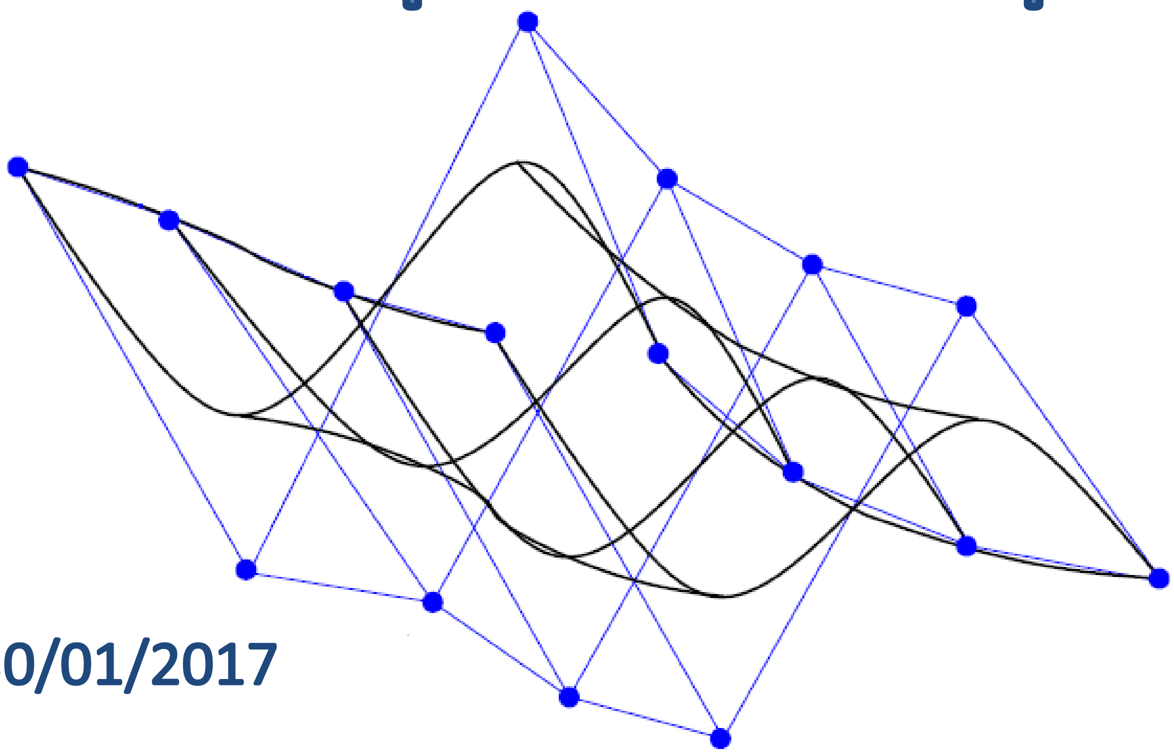
Modélisation et Programmation 3D

Surfaces paramétriques



Cours du 30/01/2017

Plan

- Introduction
- Surfaces balayées
- Carreaux surfaciques
- Visualisation OpenGL

Plan

- Introduction
- Surfaces balayées
- Carreaux surfaciques
- Visualisation OpenGL

Introduction

- **Courbes 3D :**
  - par rapport à un paramètre *u*

$$
\boxed{\begin{aligned} x &= x(u) \\ y &= y(u) \\ z &= z(u) \end{aligned}}
$$

Introduction

- **Courbes 3D :**
  - par rapport à un paramètre *u*

$$\begin{aligned} x &= x(u) \\ y &= y(u) \\ z &= z(u) \end{aligned}$$

- **Surfaces 3D :**
  - par rapport à **deux** paramètres *u et* v

$$\begin{aligned} x &= x(u, v) \\ y &= y(u, v) \\ z &= z(u, v) \end{aligned}$$

Introduction

- **Courbes 3D :**
  - par rapport à un paramètre *u*

$$\boxed{\begin{aligned} x &= x(u)\\ y &= y(u)\\ z &= z(u)\end{aligned}}$$

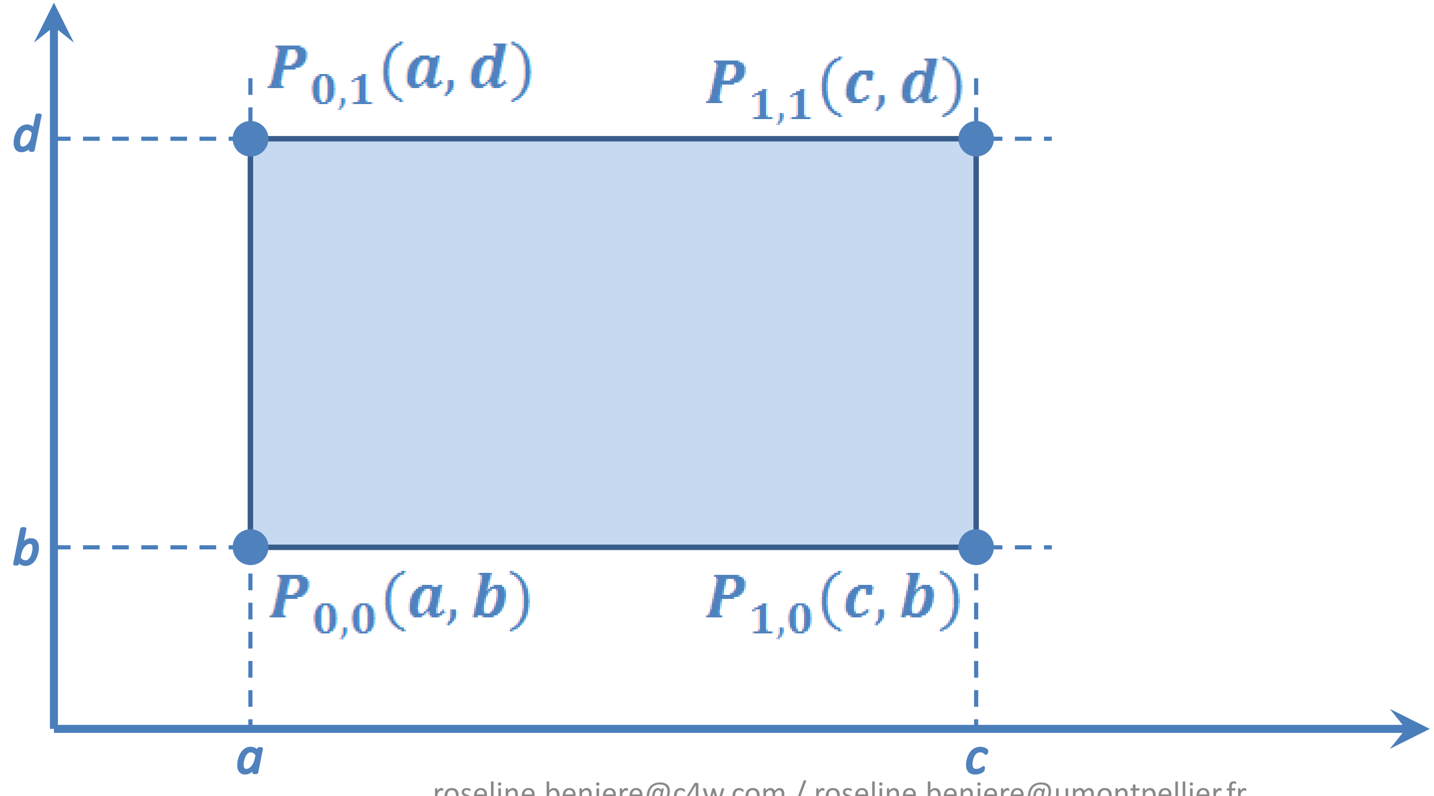
- **Surfaces 3D :**
  - par rapport à **deux** paramètres *u et* v

