UNIVERSITÉ LUMIÈRE LYON 2
UNIVERSITÉ DE LYON

# Subdivision de maillages

C. Le Bihan Gautier

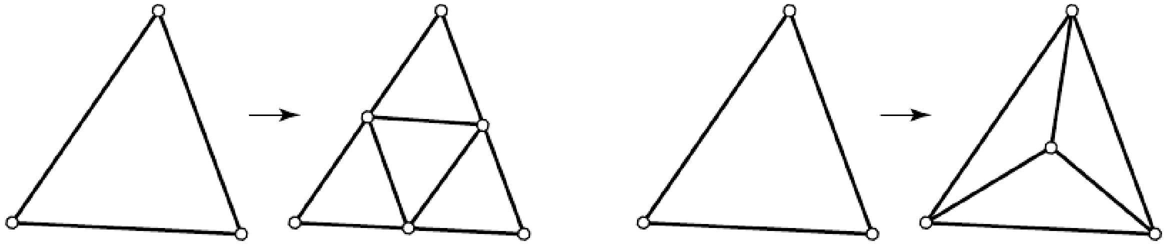
- Ce cours est une **compilation** :
  - Cours de Loïc Barthe, Modélisation géométrique (IRIT-UPS Toulouse; Equipe Vortex)
  - Cours S. Lanquetin, Université de Bourgogne
  - Cours G. Gesquière
  - Cours Ching-Kuang Shene

# Simplifier ou subdiviser ?

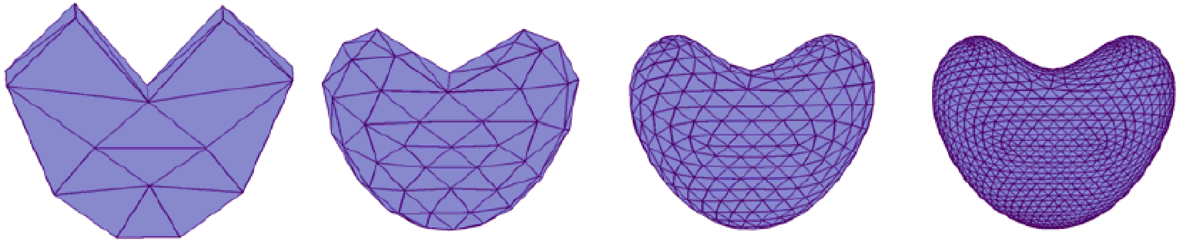
- But : Avoir accès au même objet, mais avec des représentations ayant un nombre différent de polygones. Il s'agit donc de créer une hiérarchie de maillages. Ces hiérarchies peuvent être :
  - "Bottom-top": On part du modèle détaillé (les feuilles de la hiérarchie) et on va jusqu'à la forme la plus simplifiée. **Les approches que l'on a vu jusqu'à maintenant, vont dans ce sens.**
  - **"Top-down": On part de la version simplifiée (la racine) et on ajoute progressivement des détails jusqu'à la représentation la plus fine. Des approches par surfaces de subdivision ou d'ondelettes suivent cette approche.**

# Raffinage/ Subdivision

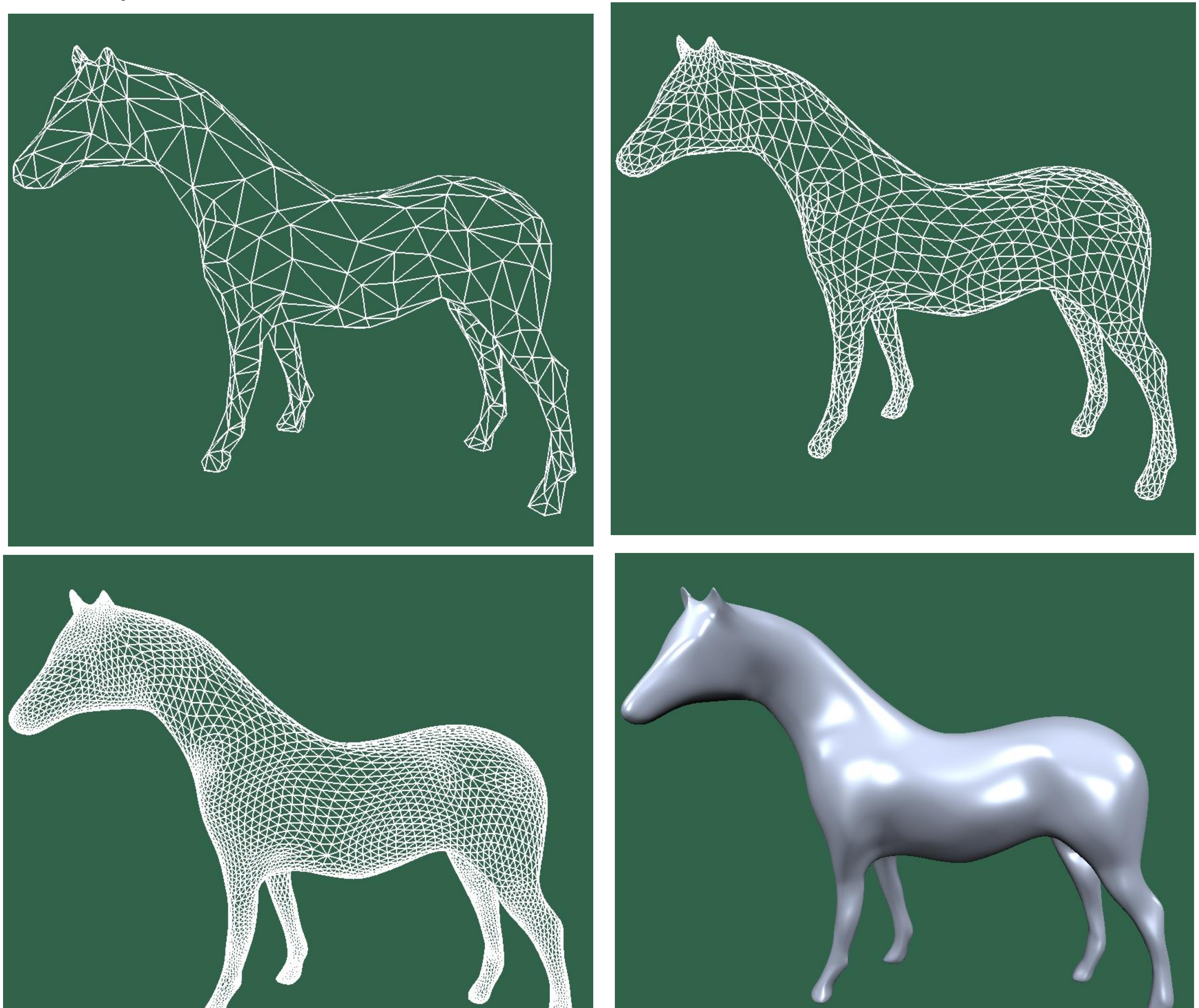
- Raffinage par découpage de triangles



- Subdivision (règle permettant de positionner les points d'un objet suite à un raffinement)

