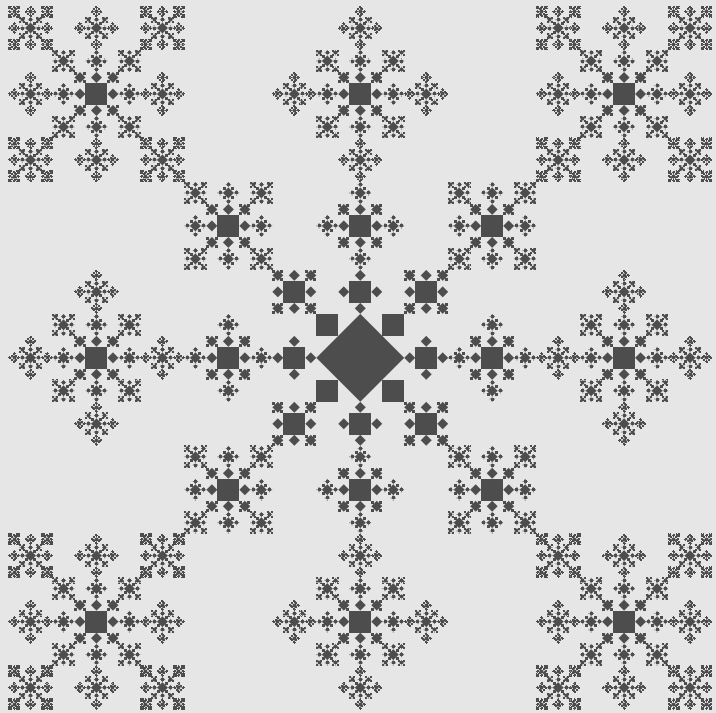
IFS - Iterated Function System

Hutchinson operator (1981)

$$\mathcal{H}(K) = \bigcup_{i \in \Sigma} T_i(K).$$

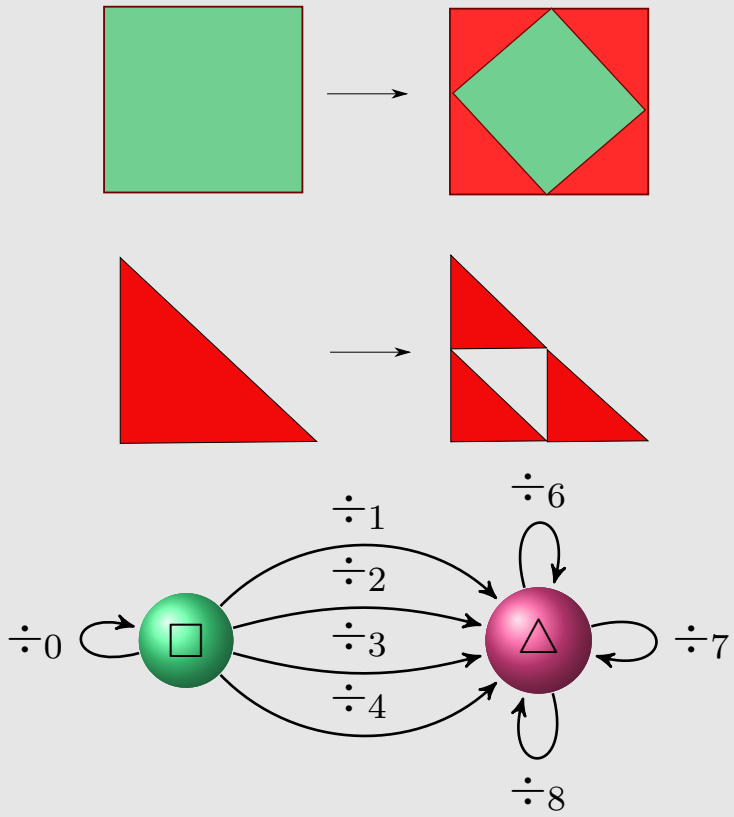
- T_i a contraction ($i \in \Sigma$)
- K a compact / an object (list of points representing a polygon, a mesh,...)