$$\boxed{\begin{aligned} x &= x(u,v)\\ y &= y(u,v)\\ z &= z(u,v)\end{aligned}}$$

➡ **Des propriétés étendues des courbes correspondantes**

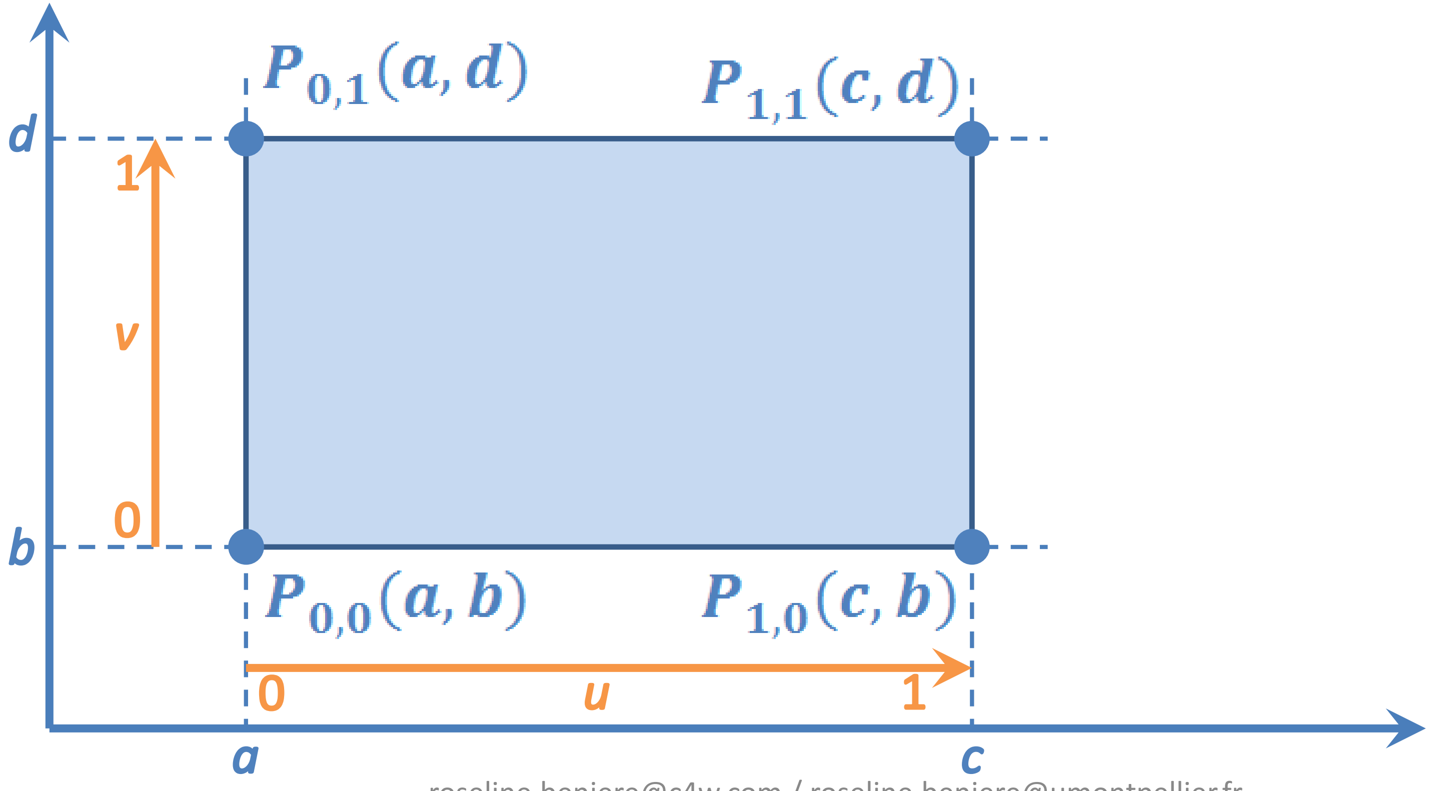
Introduction

- Exemple : carreau rectangulaire dans le plan XY
  - définit par 4 sommets : $P_{0,0}$, $P_{0,1}$, $P_{1,0}$, $P_{1,1}$.



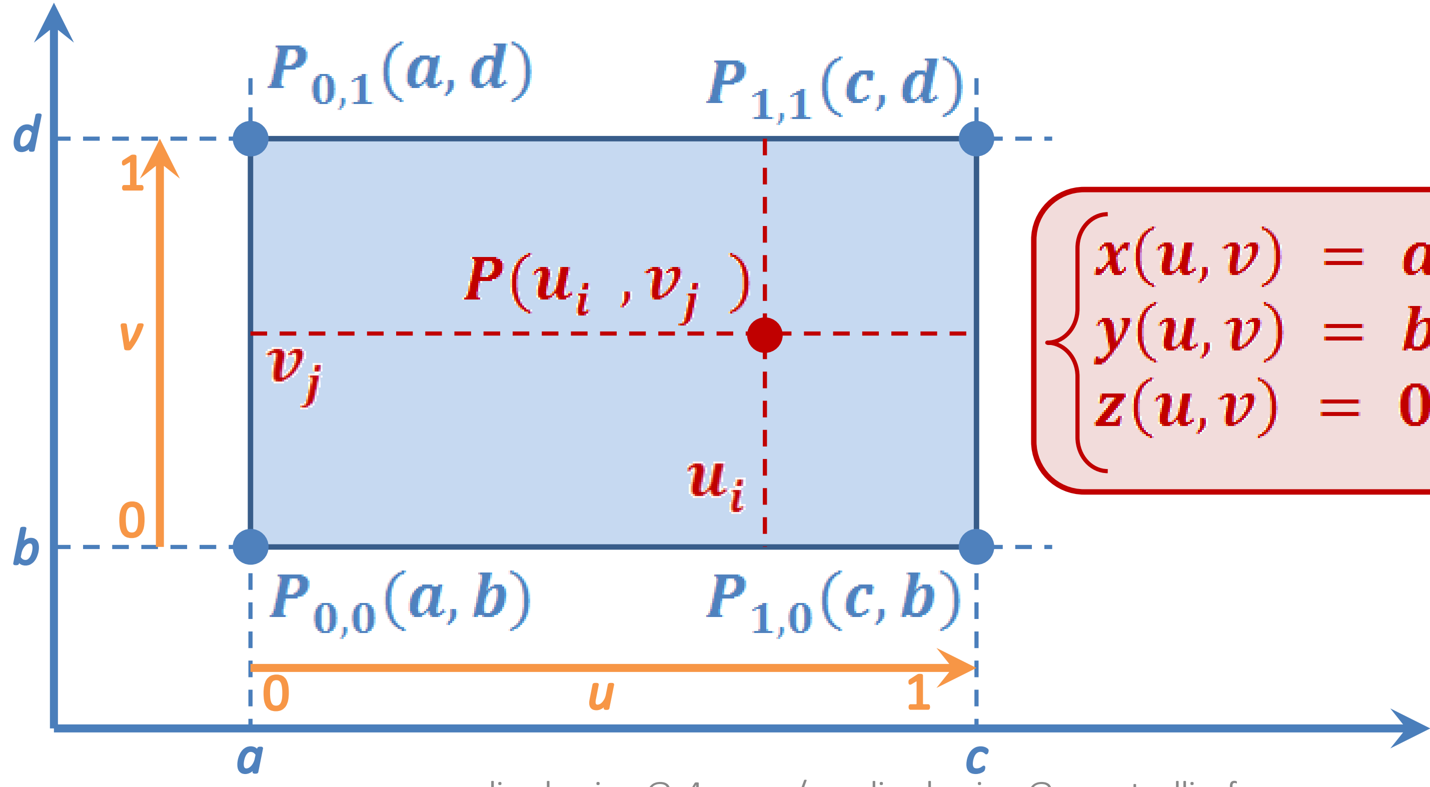
Introduction

- Exemple : carreau rectangulaire dans le plan XY
  - définit par 4 sommets : $P_{0,0}$, $P_{0,1}$, $P_{1,0}$, $P_{1,1}$.



Introduction

- Exemple : carreau rectangulaire dans le plan XY
  - définit par 4 sommets : $P_{0,0}$, $P_{0,1}$, $P_{1,0}$, $P_{1,1}$.

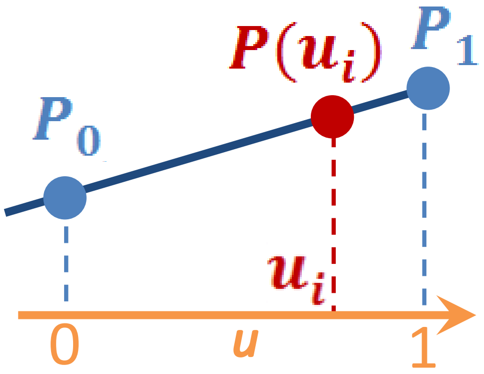


$$\begin{cases} x(u,v) = a + (c-a)u \\ y(u,v) = b + (d-b)v \\ z(u,v) = 0 \end{cases}$$