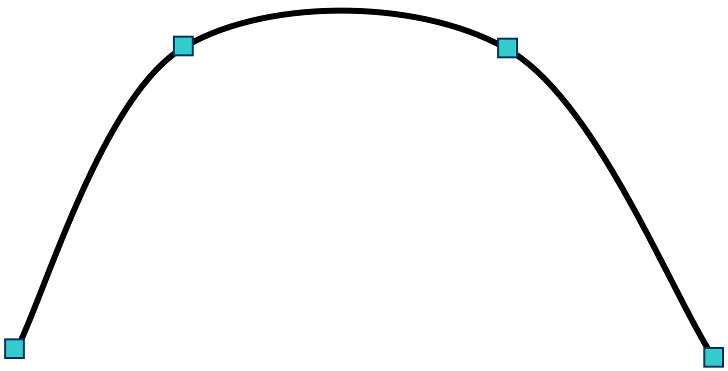


# Exemple de subdivision



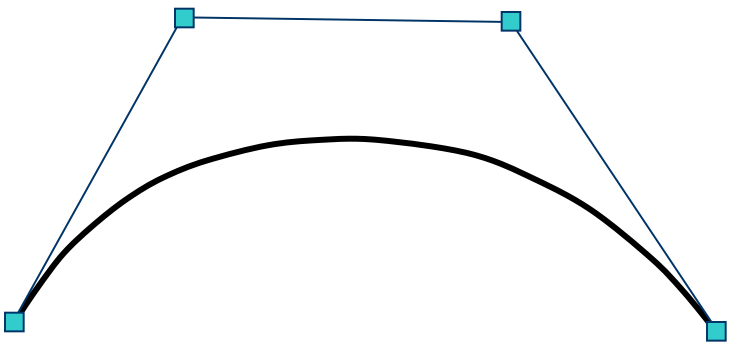
# Rappel : Interpolation / Approximation

**Interpolation**



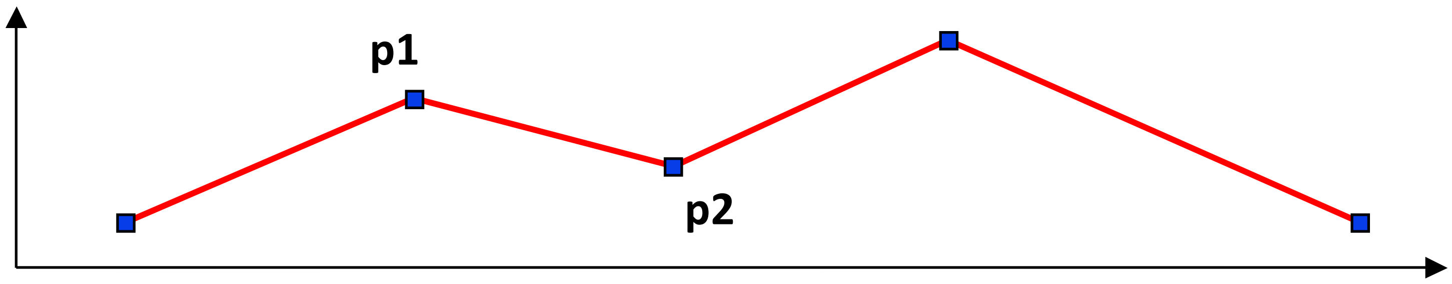
La courbe passe par
les points de contrôle

**Approximation**



La courbe est attirée par
les points de contrôle

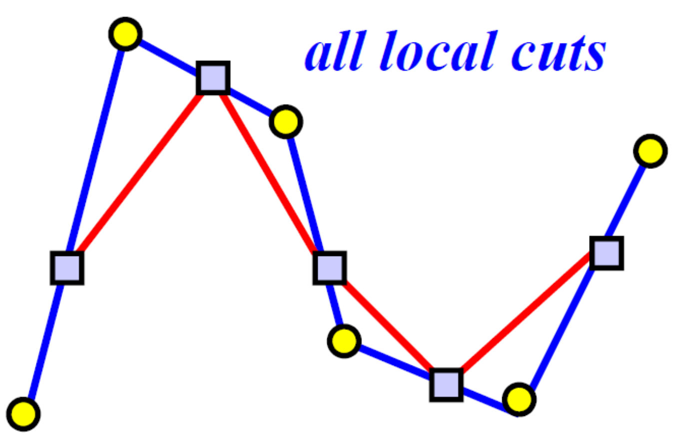
# Rappel : Polynômes linéaires par morceaux

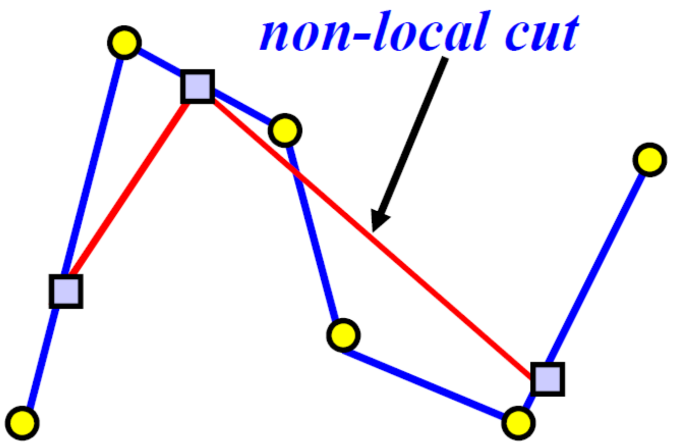



$$p(u) = u p_1 + (1 - u) p_2$$

# Un premier exemple : Curve Corner Cutting

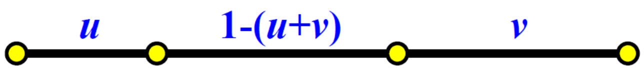
- Take two points on different edges of a polygon and join them with a line segment. Then, use this line segment to replace all vertices and edges in between. This is corner cutting!
- Corner cutting can be local or non-local.
- A cut is **local** if it removes exactly one vertex and adds two new ones. Otherwise, it is **non-local**.



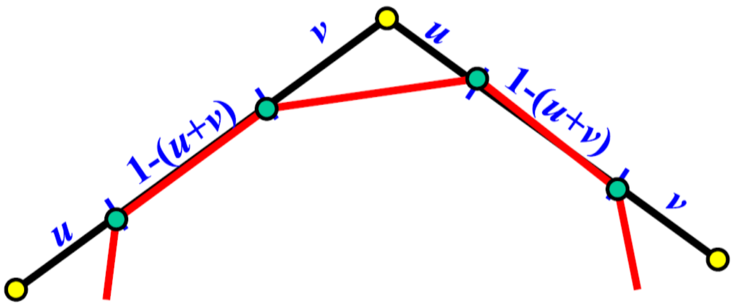



# Simple Corner Cutting: 1/5

- On each edge, choose two numbers u ≥ 0 and v ≥ 0 and u+v ≤ 1, and divide the edge in the ratio of u: 1- (u+v): v.




- Here is how to cut a corner.