We iterate: $\mathcal{H}^n(K)$

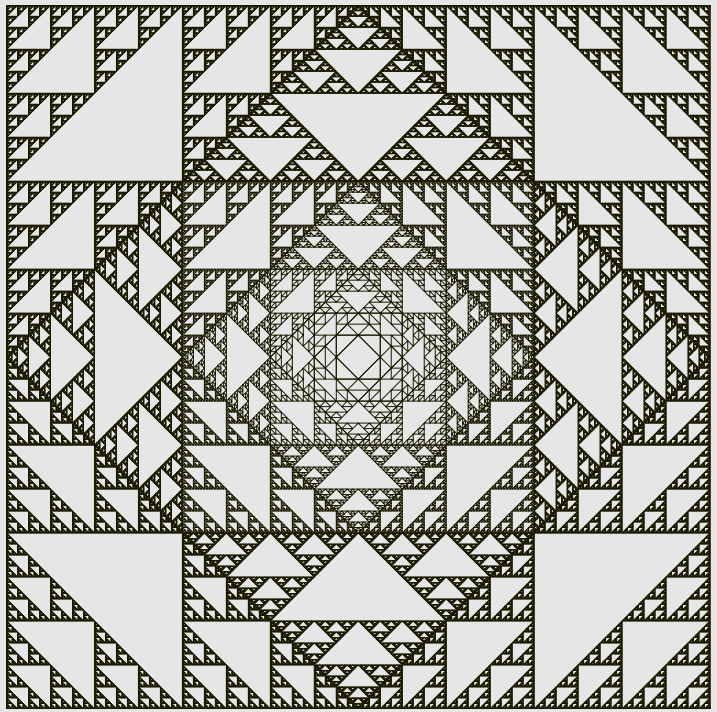


Control of the subdivision process : Controlled-IFS

Automaton

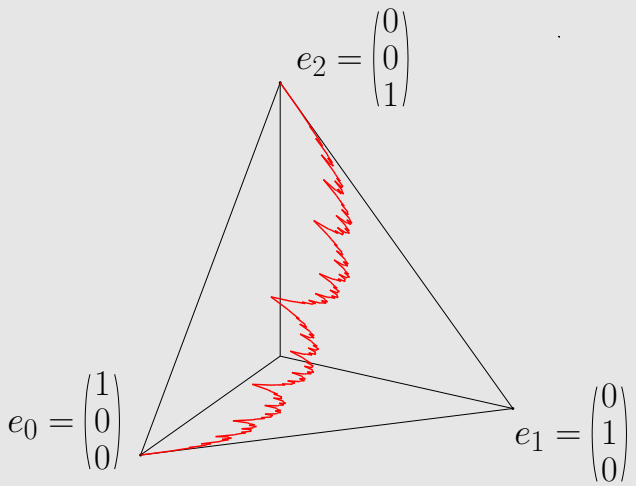


Iteration process according to the automaton

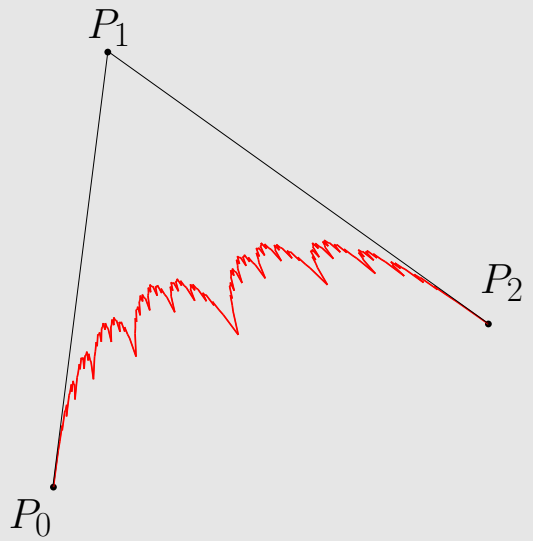


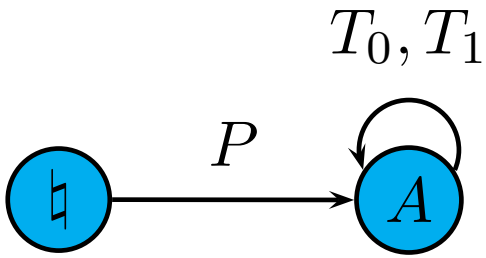
Free form fractal

$A =$ attractor of $\{T_0, T_1\}$ built in a barycentic space



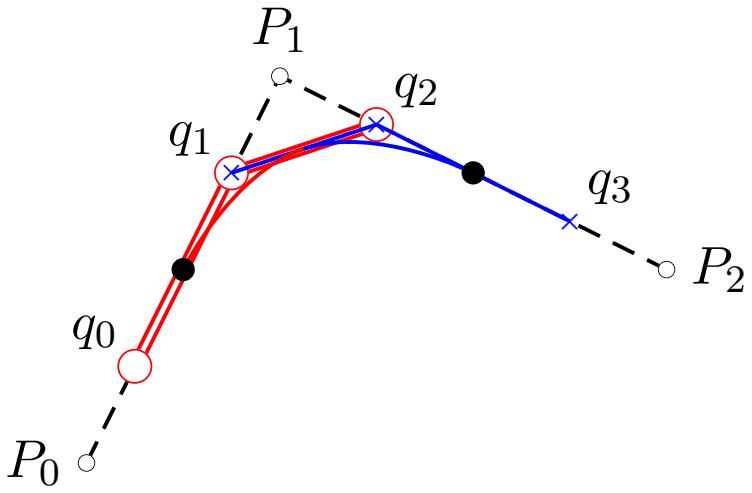
Projection of A : $(P_0P_1P2)A$





$$P = (P_0P_1P_2)$$

Example : Uniform quadratic Spline - Chaikin algorithm

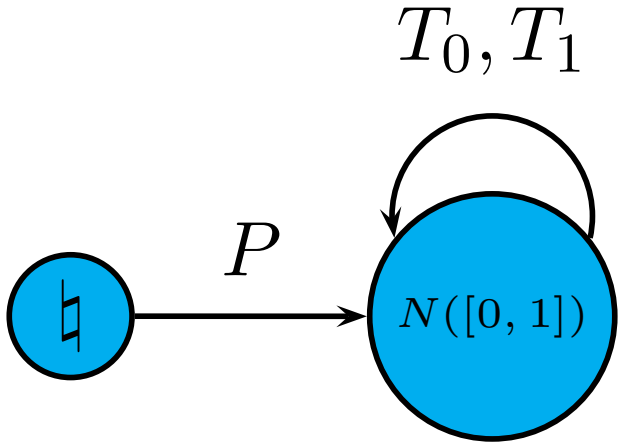


Consequently

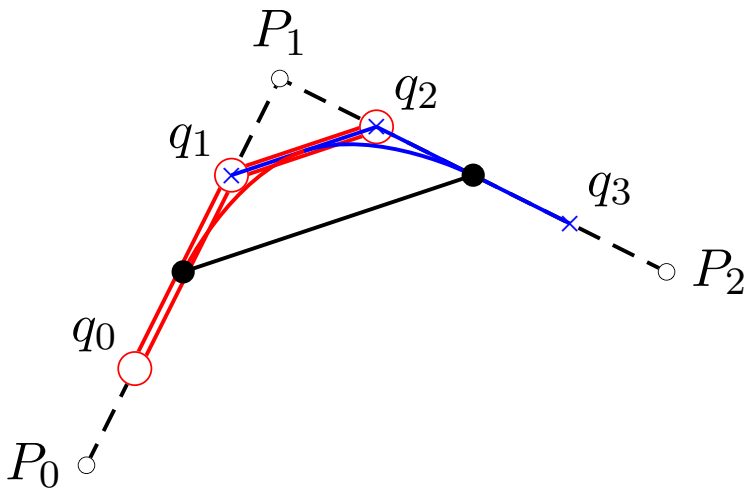
$C([0,1]) = PN([0,1])$
$N([0,1]) = T_0 N([0,1]) \bigcup T_1 N([0,1])$
$N([0,1])$ =Attractor of $\{T_0, T_1\}$

$$T_0 = \begin{pmatrix} 3/4 & 1/4 & 0 \\ 1/4 & 3/4 & 3/4 \\ 0 & 0 & 1/4 \end{pmatrix}$$

$$T_1 = \begin{pmatrix} 1/4 & 0 & 0 \\ 3/4 & 3/4 & 1/4 \\ 0 & 1/4 & 3/4 \end{pmatrix}$$



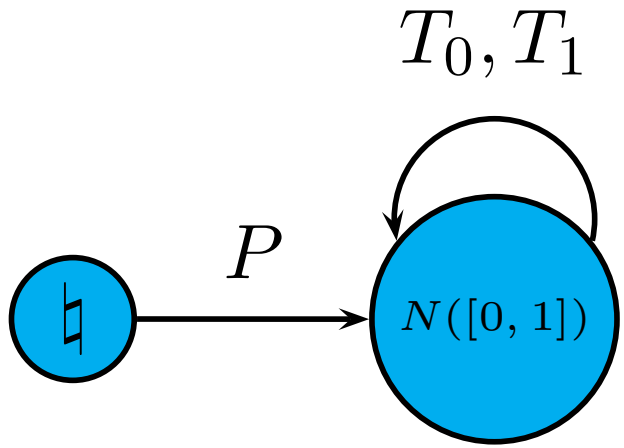
Example : Uniform quadratic Spline - Chaikin algorithm



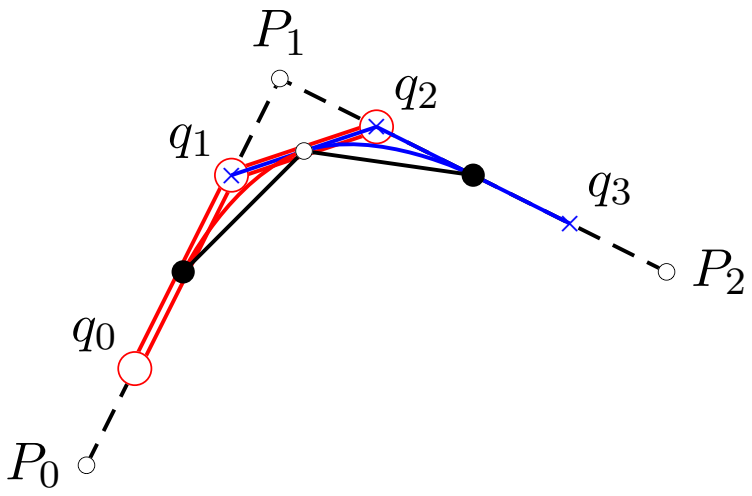
$$K = \begin{pmatrix} 1/2 & 0 \\ 1/2 & 1/2 \\ 0 & 1/2 \end{pmatrix}$$

$$T_0 = \begin{pmatrix} 3/4 & 1/4 & 0 \\ 1/4 & 3/4 & 3/4 \\ 0 & 0 & 1/4 \end{pmatrix}$$

$$T_1 = \begin{pmatrix} 1/4 & 0 & 0 \\ 3/4 & 3/4 & 1/4 \\ 0 & 1/4 & 3/4 \end{pmatrix}$$



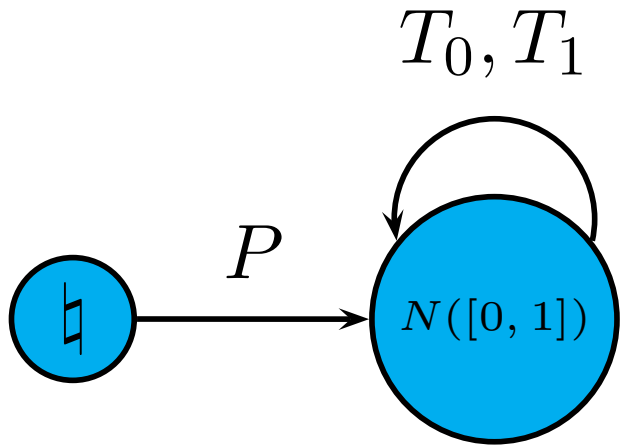
Example : Uniform quadratic Spline - Chaikin algorithm