Introduction

- Courbe 3D ➡ cas de la droite

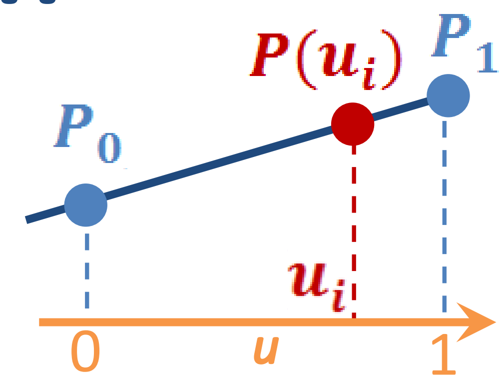
$$\boxed{P(u) = (1-u)P_0 + uP_1}$$



Introduction

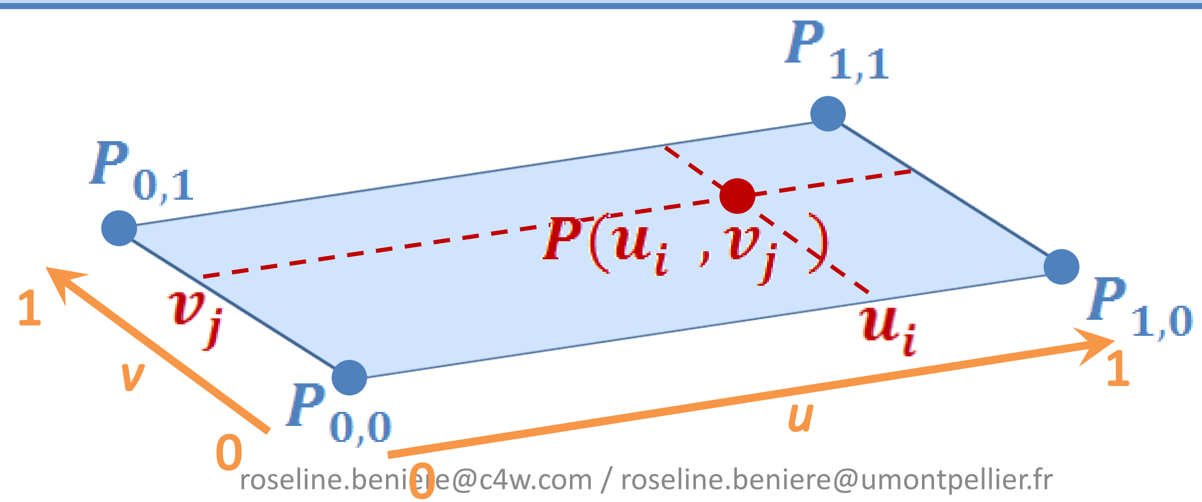
- Courbe 3D ➡ cas de la droite

$$P(u) = (1-u)P_0 + uP_1$$



- Surface 3D ➡ cas du plan

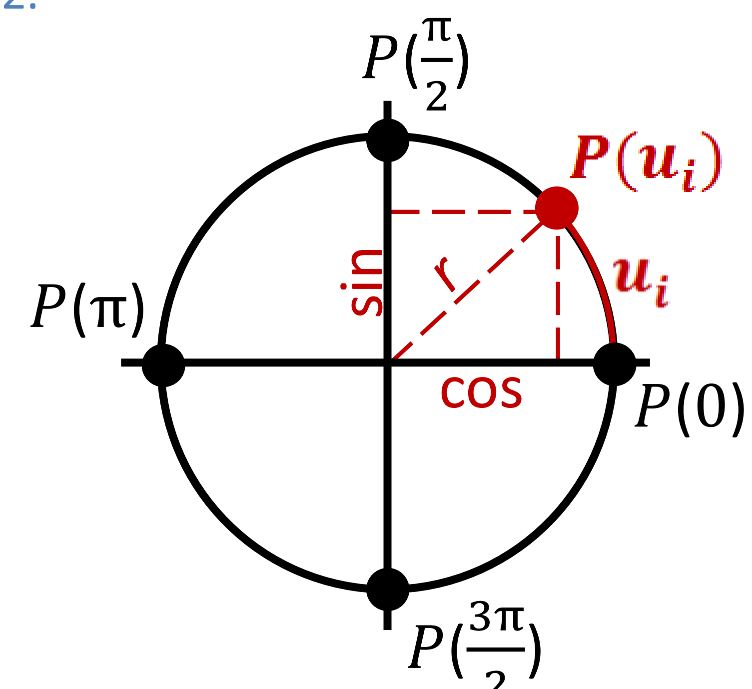
$$P(u,v) = P_{0,0} + (P_{1,0} - P_{0,0})u + (P_{0,1} - P_{0,0})v$$



Introduction

- **Rappel cercle 2D :**
  - ➢ centré à l'origine et définie par un rayon r et un paramètre *u* défini dans IR^2.

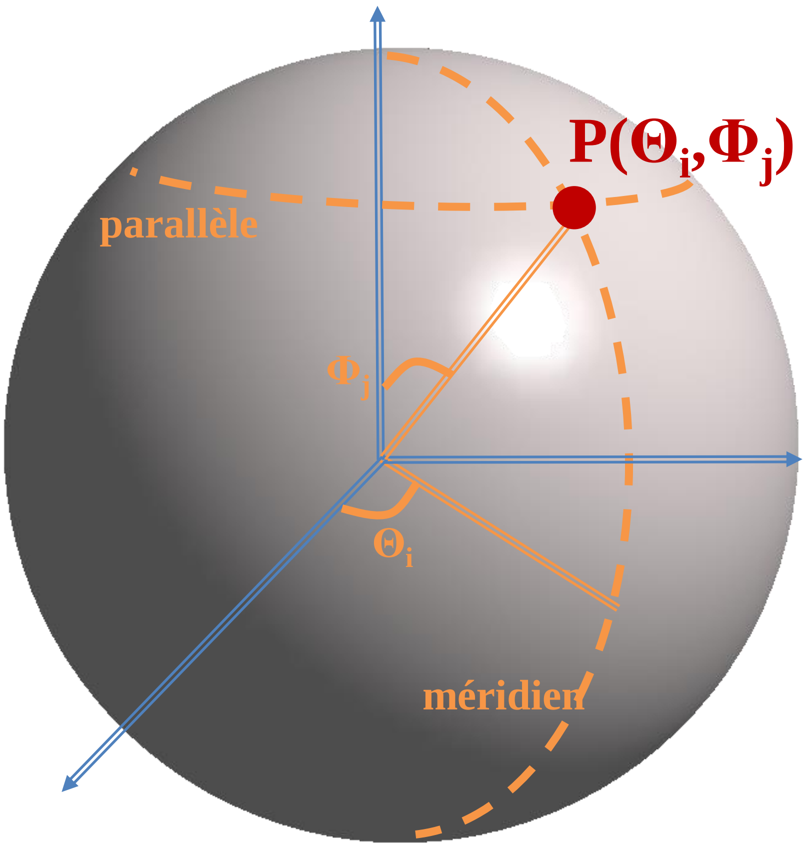
$$u \rightarrow P(u) = \begin{cases} x(u) = r.\cos(u) \\ y(u) = r.\sin(u) \end{cases}$$



Introduction

- **Surface 3D ➡ cas de la sphère**
  - ➢ centrer à l'origine
  - ➢ chaque point est définit par 2 angles

$$P(\phi, \theta) \Rightarrow \begin{cases} x = r\sin(\phi)\cos(\theta) \\ y = r\sin(\phi)\sin(\theta) \\ z = r\cos(\phi) \end{cases}$$