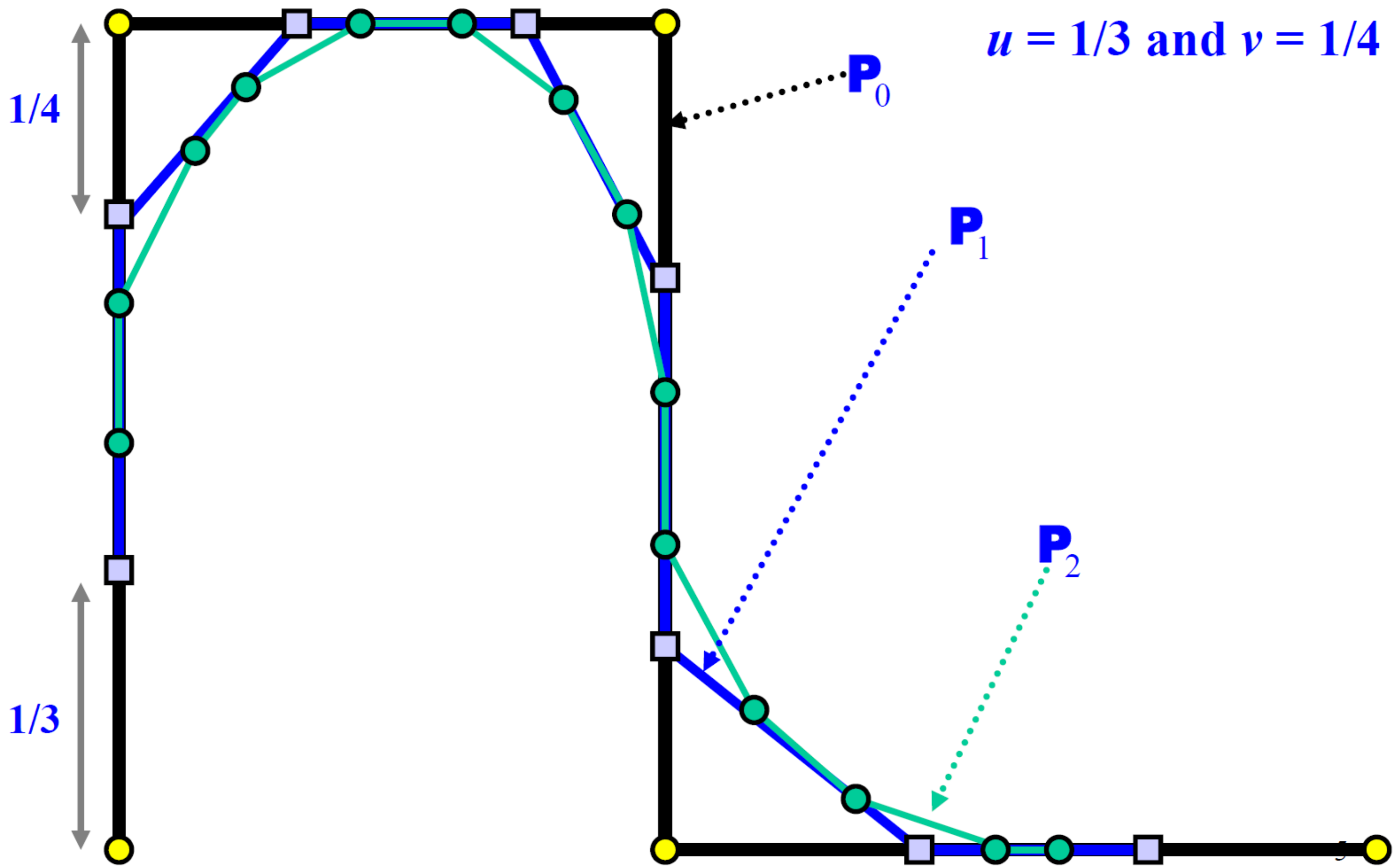
# Simple Corner Cutting: 2/5

- Suppose we have a polyline $P_0$. Divide its edges with the above scheme, yielding a new polyline $P_1$.
- Dividing $P_1$ yields $P_2$, …., and so on. What is

$$\mathbf{P}_{\infty} = \underset{i \to \infty}{\text{Limit}}\, \mathbf{P}_i$$

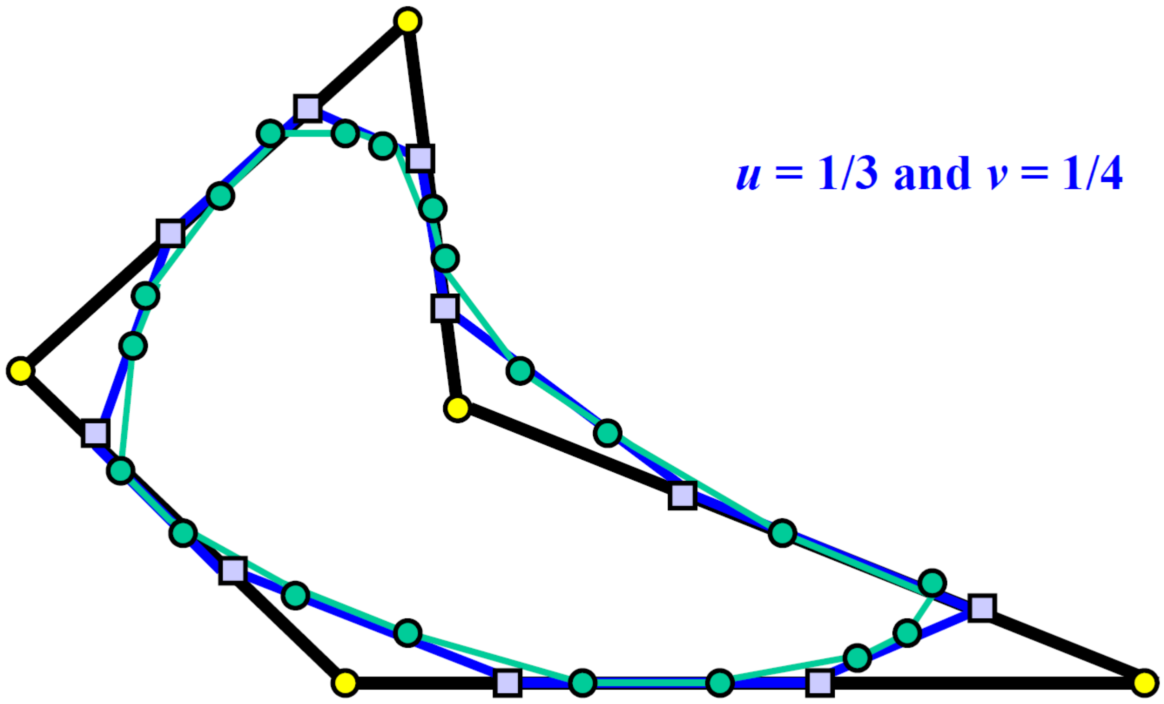
- The u's and v's do not have to be the same for edge. Moreover, the u's and v's used to divide Pi do not have to be equal to those u's and v's used to divide Pi+1.

# Simple Corner Cutting: 3/5

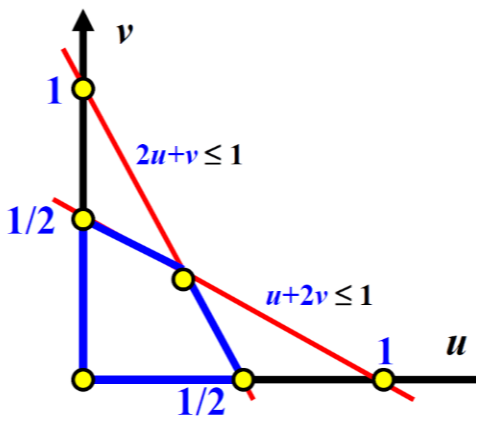


# Simple Corner Cutting: 4/5

- For a polygon, one more leg from the last point to the first must also be divided accordingly.



# Simple Corner Cutting: 5/5



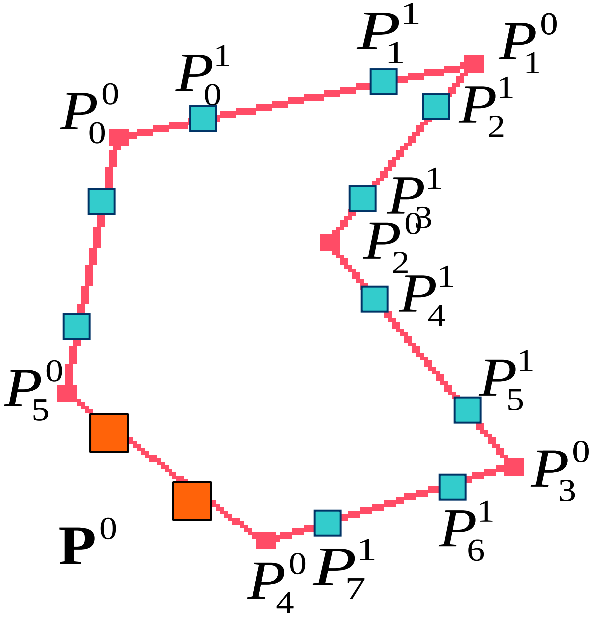

**Chaikin used $u = v = 1/4$**

- The following result was proved by Gregory and Qu, de Boor, and Paluszny, Prautzsch and Schäfer.
- If all u's and v's lies in the interior of the area bounded by $u \ge 0$, $v \ge 0$, $u+2v \le 1$ and $2u+v \le 1$, then $P^\infty$ is a $C^1$ curve.
- This procedure was studied by Chaikin in 1974, and was later proved that the limit curve is a B-spline curve of degree 2.