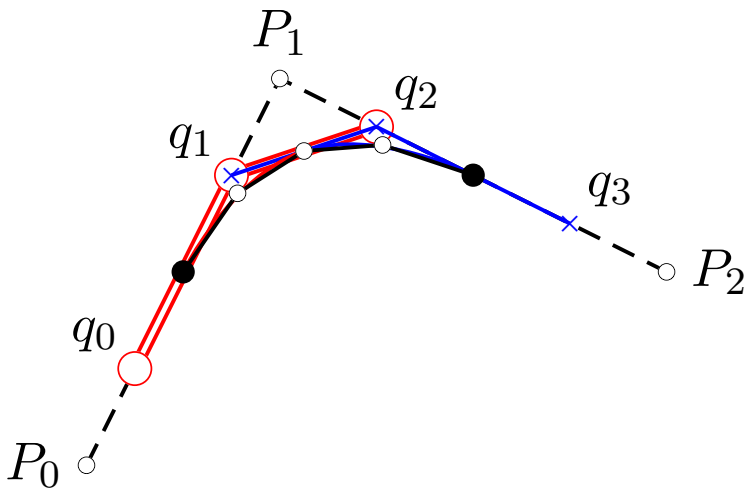
$$K = \begin{pmatrix} 1/2 & 0 \\ 1/2 & 1/2 \\ 0 & 1/2 \end{pmatrix}$$

$$T_0 = \begin{pmatrix} 3/4 & 1/4 & 0 \\ 1/4 & 3/4 & 3/4 \\ 0 & 0 & 1/4 \end{pmatrix}$$

$$T_1 = \begin{pmatrix} 1/4 & 0 & 0 \\ 3/4 & 3/4 & 1/4 \\ 0 & 1/4 & 3/4 \end{pmatrix}$$



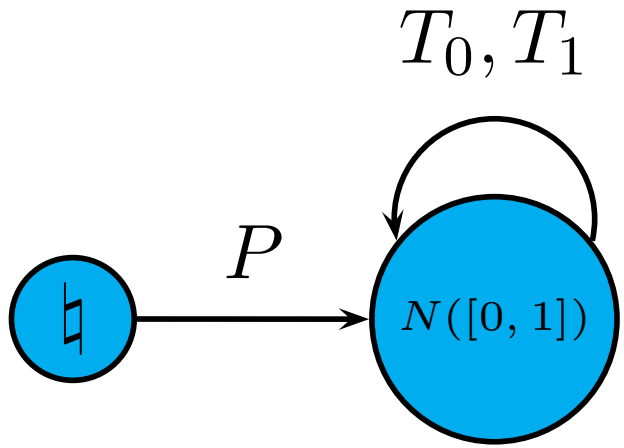
Example : Uniform quadratic Spline - Chaikin algorithm



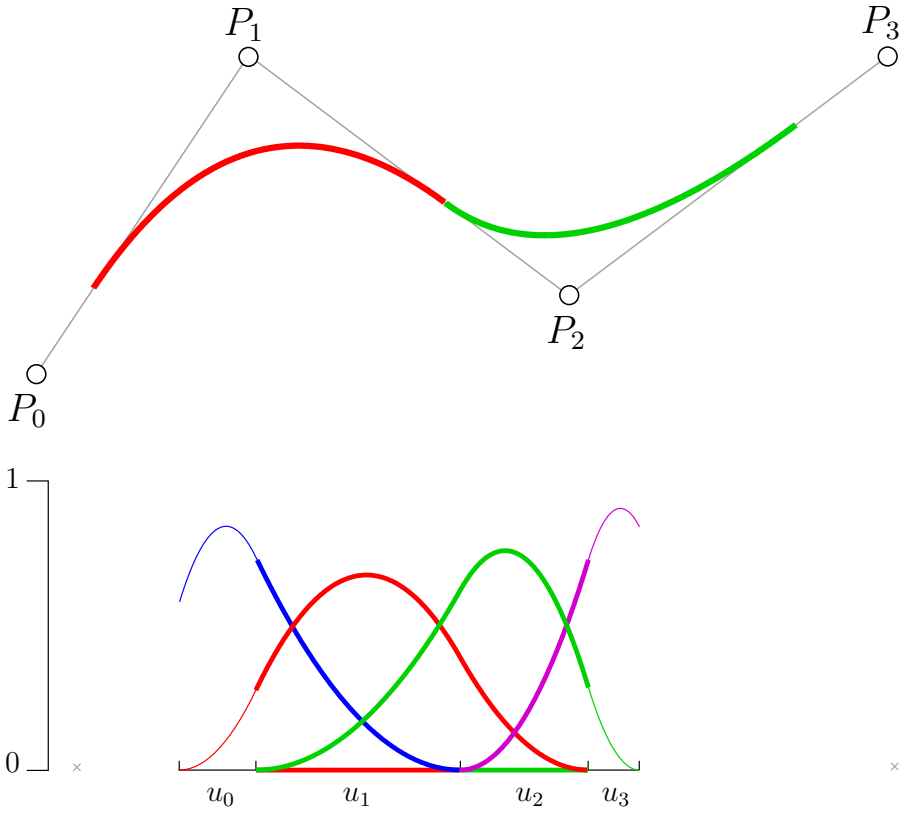
$$K = \begin{pmatrix} 1/2 & 0 \\ 1/2 & 1/2 \\ 0 & 1/2 \end{pmatrix}$$

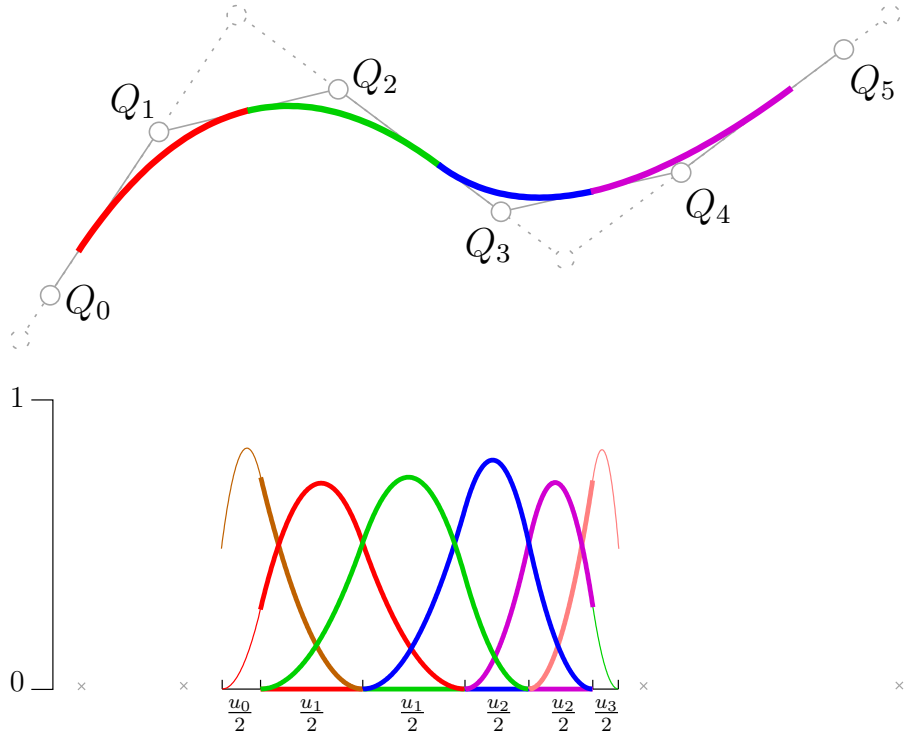
$$T_0 = \begin{pmatrix} 3/4 & 1/4 & 0 \\ 1/4 & 3/4 & 3/4 \\ 0 & 0 & 1/4 \end{pmatrix}$$

$$T_1 = \begin{pmatrix} 1/4 & 0 & 0 \\ 3/4 & 3/4 & 1/4 \\ 0 & 1/4 & 3/4 \end{pmatrix}$$