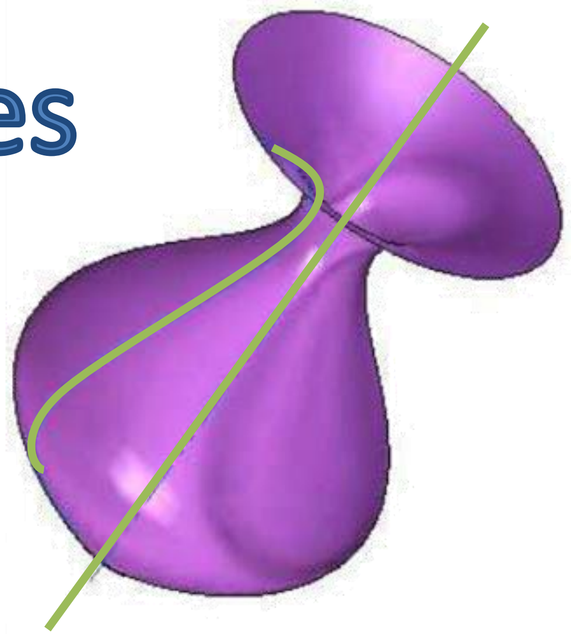
Plan

- Introduction
- Surfaces balayées
- Carreaux surfaciques
- Visualisation OpenGL

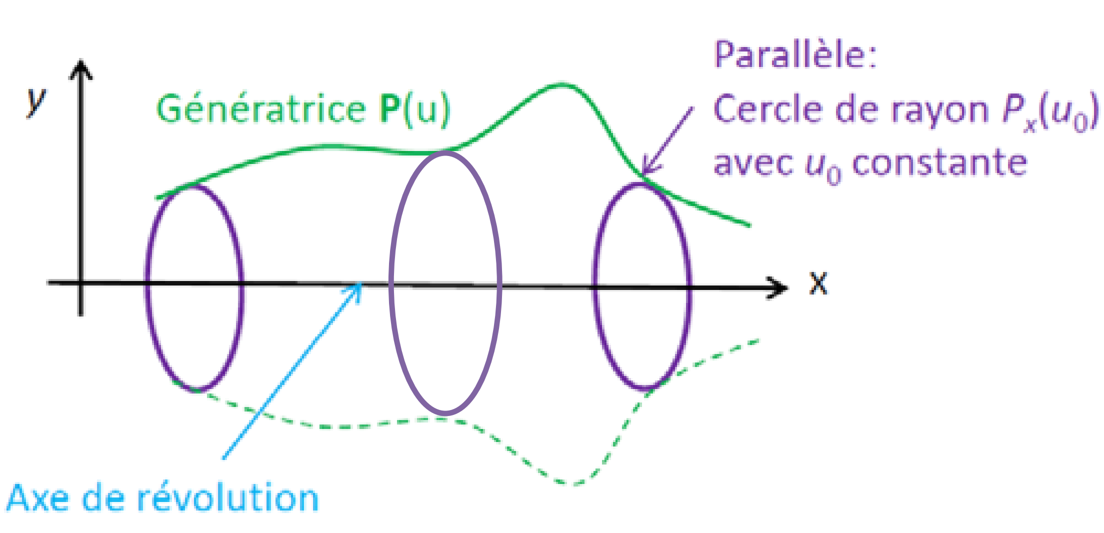
Surfaces balayées

- **Surfaces de révolution**
  - ➢ axe de révolution + génératrice

Surfaces balayées

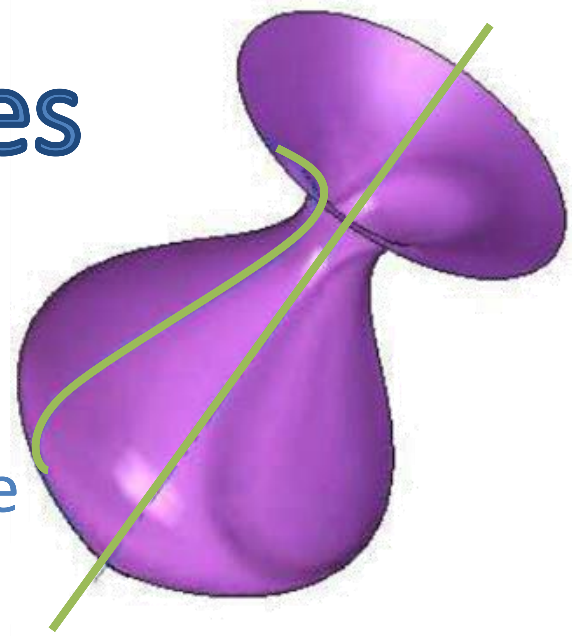


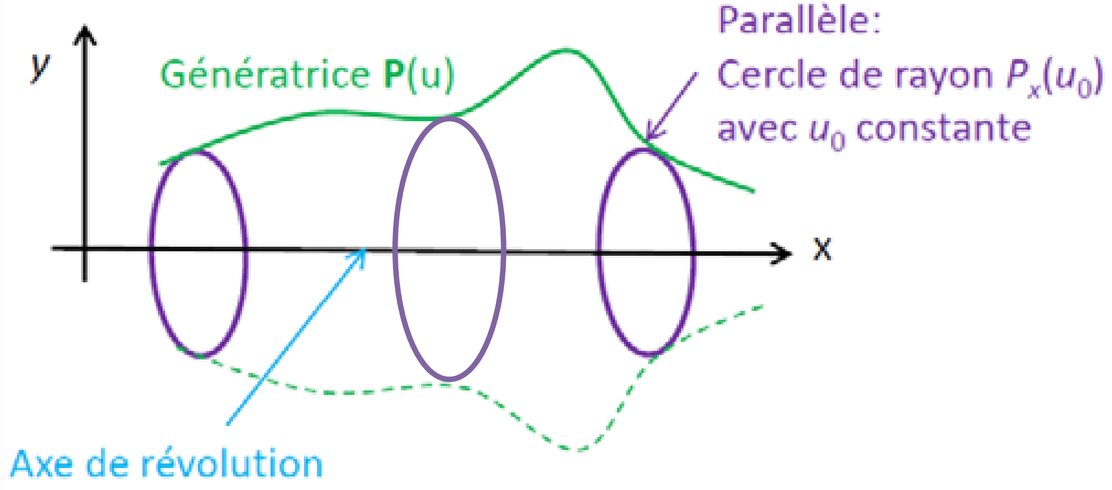
- **Surfaces de révolution**
  - axe de révolution + génératrice



Surfaces balayées

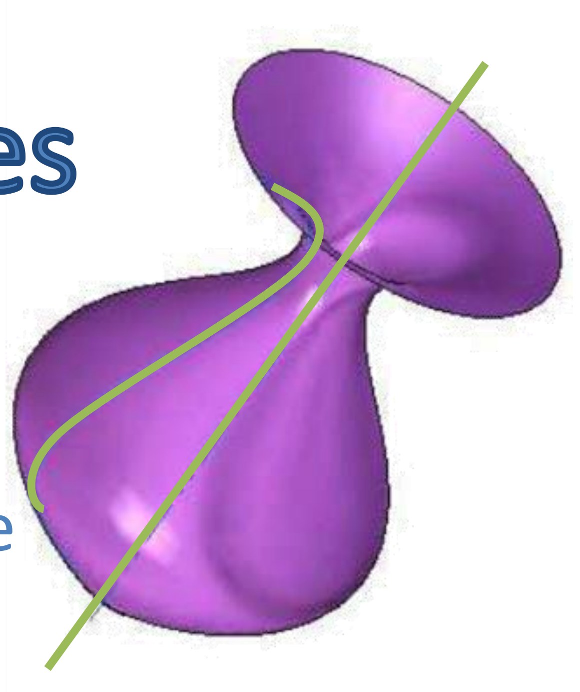
- **Surfaces de révolution**
  - axe de révolution + génératrice
  + position entre l'axe et la génératrice

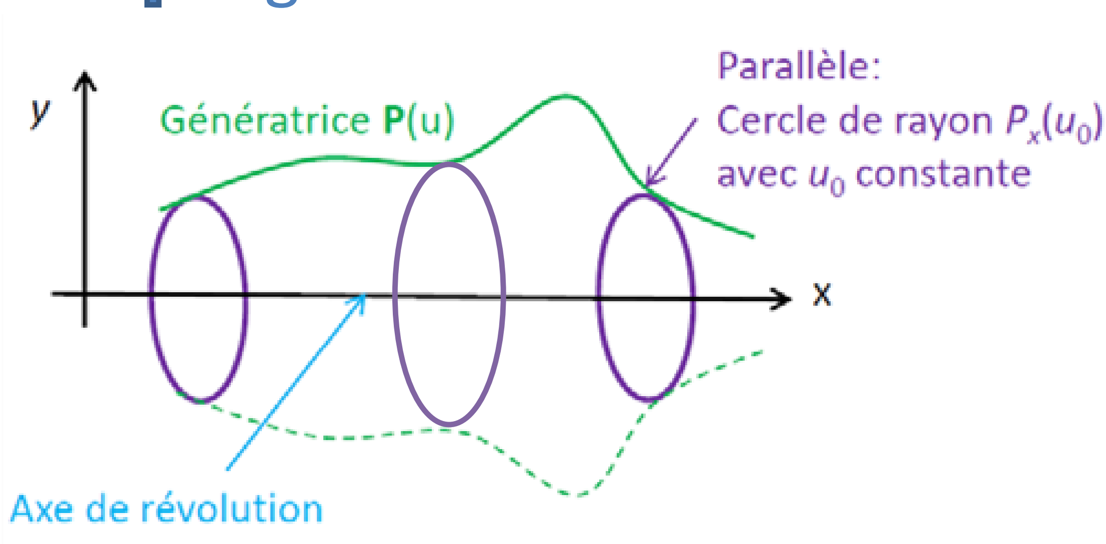




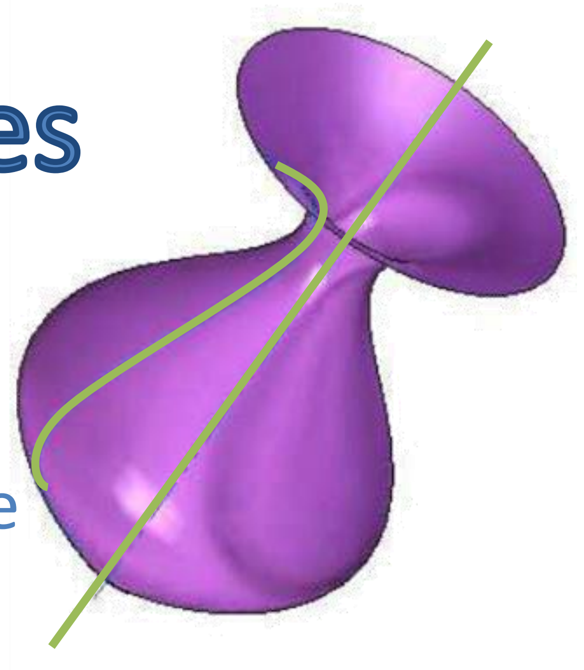
Surfaces balayées

- Surfaces de révolution
  - axe de révolution + génératrice + position entre l'axe et la génératrice + angle de rotation