# Exemple: Algorithme de Chaikin

$$P_{2i}^{k+1} = \frac{3}{4} P_i^k + \frac{1}{4} P_{i+1}^k$$

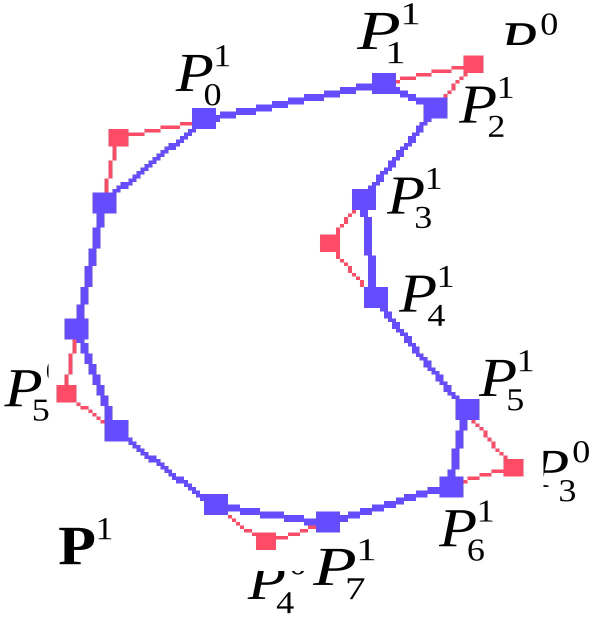
$$P_{2i+1}^{k+1} = \frac{1}{4} P_i^k + \frac{3}{4} P_{i+1}^k$$



# Exemple: Algorithme de Chaikin

$$P_{2i}^{k+1} = \frac{3}{4} P_i^k + \frac{1}{4} P_{i+1}^k$$

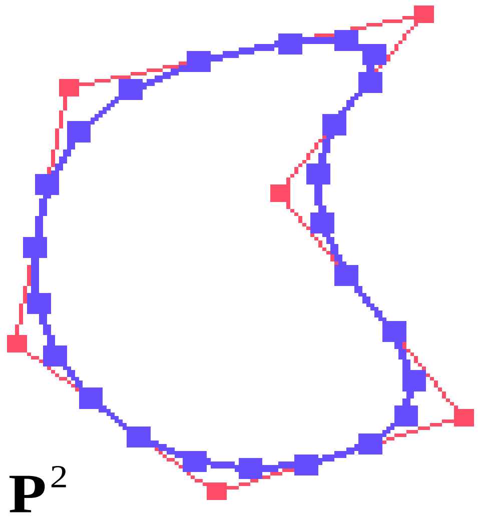
$$P_{2i+1}^{k+1} = \frac{1}{4} P_i^k + \frac{3}{4} P_{i+1}^k$$



# Exemple: Algorithme de Chaikin

$$P_{2i}^{k+1} = \frac{3}{4} P_i^k + \frac{1}{4} P_{i+1}^k$$

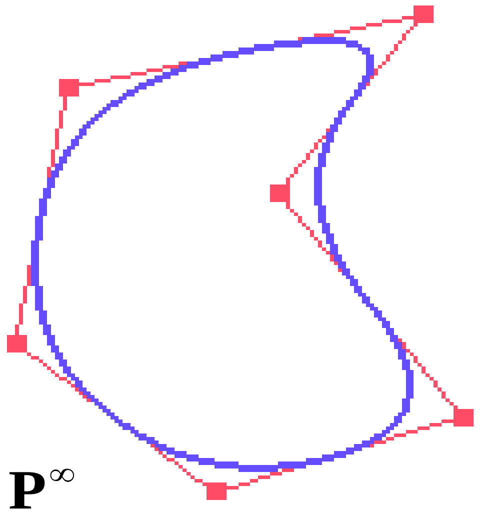
$$P_{2i+1}^{k+1} = \frac{1}{4} P_i^k + \frac{3}{4} P_{i+1}^k$$



# Exemple: Algorithme de Chaikin

$$P_{2i}^{k+1} = \frac{3}{4} P_i^k + \frac{1}{4} P_{i+1}^k$$

$$P_{2i+1}^{k+1} = \frac{1}{4} P_i^k + \frac{3}{4} P_{i+1}^k$$

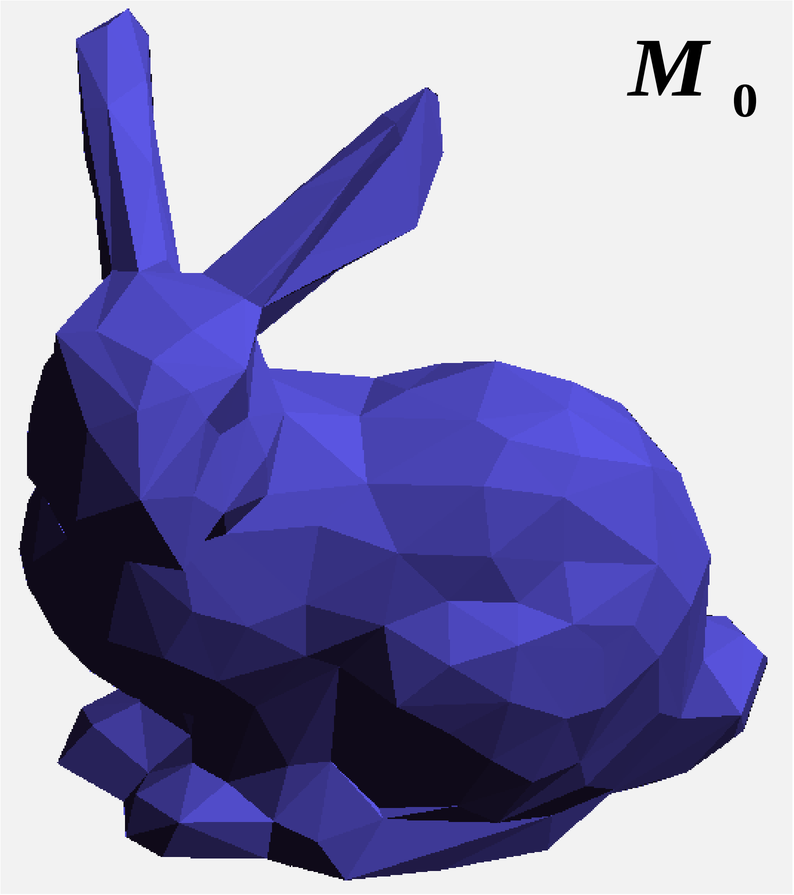


# Subdivision Surfaces : Avantages

- Il est facile de modéliser un grand nombre de surfaces de différents types.
- Il génère généralement des surfaces lisses.
- L'interaction avec les modèles est simple et intuitive.
- Il peut modéliser les caractéristiques nettes et semi-nettes des surfaces.
- Sa représentation est simple et compacte.