NURBS : node insertion algorithm



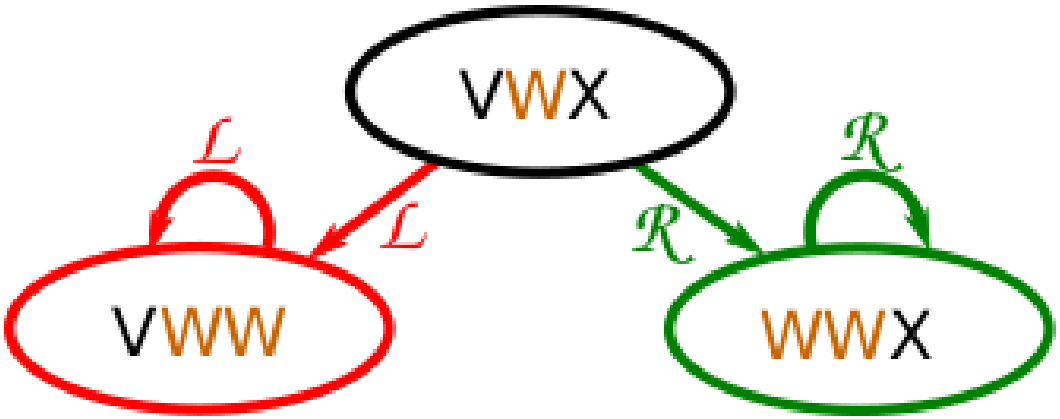


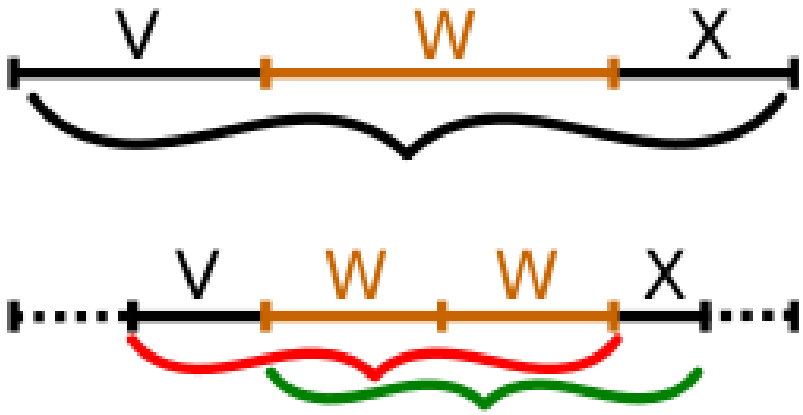
NURBS : node insertion algorithm

Initial inter-nodal vector:

NURBS : node insertion algorithm

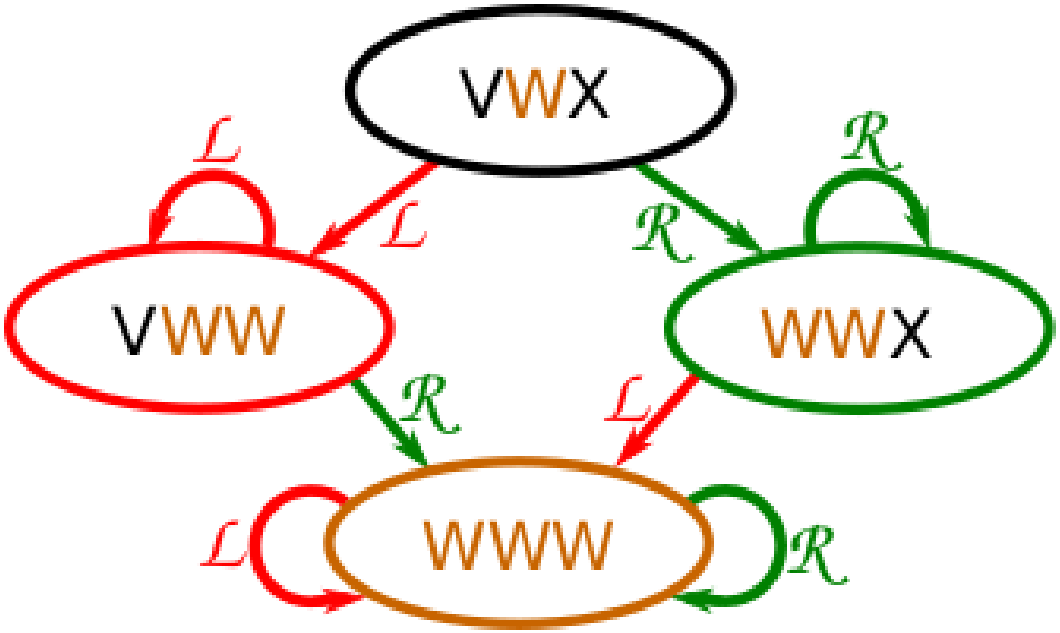
After 1 inter-nodes duplication:

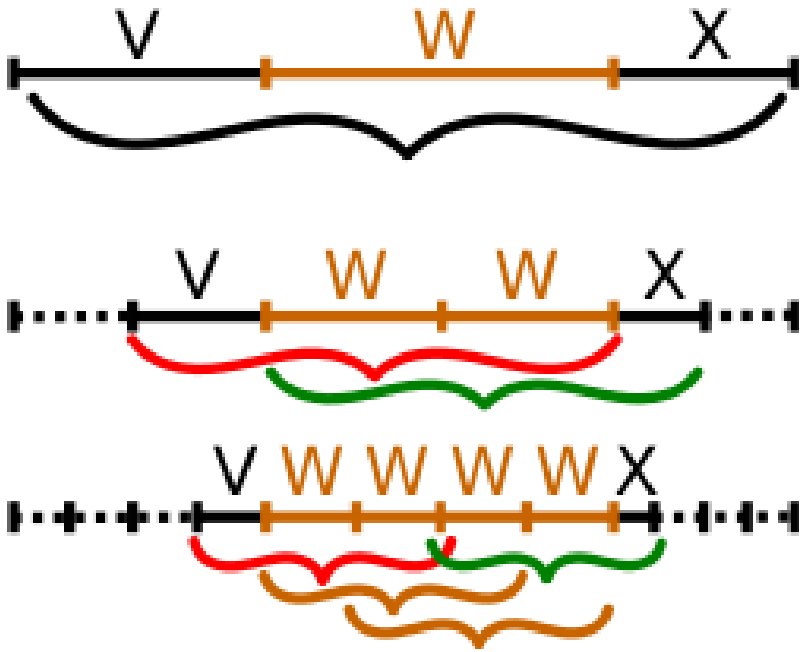




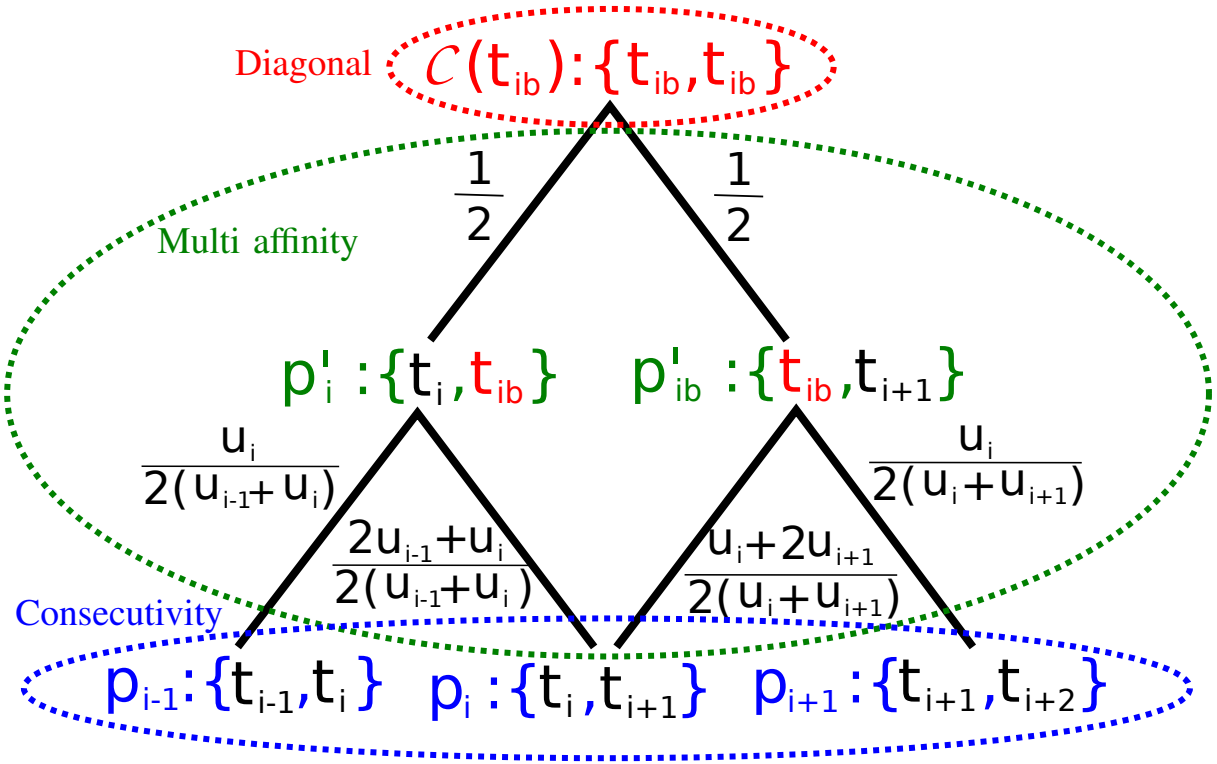
NURBS : node insertion algorithm

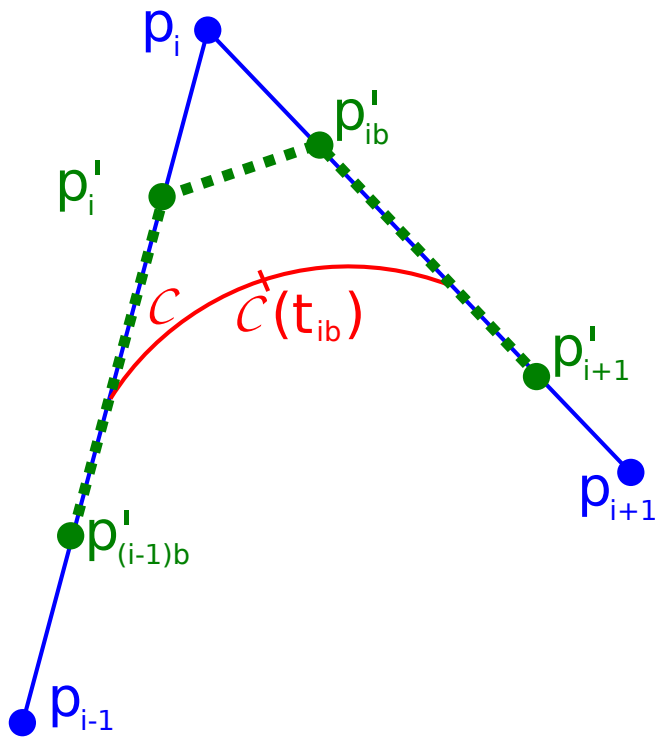
After 2 inter-nodes duplications:





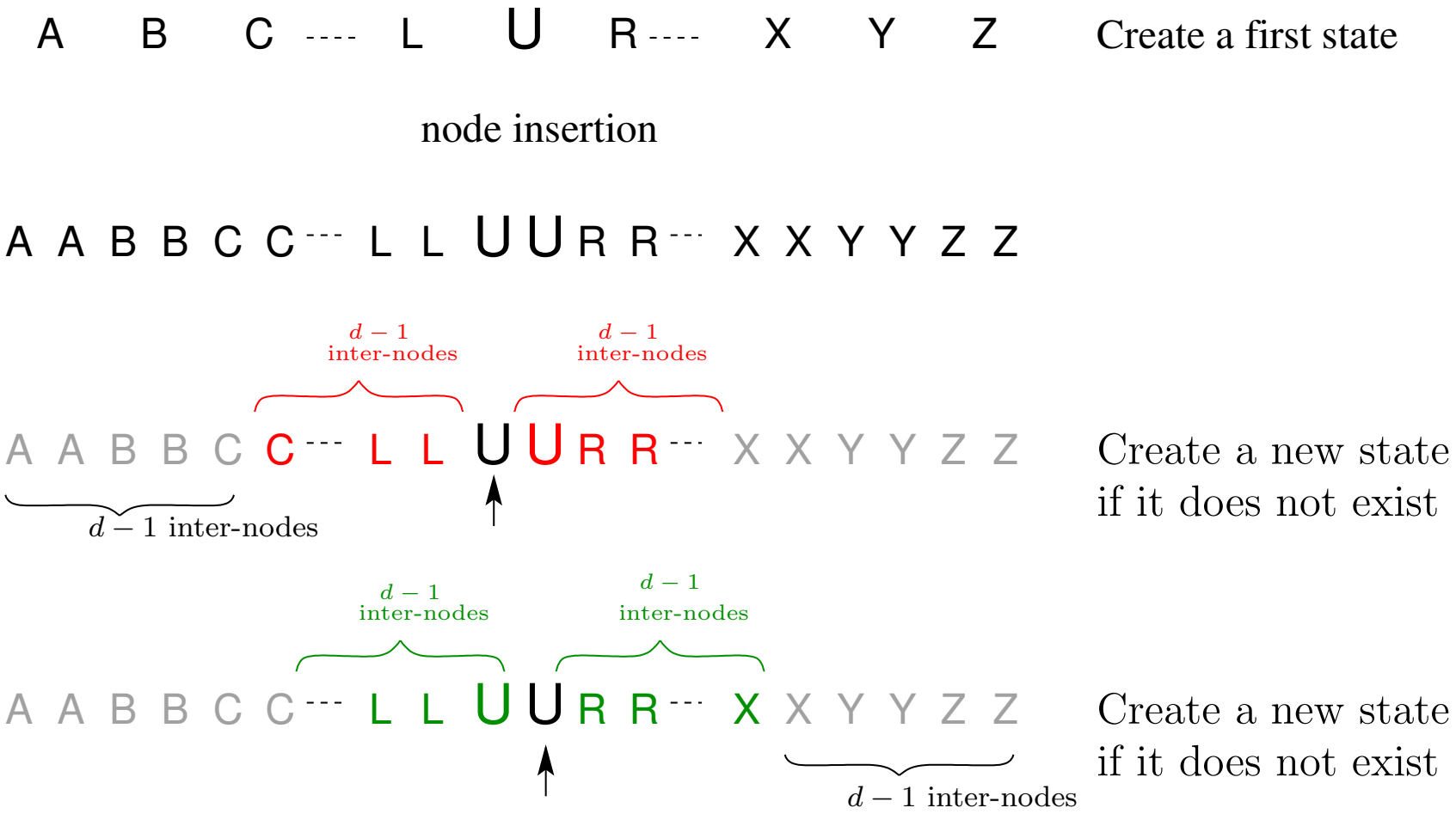
Computation of the subdivision matrices = blossoming