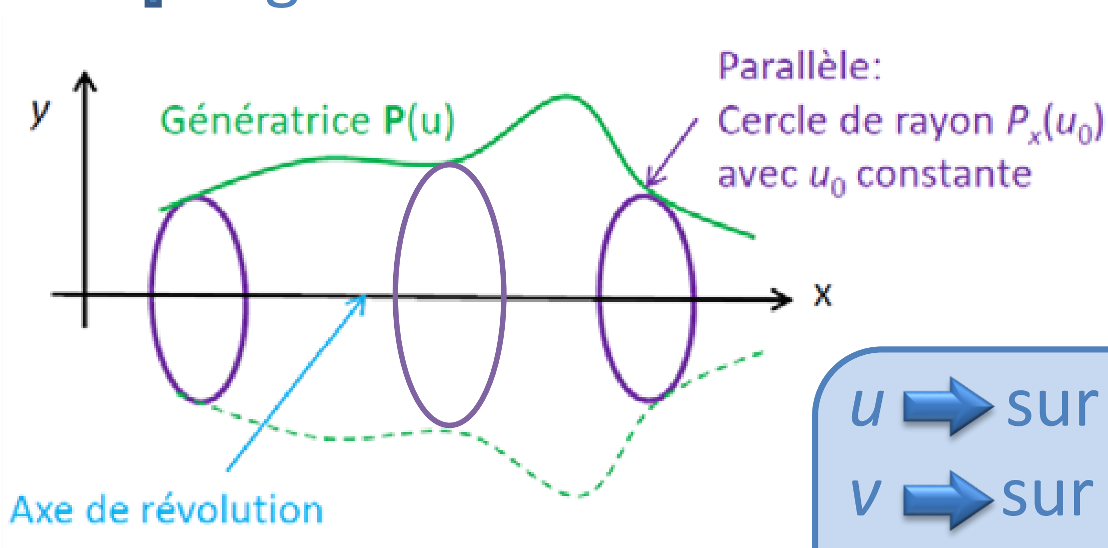




Surfaces balayées



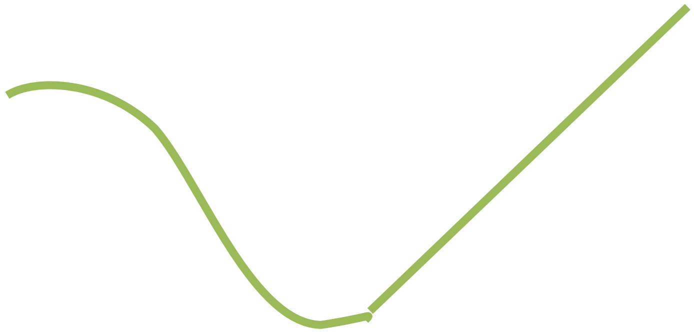
- **Surfaces de révolution**
  - ➤ axe de révolution ✚ génératrice
  ✚ position entre l'axe et la génératrice
  ✚ angle de rotation



u ➡ sur la génératrice
v ➡ sur l'angle (position sur le cercle correspondant à *u*)

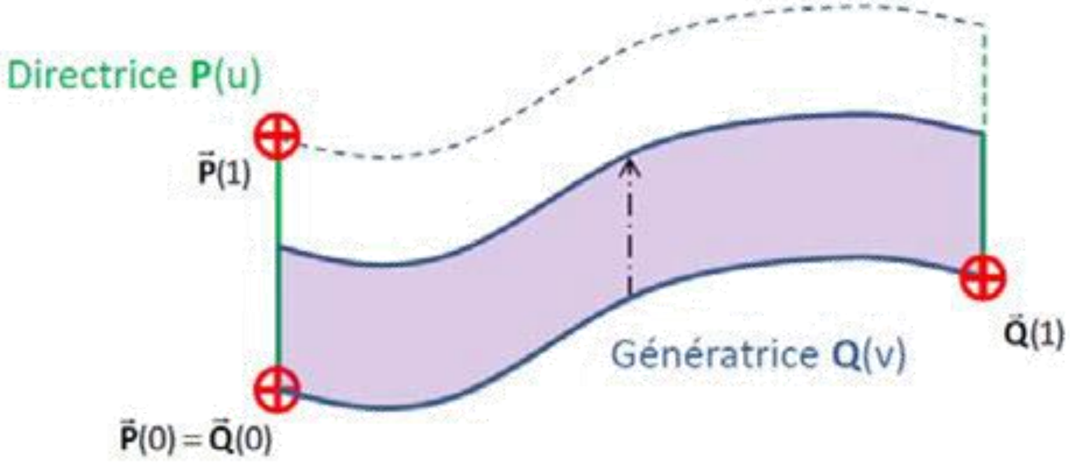
Surfaces balayées

- **Surfaces cylindriques**
  - droite directrice + génératrice



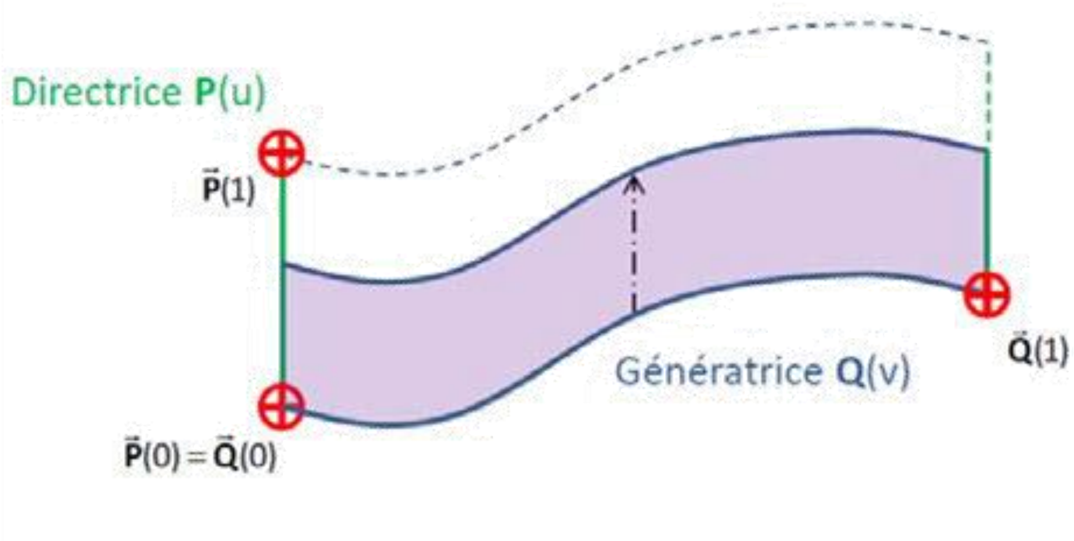
Surfaces balayées

- **Surfaces cylindriques**
  - droite directrice + génératrice



Surfaces balayées

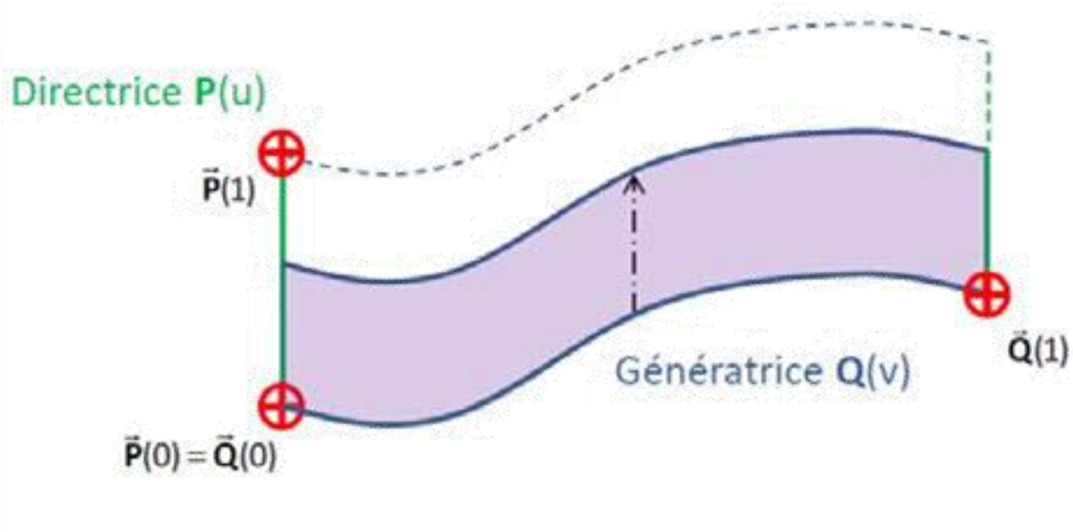
- **Surfaces cylindriques**
  - ➢ droite directrice + génératrice