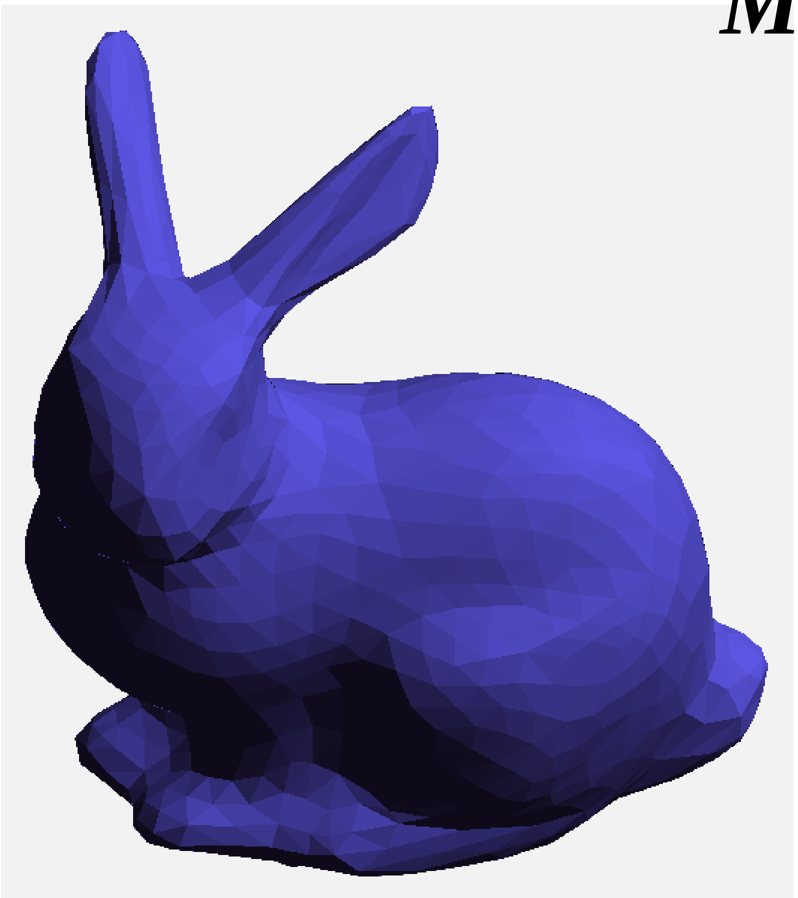
# Exemple : Surface de subdivision

- Maillage initial
- Règles de subdivision



# Exemple : Surface de subdivision

$M_1 = S \, . \, M_0$

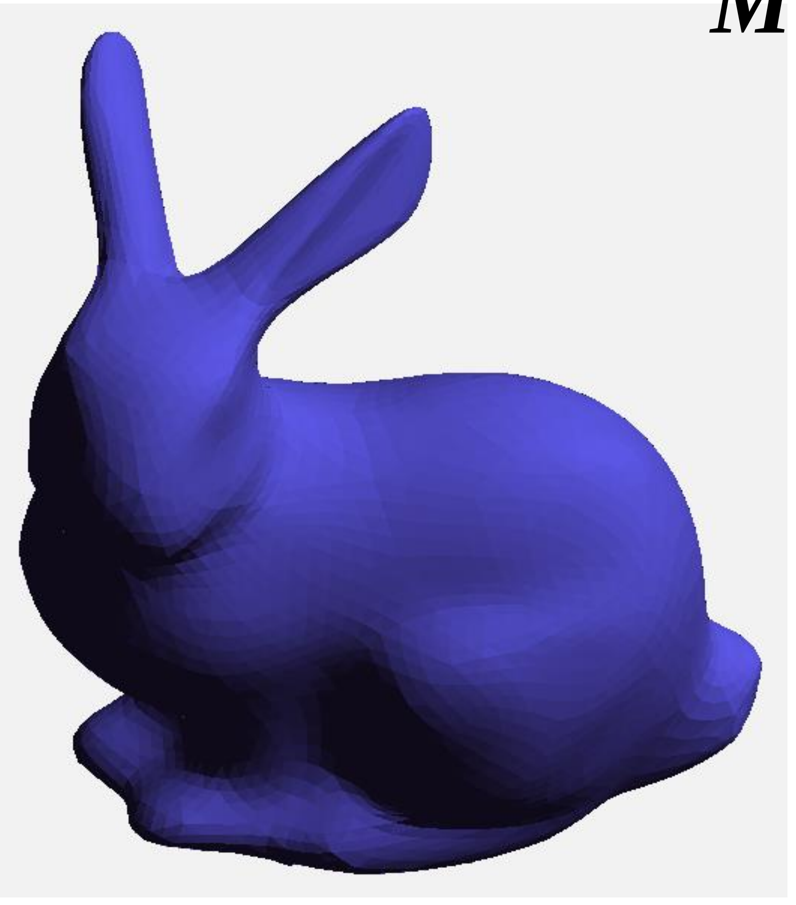


$\boldsymbol{M}_1$

# Exemple : Surface de subdivision

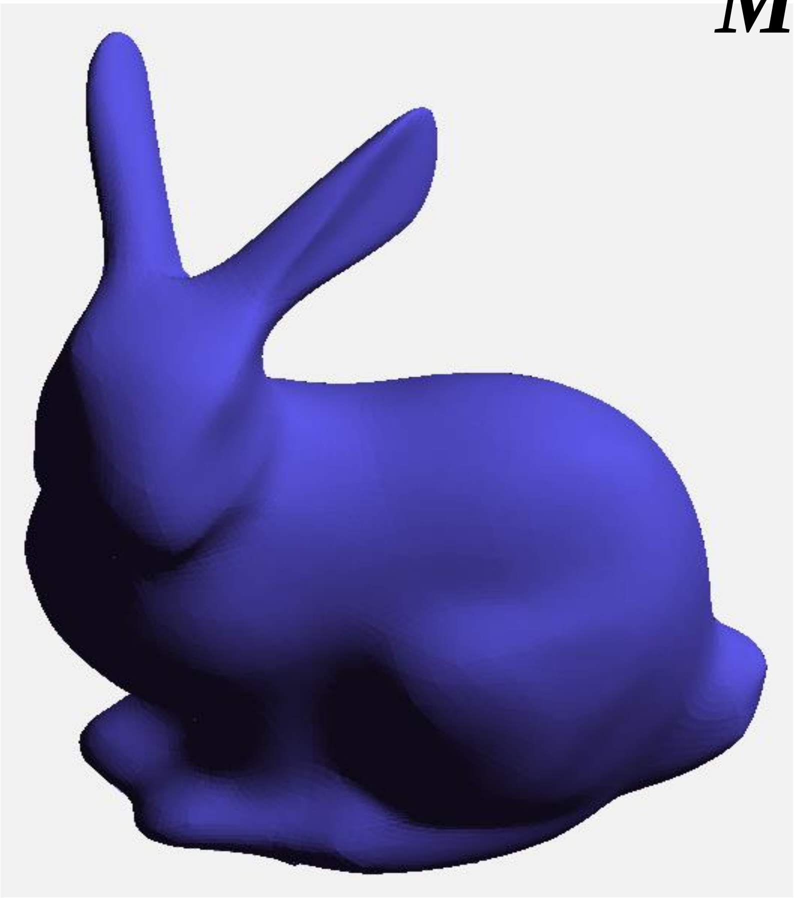
$M_2 = S \,.\, M_1$
$= S_2 \,.\, M_0$

$\boldsymbol{M}_{\mathbf{2}}$



# Exemple : Surface de subdivision

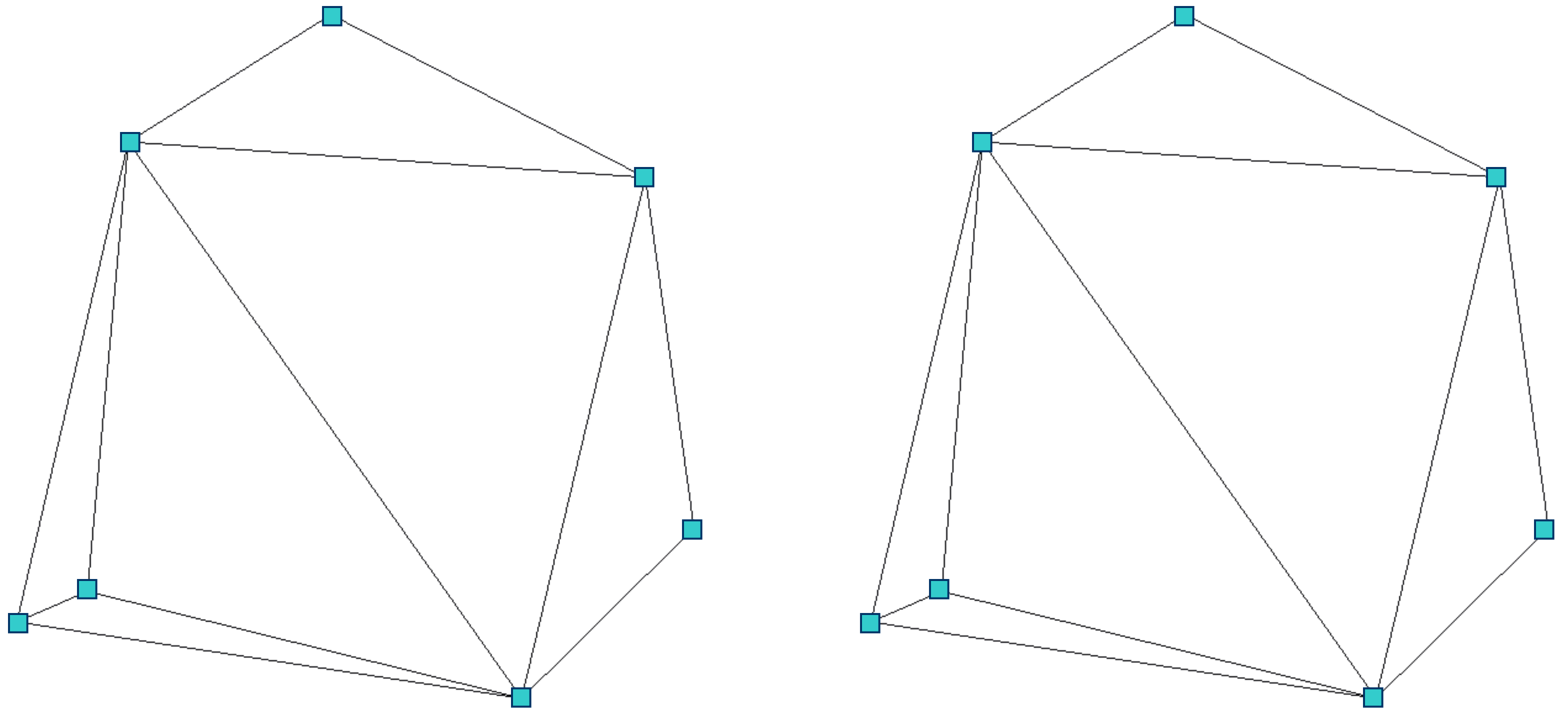
$M_{k+1} = S \,.\, M_k$



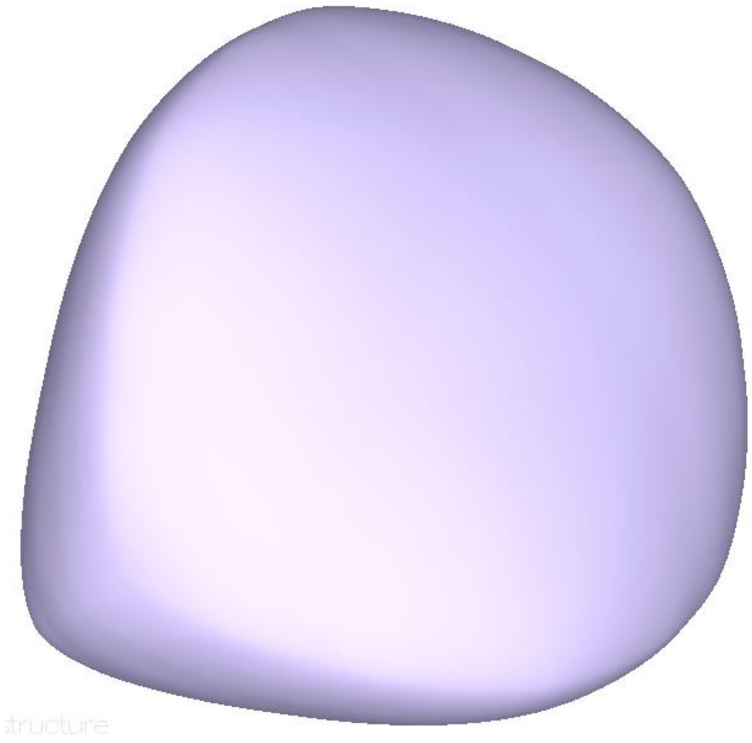
Surface lisse

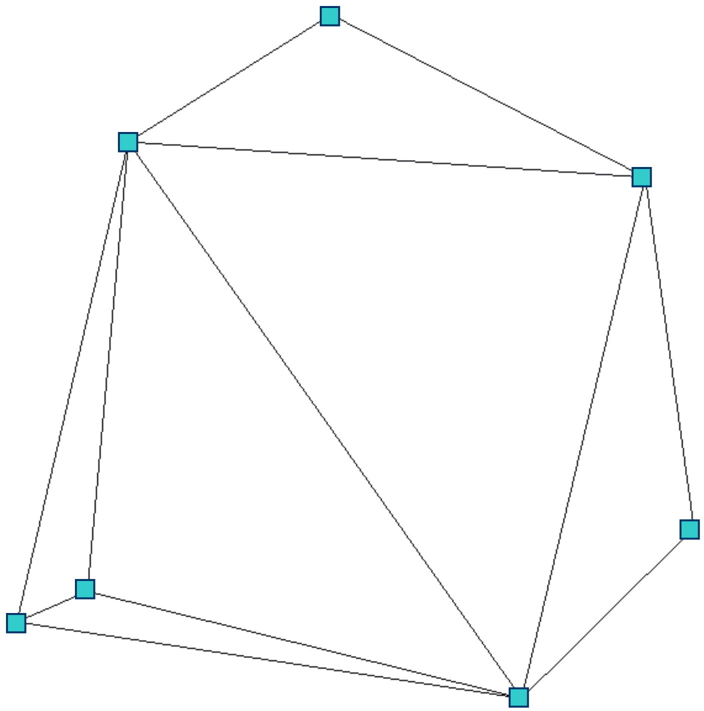
$\boldsymbol{M}_{3}$

# Interpolation / Approximation

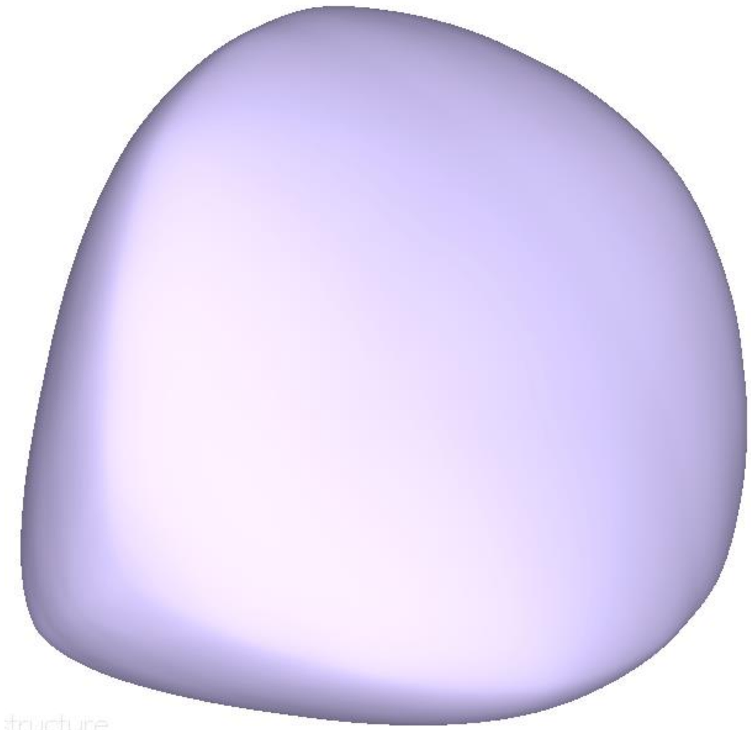


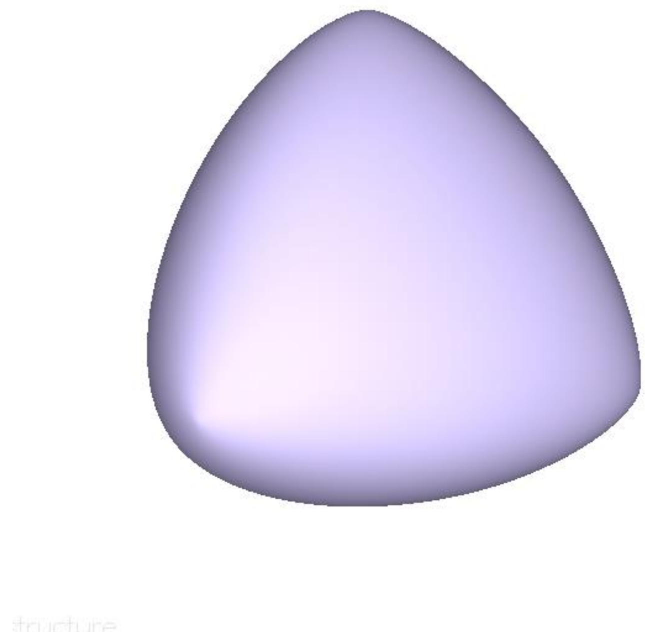
# Interpolation / Approximation





# Interpolation / Approximation



# Loop’s Algorithm: 1/4

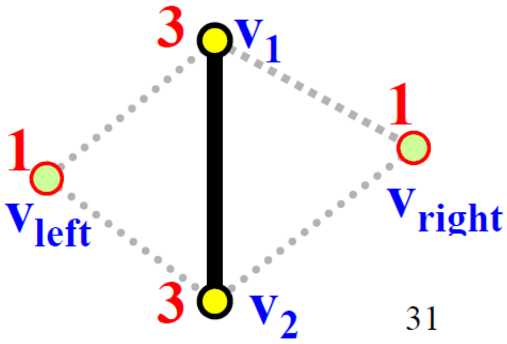