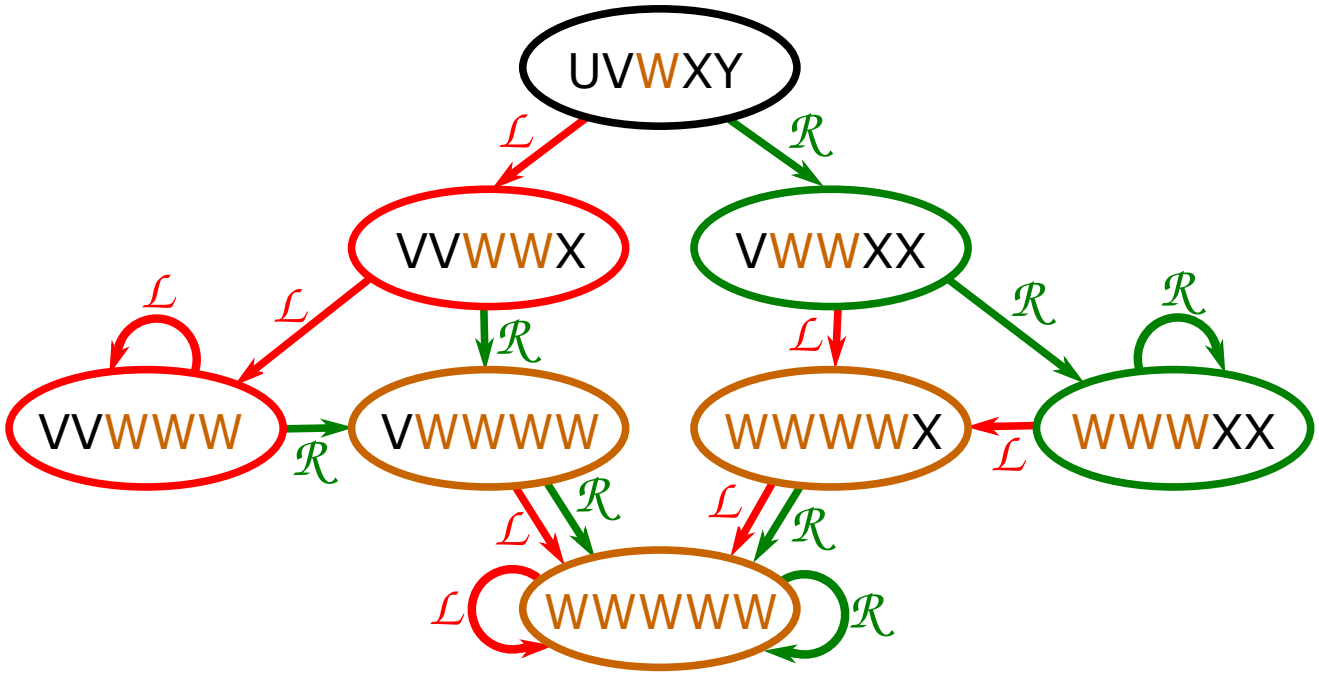
$$\mathcal{M}_{\mathcal{L}}(u_1,u_2,u_3)=\begin{pmatrix}\frac{u_1+2u_2}{2(u_1+u_2)} & \frac{u_2}{2(u_1+u_2)} & 0\\ \frac{u_1}{2(u_1+u_2)} & \frac{2u_1+u_2}{2(u_1+u_2)} & \frac{u_2+2u_3}{2(u_2+u_3)}\\ 0 & 0 & \frac{u_2}{2(u_2+u_3)}\end{pmatrix},\ \mathcal{M}_{\mathcal{R}}(u_1,u_2,u_3)=\begin{pmatrix}\frac{u_2}{2(u_1+u_2)} & 0 & 0\\ \frac{2u_1+u_2}{2(u_1+u_2)} & \frac{u_2+2u_3}{2(u_2+u_3)} & \frac{u_3}{2(u_2+u_3)}\\ 0 & \frac{u_2}{2(u_2+u_3)} & \frac{2u_2+u_3}{2(u_2+u_3)}\end{pmatrix}$$

Generalization to any degree d



Subdivision matrices can be calculated using blossoming.

Properties of automata

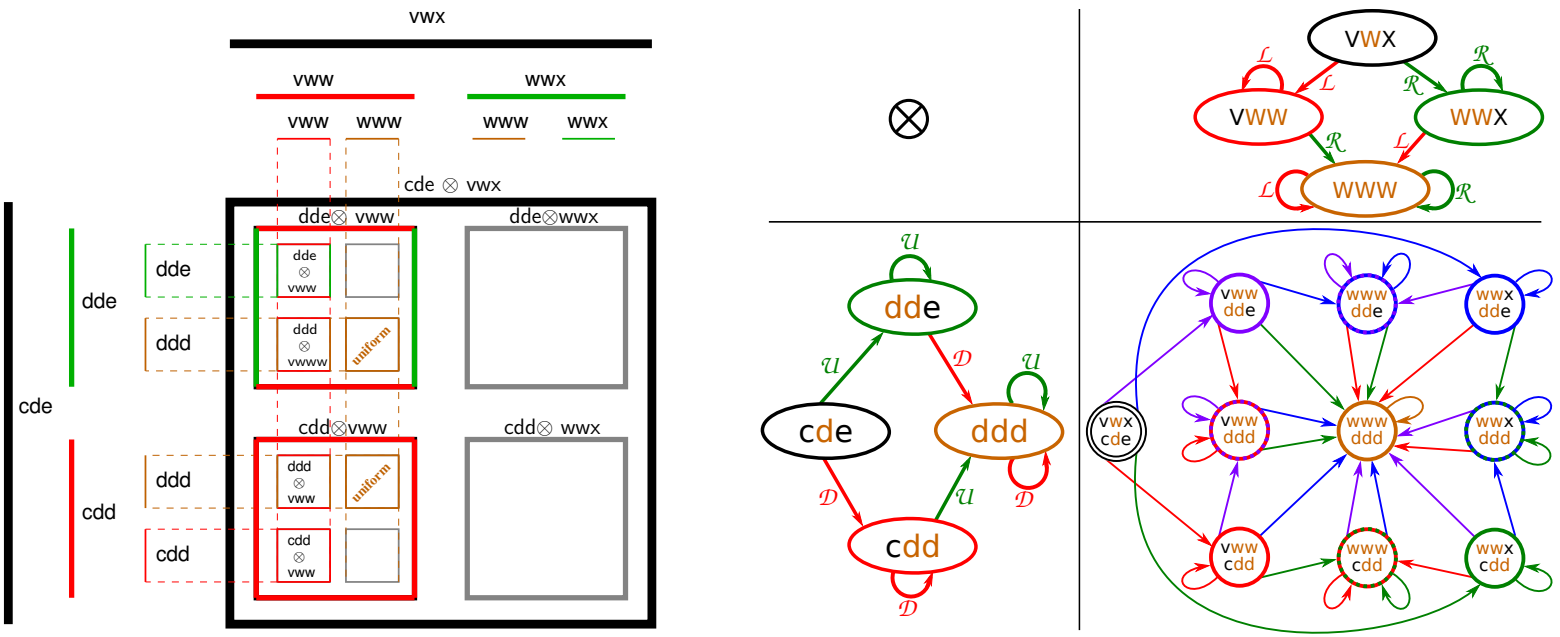


All states are transitional, but 3:

- one uniform state,
- two stationary states defining the beginning and the end of the curve