u ➡ sur la droite
v ➡ sur la génératrice

Surfaces balayées

- **Surfaces cylindriques**
  - droite directrice + génératrice

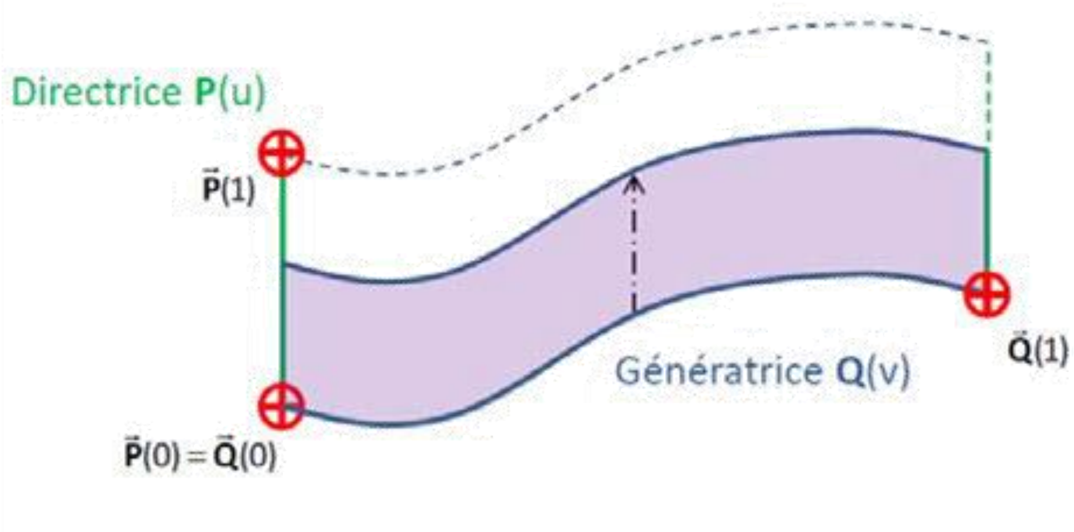


u ➡ sur la droite
v ➡ sur la génératrice

- si courbe = cercle ➡ surface = cylindre

Surfaces balayées

- Surfaces cylindriques
  - ➢ droite directrice + génératrice



u ➡ sur la droite
v ➡ sur la génératrice

➢ si courbe = cercle ➡ surface = cylindre

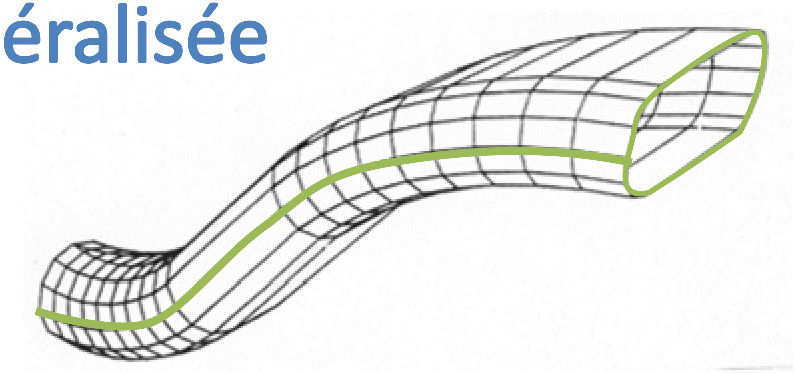
➢ droite + esquisse (ou profil) ➡ surface = extrusion

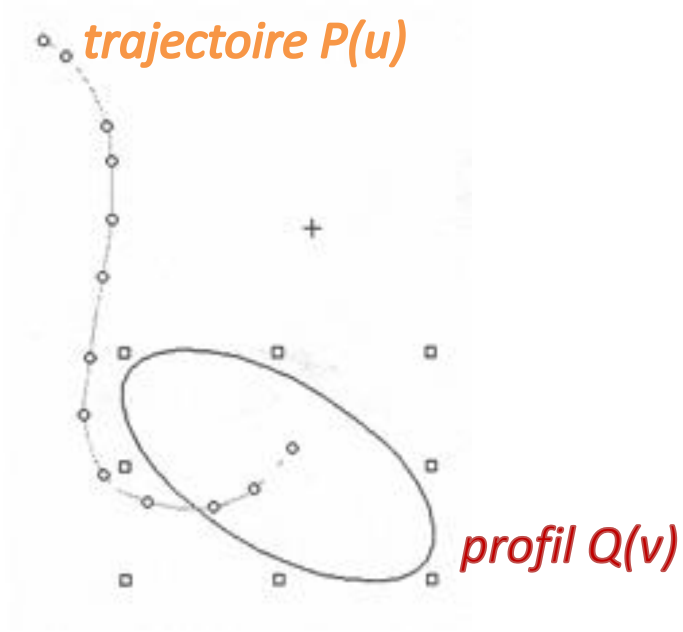
Surfaces balayées

- **Surfaces d'extrusion généralisée**
  - courbe trajectoire
  - + courbe plane fermée

Surfaces balayées

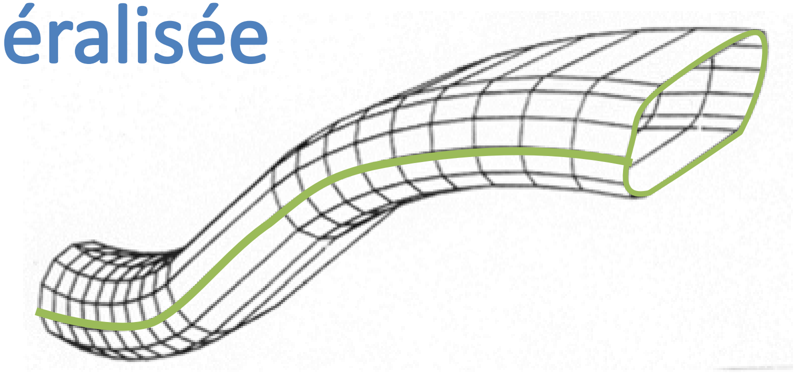
- **Surfaces d'extrusion généralisée**
  - courbe trajectoire
  - + courbe plane fermée

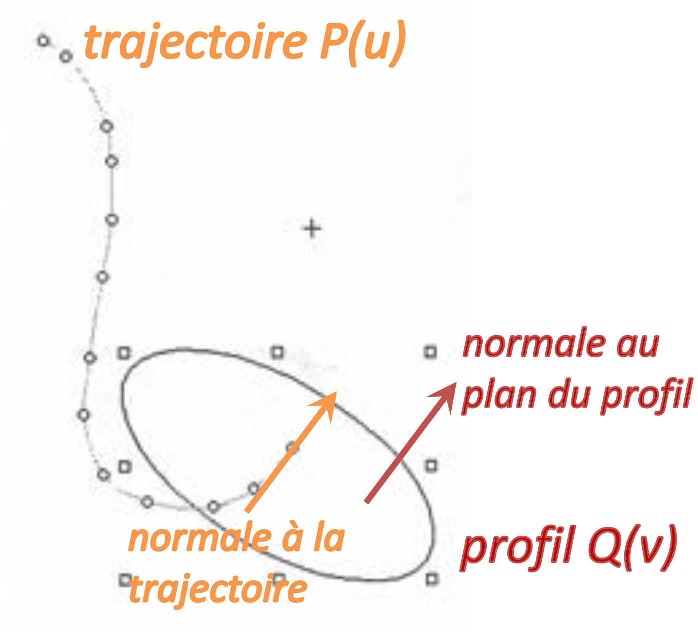




Surfaces balayées

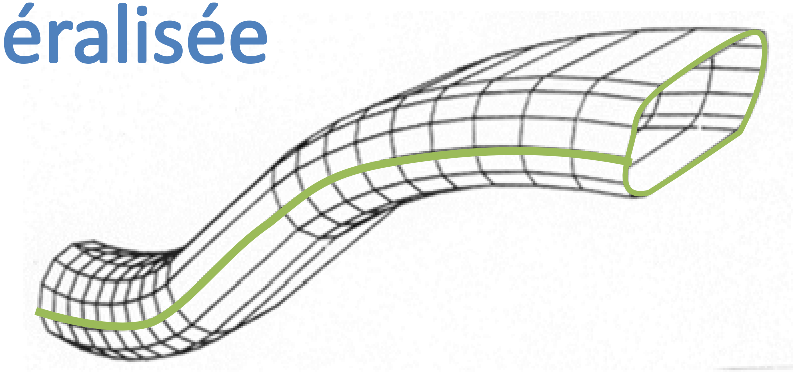
- Surfaces d'extrusion généralisée
  - ➢ courbe trajectoire
  - ✚ courbe plane fermée