Loop’s (i.e., Charles Loop’s)

// Étape 1 : Calculer les nouveaux points pour chaque arête

Pour chaque arête e dans M

```
    v1, v2 = Sommets(e)
    v_nouveau = (3/8) * (v1 + v2) + (1/8) * (v_left(e) + v_right(e))
    AjouterSommet(M', v_nouveau)
```

$$\mathbf{e} = \frac{3}{8}(\mathbf{v}_1 + \mathbf{v}_2) + \frac{1}{8}(\mathbf{v}_{\text{left}} + \mathbf{v}_{\text{right}})$$



# Loop’s Algorithm: 2/4

// Étape 2 : Mettre à jour les positions des sommets existants

Pour chaque sommet v dans M

n = Valence(v)

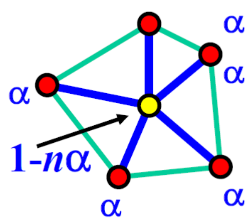
alpha = CalculerAlpha(n)

v_nouveau = (1 - n*alpha) * v + alpha * SommeVoisins(v)

AjouterSommet(M', v_nouveau)

- **For each vertex v, its new vertex point v’ is computed below, where $\mathbf{v}_1$, $\mathbf{v}_2$, …, $\mathbf{v}_n$ are adjacent vertices**

$$\mathbf{v}' = (1 - n\alpha)\mathbf{v} + \alpha \sum_{j=1}^{n} \mathbf{v}_j$$