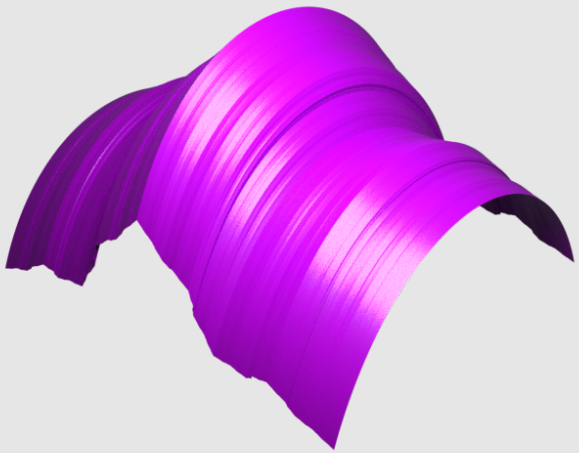
Generalization to surfaces

Simply compute the automaton tensor product

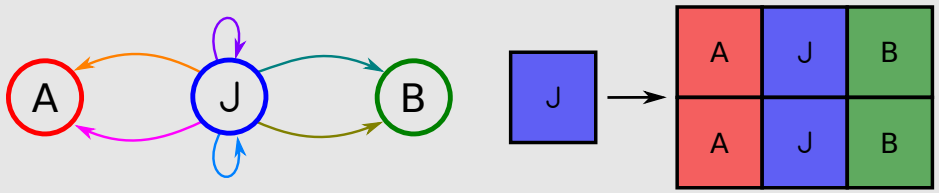


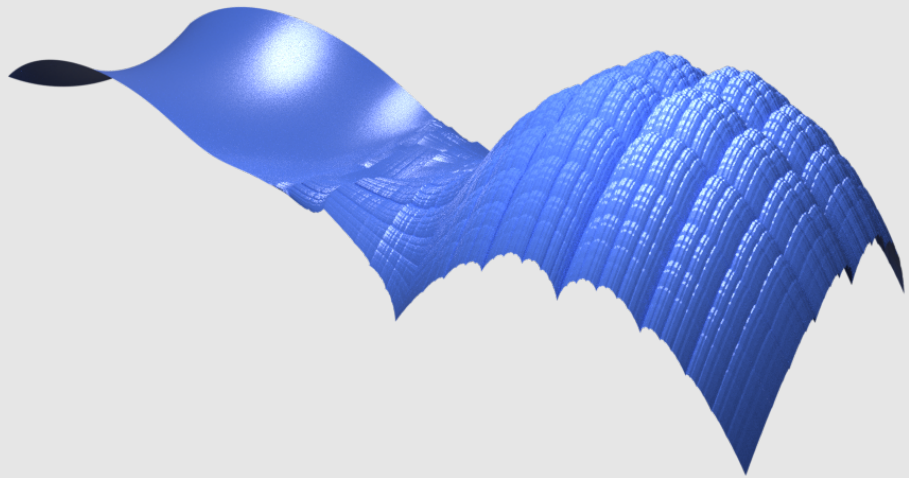
Applications

Fractal ⊗ NURBS



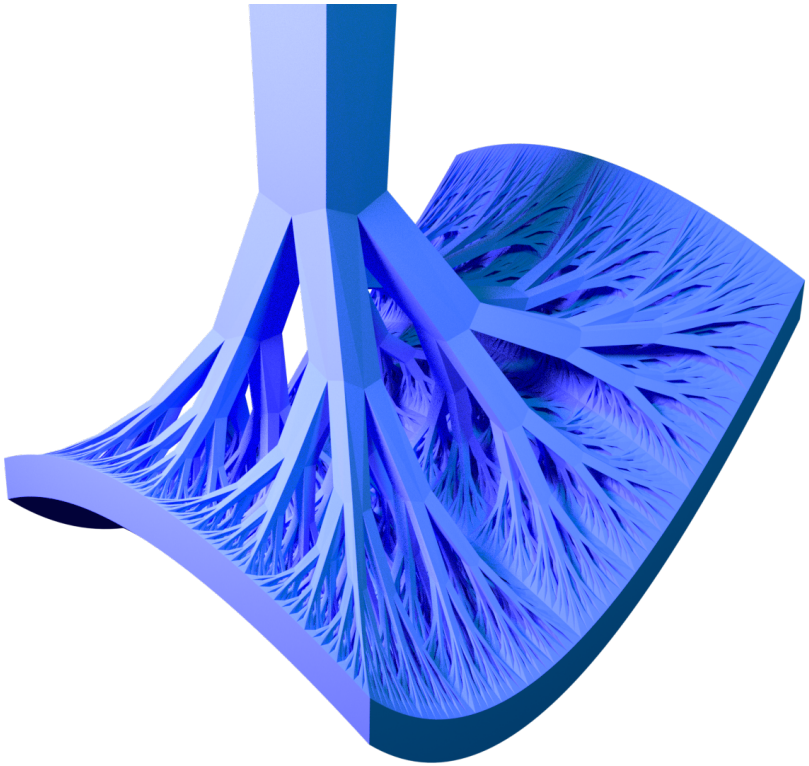
Junction between surfaces

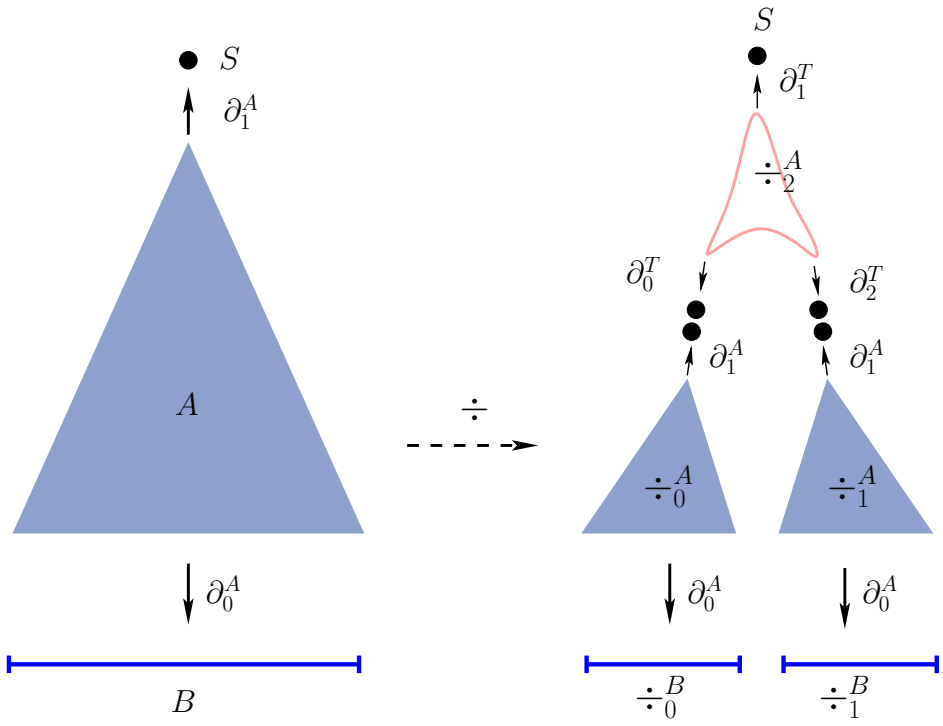




Applications

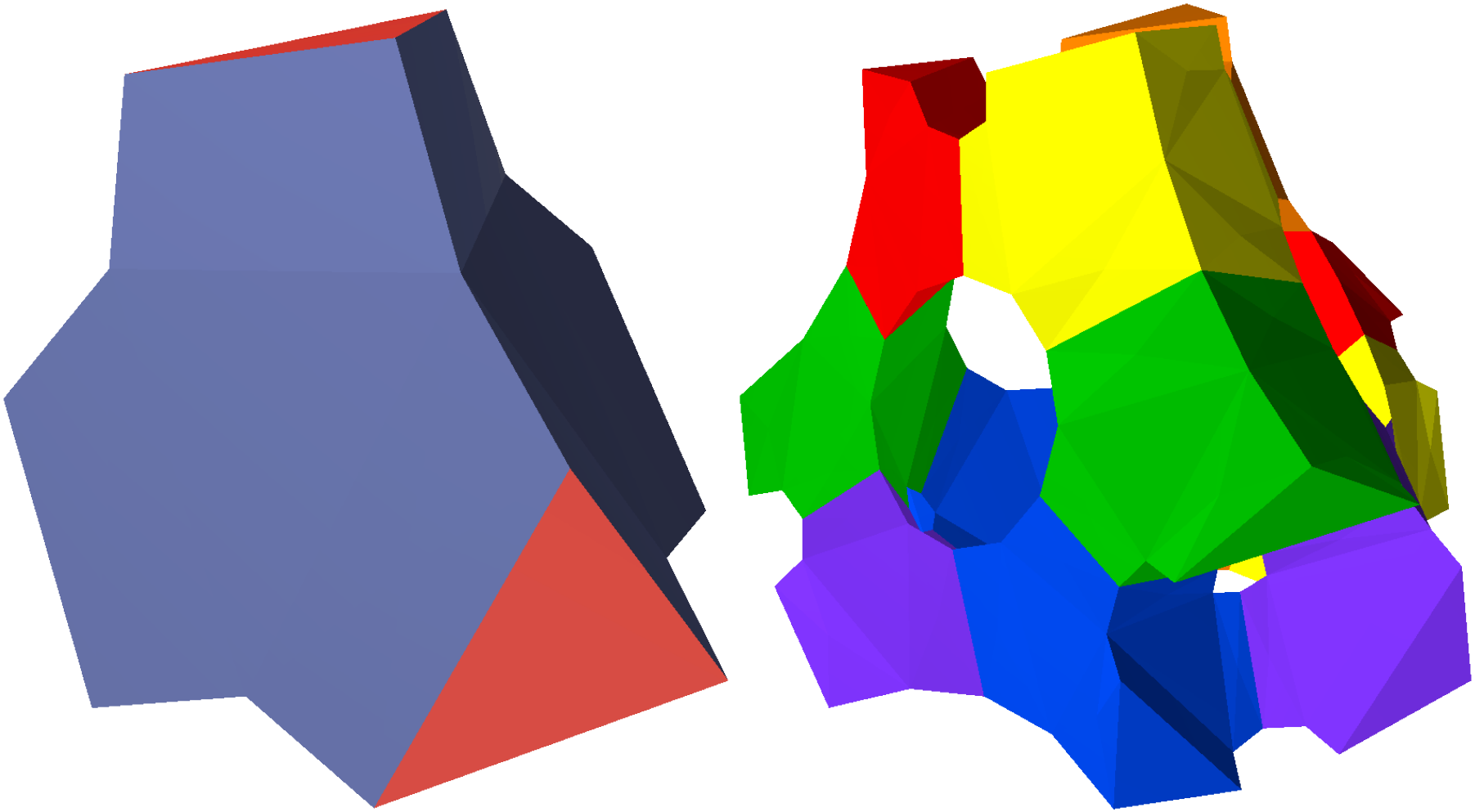
Building a tree supporting a given NURBS