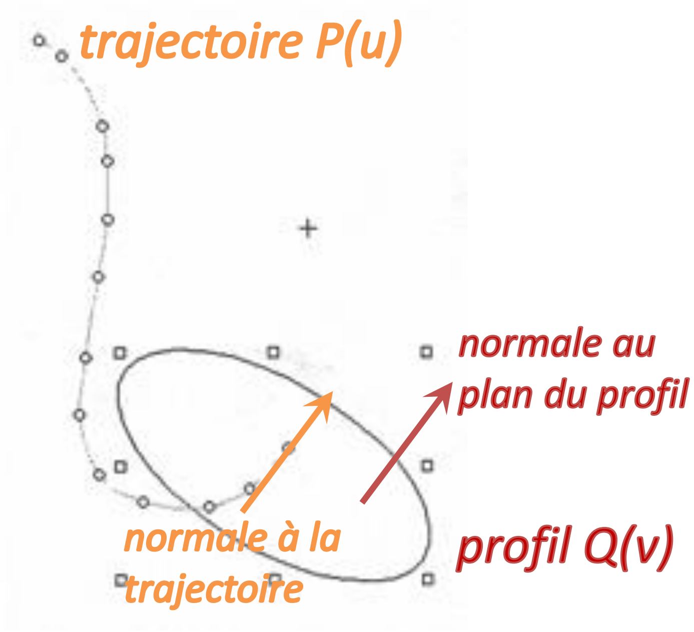




Surfaces balayées

- **Surfaces d'extrusion généralisée**
  - ➤ courbe trajectoire
  - ✚ courbe plane fermée

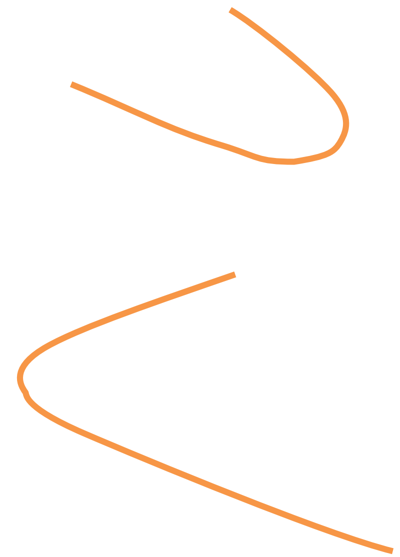




u ➡ sur la trajectoire
v ➡ sur le profil

Surfaces balayées

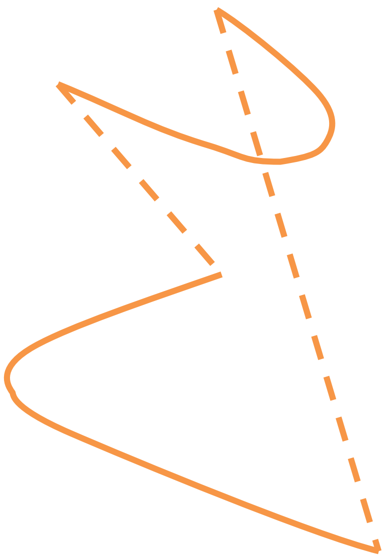
- **Surfaces réglées**
  - deux courbes directrices



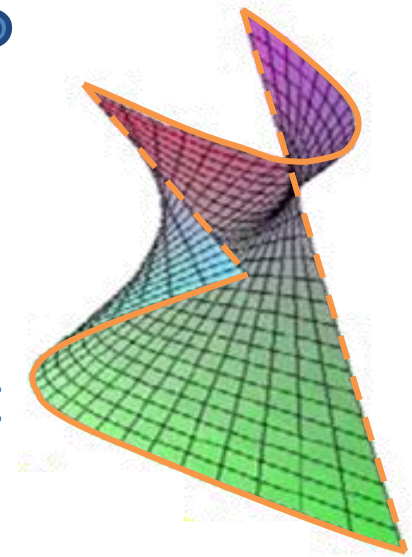
Surfaces balayées

- **Surfaces réglées**
  - deux courbes directrices

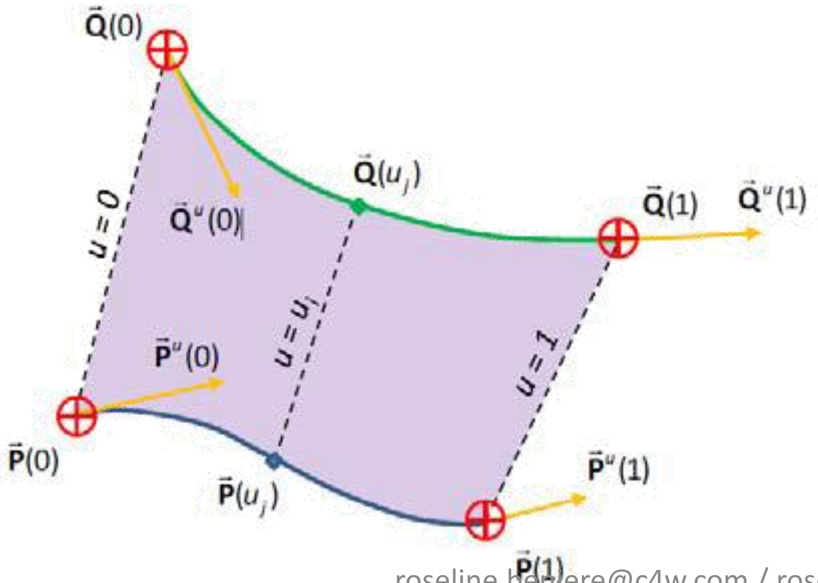
➡ un segment de droite entre un point de la première courbe et un point de la seconde courbe au même *u*.



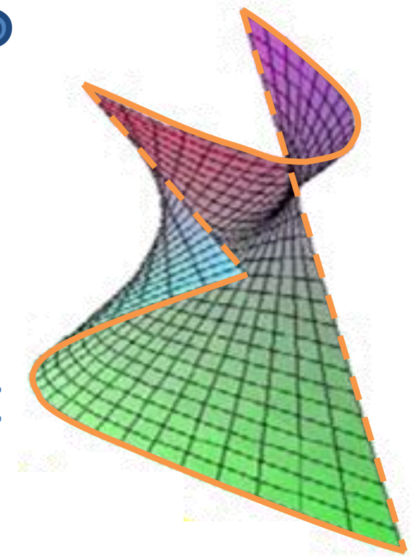
Surfaces balayées



- **Surfaces réglées**
  - ➢ deux courbes directrices
  - ➡ un segment de droite entre un point de la première courbe et un point de la seconde courbe au même *u*.

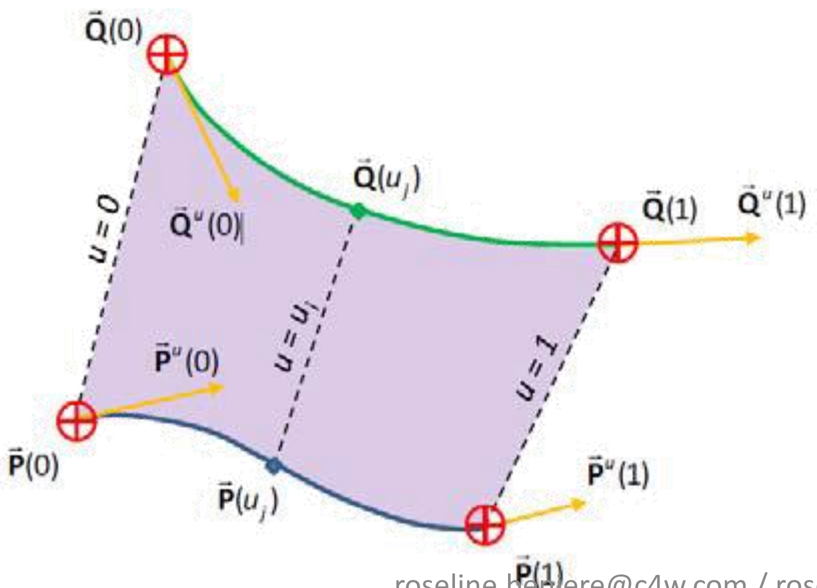


Surfaces balayées



- **Surfaces réglées**
  - ➢ deux courbes directrices

➡ un segment de droite entre un point de la première courbe et un point de la seconde courbe au même *u*.



u ➡ sur la courbe 1 et
sur la courbe 2
v ➡ sur le segment

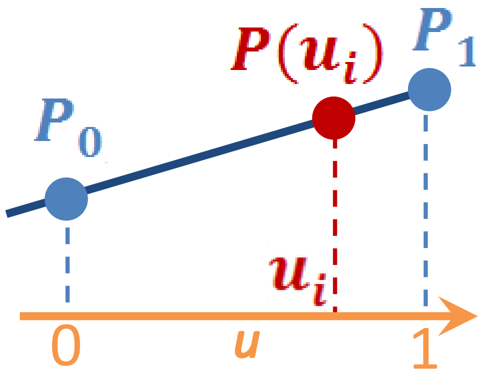
Surfaces balayées

- **Surfaces réglées (formule mathématique)**
  - deux courbes directrices $P(u)$ et $Q(u)$ définies dans l'espace 3D, en fonction d'un paramètre u variant de 0 à1
  - si $P(u)$ et $Q(v)$, il faut faire un changement de variable $v = f(u)$
  - un segment de droite entre $\boldsymbol{P}(\boldsymbol{u_i})$ et $\boldsymbol{Q}(\boldsymbol{u_i})$