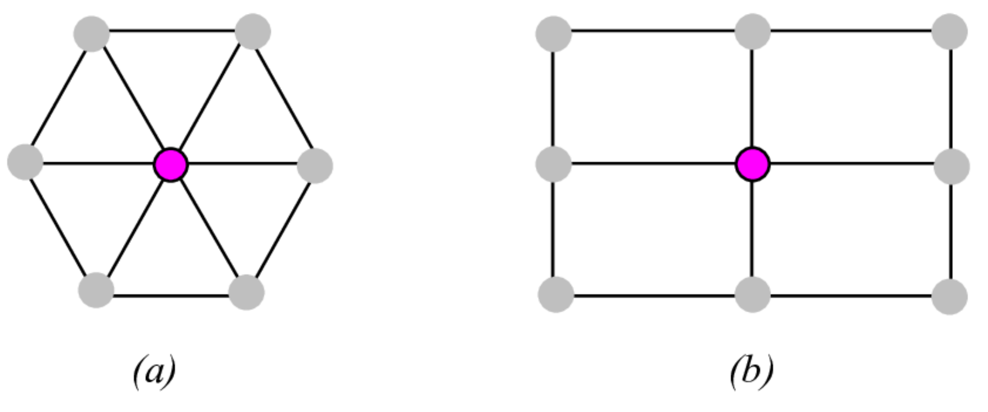

**where α is**

$$\alpha = \begin{cases} \dfrac{3}{16} & n=3 \\ \dfrac{1}{n}\left[\dfrac{5}{8} - \left(\dfrac{3}{8} + \dfrac{1}{4}\cos\dfrac{2\pi}{n}\right)^2\right] & n>3 \end{cases}$$

# Loop’s Algorithm : Valence

Valence : correspond au nombre d'arêtes connectées à un sommet.

- Pour un sommet intérieur du maillage, la valence est égale au nombre de faces adjacentes à ce sommet
- Pour un sommet sur le bord du maillage, la valence est égale au nombre d'arêtes connectées à ce sommet



# Loop’s Algorithm: 3/4

// Étape 3 : Créer les nouvelles faces

Pour chaque face f dans M

    x1, x2, x3 = Sommets(f)

    e1, e2, e3 = Arêtes(f)

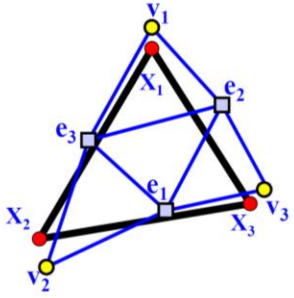
    x1x2, x2x3, x3x1 = NouveauxSommets(e1, e2, e3)

    AjouterFace(M’, x1, x1x2, x3x1)

    AjouterFace(M’, x2, x2x3, x1x2)