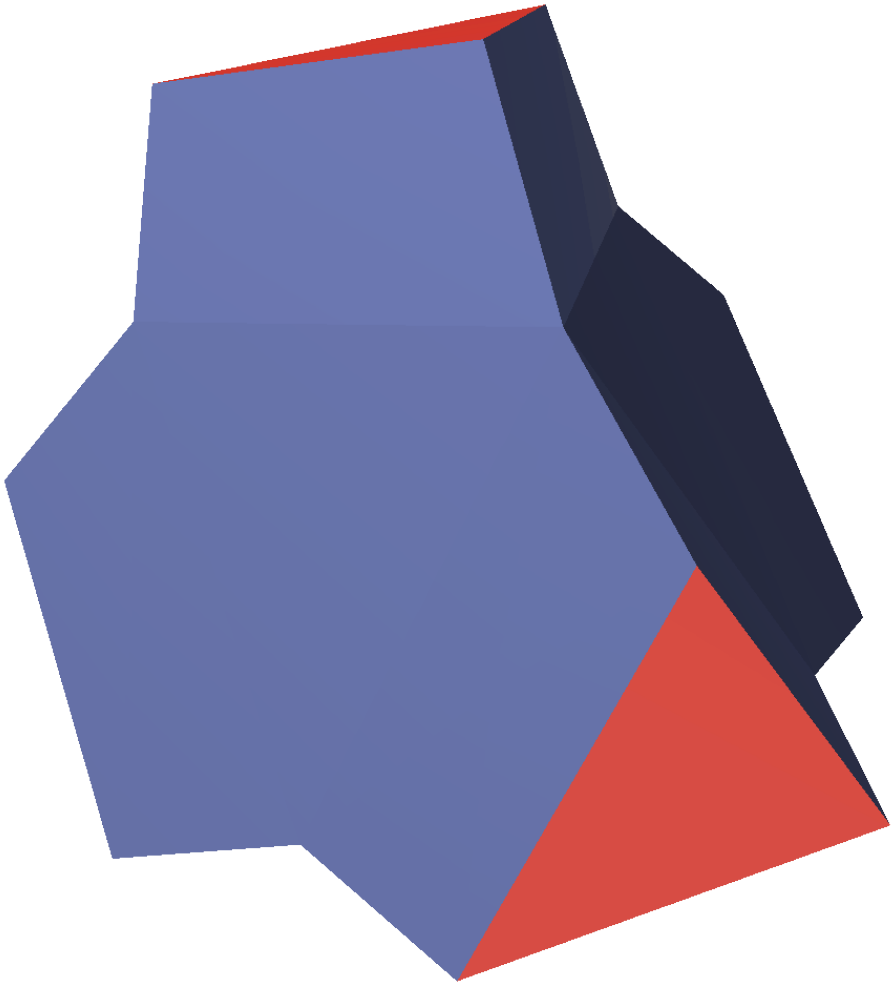
Applications

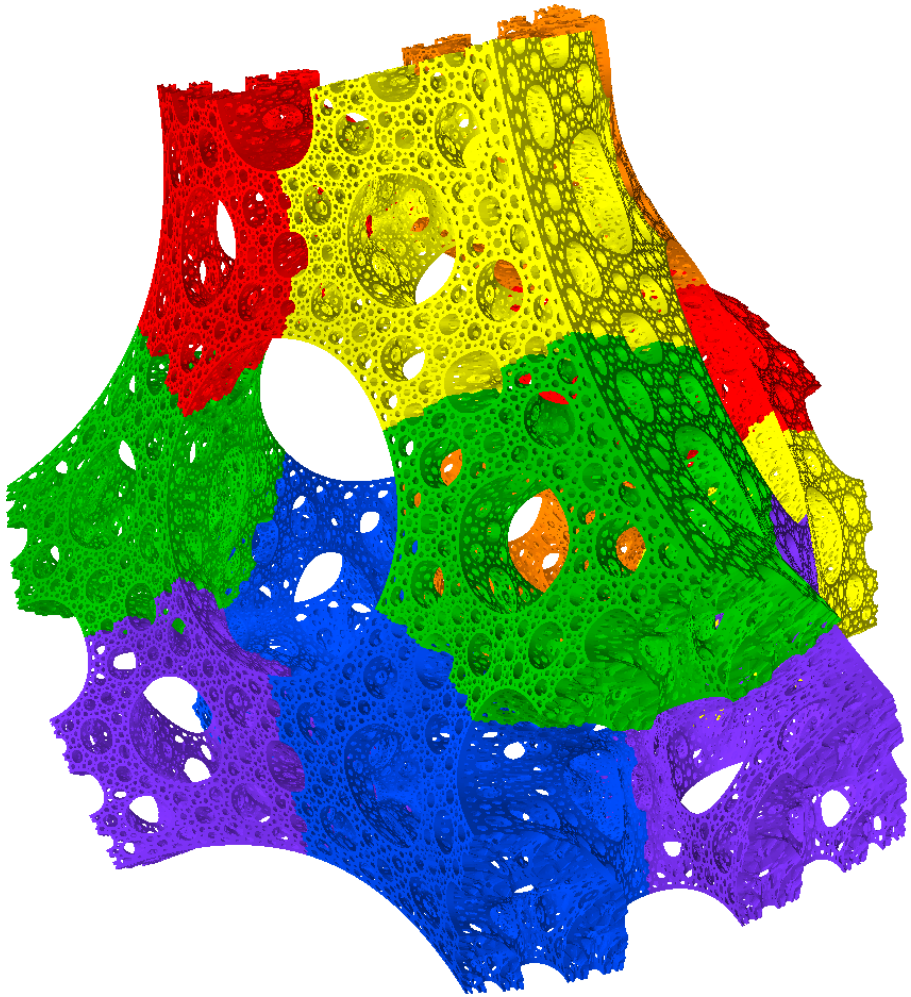
Lacunar structure with NURBS edges



Applications

Lacunar structure with NURBS edges



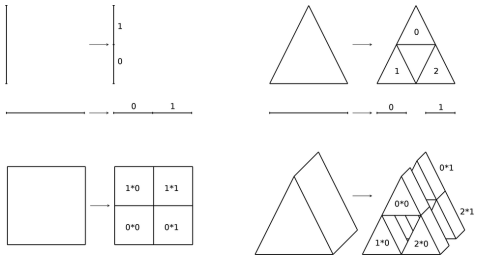


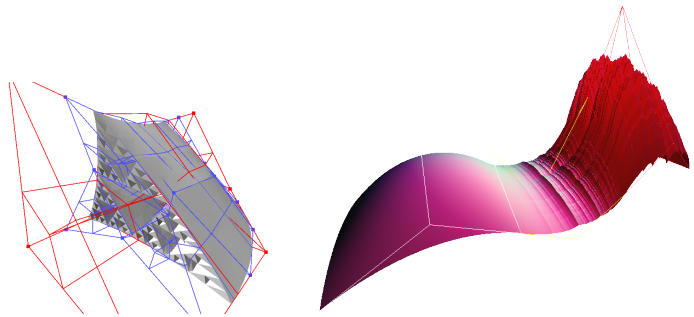
Conclusion

Conclusion

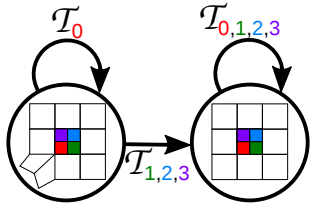
- Common formalism for fractals, subdivision surfaces, NURBS.
- Interaction, fusion between geometry representations

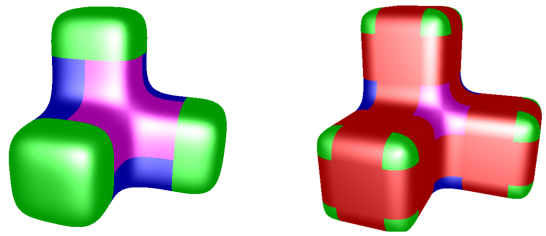
- "automaton tensor product", surfaces junction (Podkorytov 2013)





- Subdivision surfaces (Morlet 2018)





- Rational surfaces = control points with homogeneous coordinates

Future works

Future works and works in progress

- NURBS on irregular grids of control points
- Geometry / topology optimization