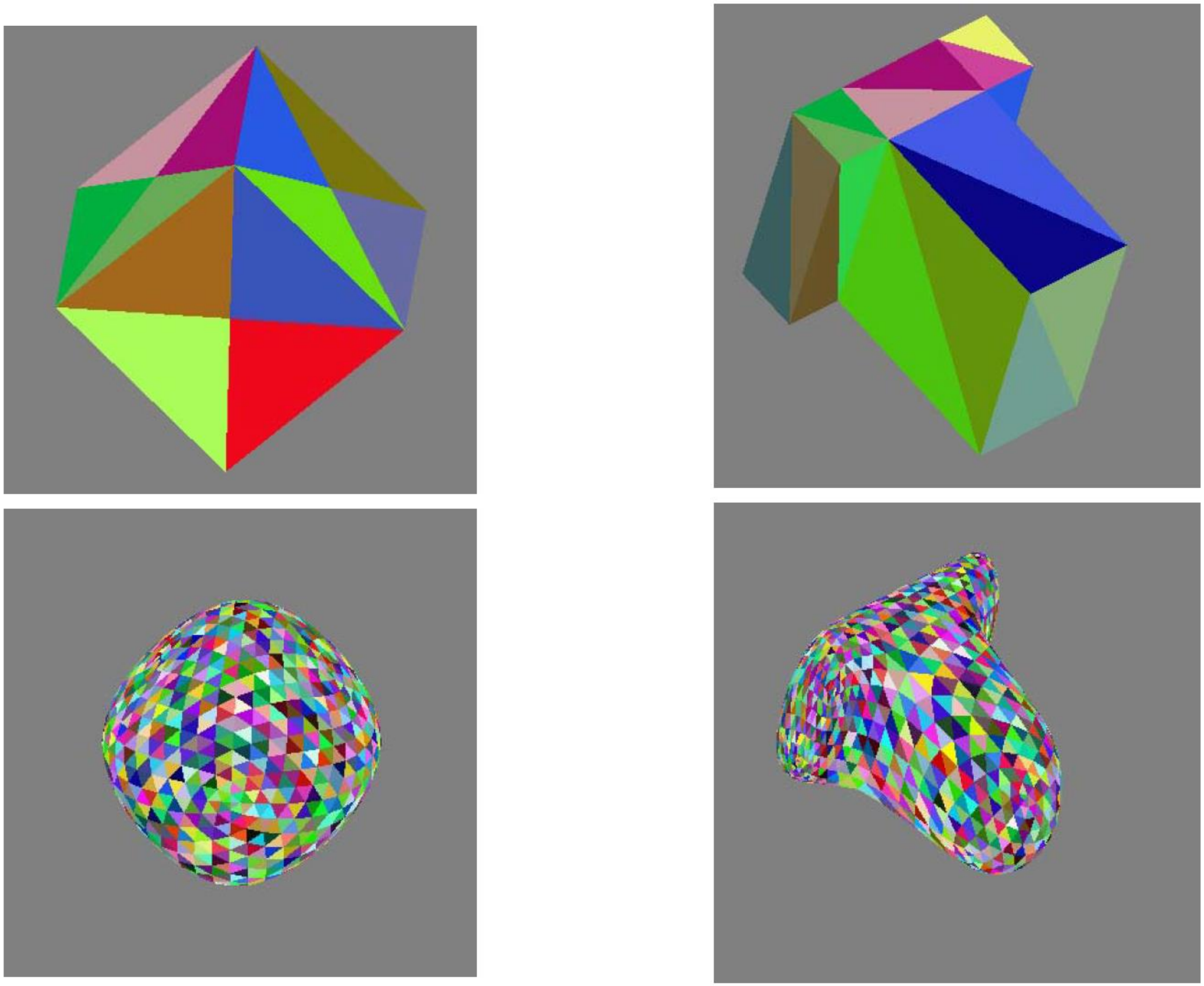
    AjouterFace(M’, x3, x3x1, x2x3)

    AjouterFace(M’, x1x2, x2x3, x3x1)




- **Let a triangle be defined by $X_1$, $X_2$ and $X_3$ and the corresponding new vertex points be $v_1$, $v_2$ and $v_3$.**
- **Let the edge points of edges $v_1v_2$, $v_2v_3$ and $v_3v_1$ be $e_3$, $e_1$ and $e_2$. The new triangles are $v_1e_2e_3$, $v_2e_3e_1$, $v_3e_1e_2$ and $e_1e_2e_3$. This is a 1-to-4 scheme.**
- **This algorithm was developed by Charles Loop in 1987.**

# Loop’s Algorithm 5/5

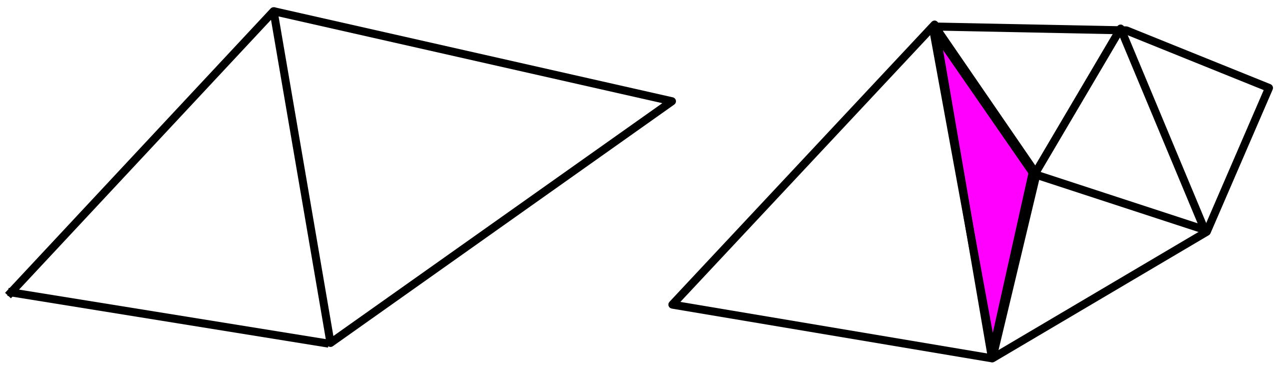


# Subdivision adaptative

Principe de subdivision adaptive ou non-uniforme

- Où subdiviser ?
- Comment subdiviser ?

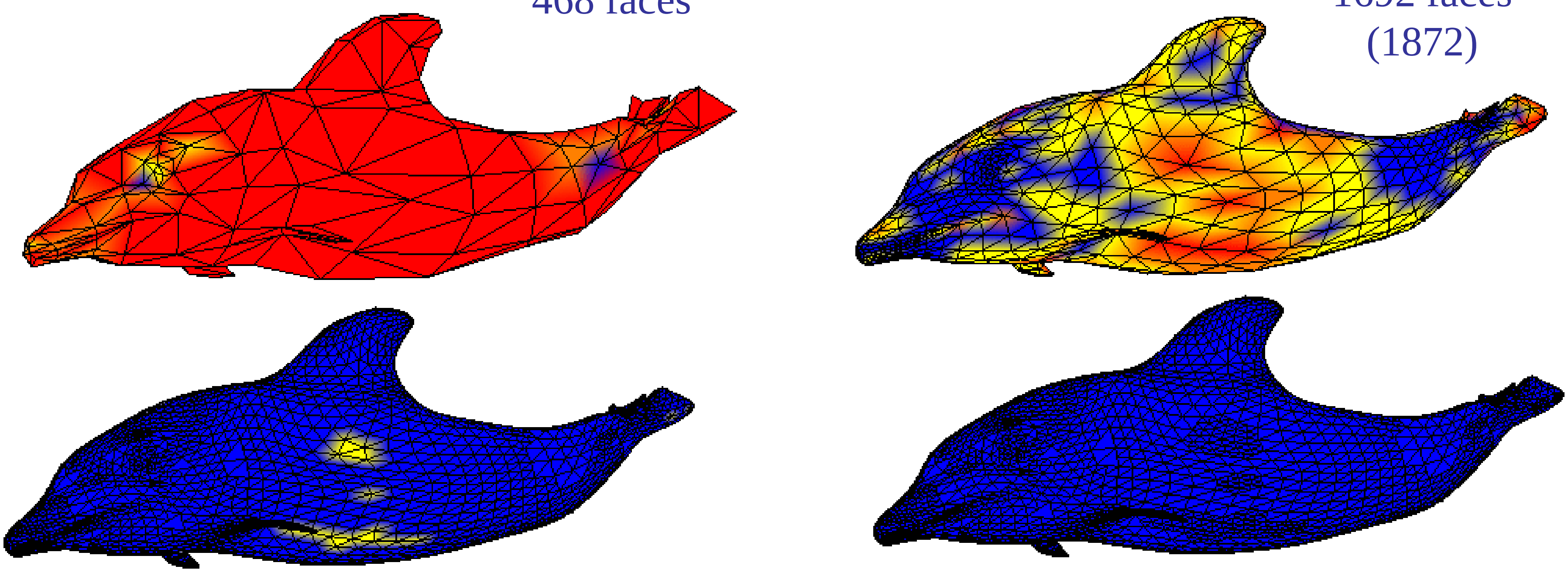
# Problème de la subdivision adaptative



- Éviter les trous
- Générer un "petit" nombre de faces
- Obtenir un maillage progressif

# Subdivision Adaptive

- Utiliser des règles pour subdiviser seulement certaines zones en fonction de critères
- Résultats




S. Lanquetin