Surfaces balayées

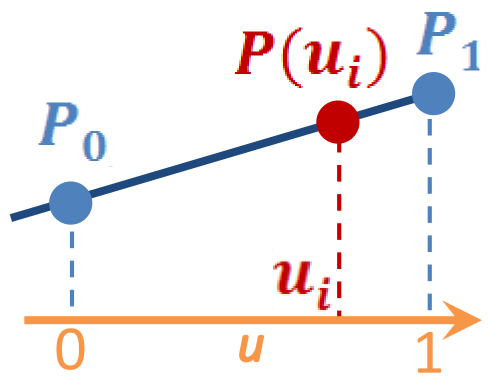
- **Surfaces réglées (formule mathématique)**
  - rappel équation de la droite 3D

$$\boxed{P(u) = (1-u)P_0 + uP_1}$$



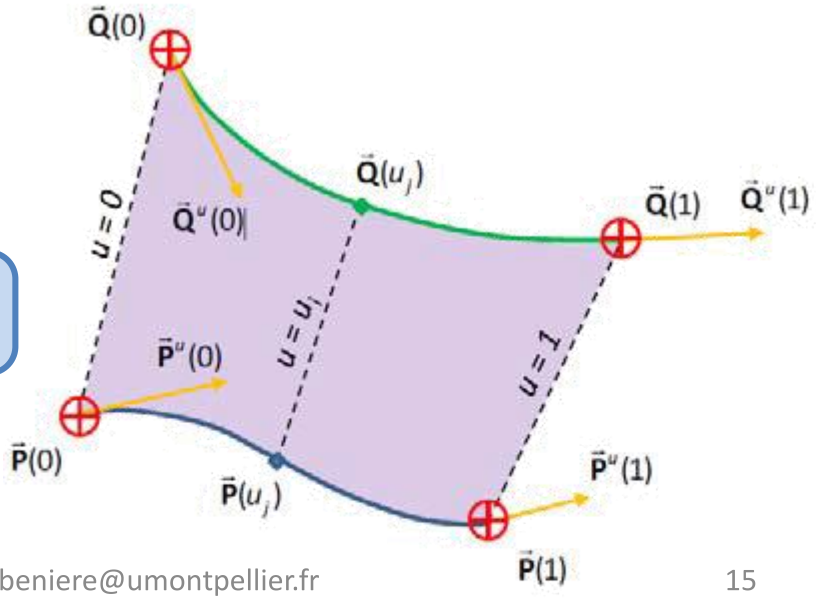
Surfaces balayées

- **Surfaces réglées (formule mathématique)**
  - rappel équation de la droite 3D



$$\boxed{P(u) = (1-u)P_0 + uP_1}$$

  - *u* sur les courbes
  - *v* sur le segment

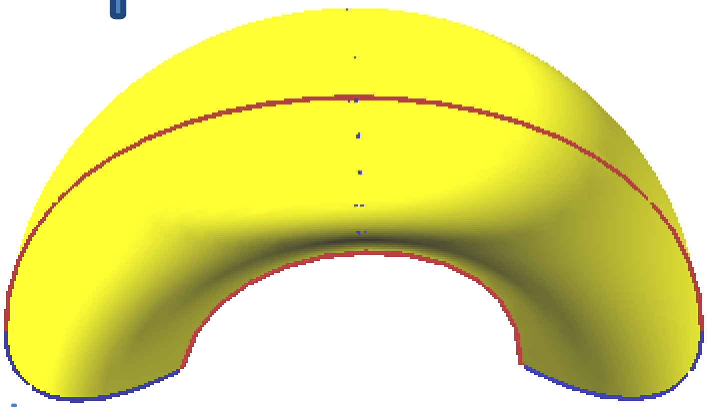


$$\boxed{P(u,v) = (1-v)P(u) + vQ(u)}$$

Plan

- Introduction
- Surfaces balayées
- Carreaux surfaciques
- Visualisation OpenGL

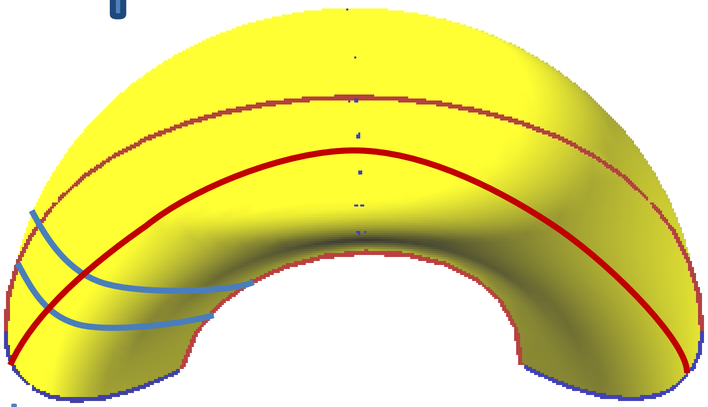
Carreaux surfaciques



- **Caractéristiques générales :**
  - élément de bases de la surfaces complexes
  - équivalent aux segments des courbes
  - un carreaux faces est considéré comme **bi-paramétrique** puisqu'il est définit en fonction de deux paramètres

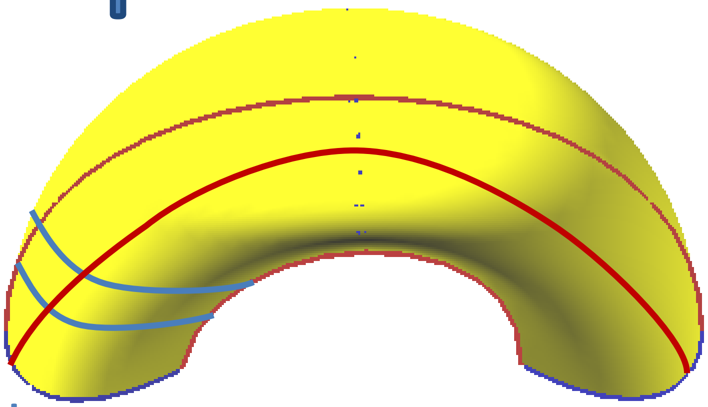
roseline.beniere@c4w.com / roseline.beniere@umontpellier.fr

Carreaux surfaciques



- **Caractéristiques générales :**
  - ➢ élément de bases de la surfaces complexes
  - ➢ équivalent aux segments des courbes
  - ➢ un carreaux faces est considéré comme **bi-paramétrique** puisqu'il est définit en fonction de deux paramètres
  - ➢ en fixant *u* ou *v*, on génère une courbe **iso-paramétrique** sur la surface définie en fonction du 2ème paramètre

Carreaux surfaciques



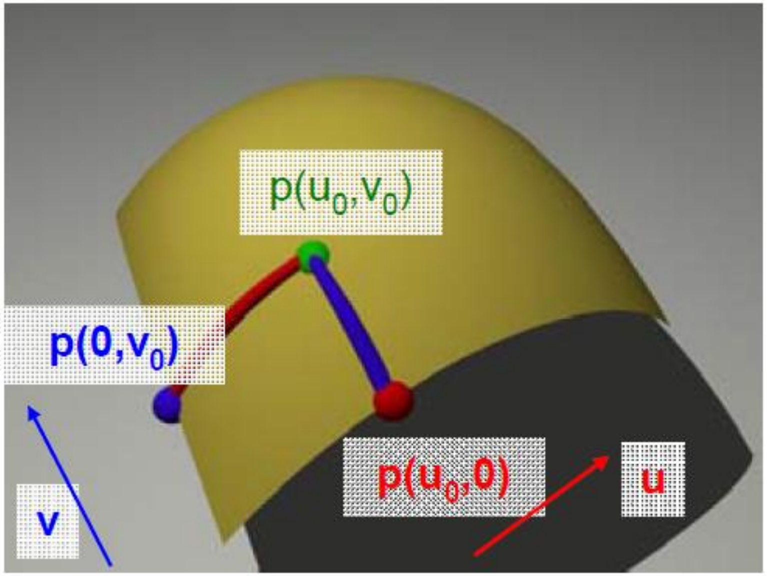
- **Caractéristiques générales :**
  - élément de bases de la surfaces complexes
  - équivalent aux segments des courbes
  - un carreaux faces est considéré comme **bi-paramétrique** puisqu'il est définit en fonction de deux paramètres
  - en fixant *u* ou *v*, on génère une courbe **iso-paramétrique** sur la surface définie en fonction du 2ème paramètre
  - une surface est ainsi décrite par un réseau de courbes iso-paramétriques.

Carreaux surfaciques

- **Caractéristiques générales :**
  - une surface paramétrique est définie par :

$$P(u,v) = \begin{cases} x = x(u,v) \\ y = y(u,v) \\ z = z(u,v) \end{cases}$$

  - le point $P(u,v)$ est l'intersection entre 2 courbes iso-paramétriques celle en u constant (en bleu) et celle en v constant (en rouge)



Carreaux surfaciques

- **Principe ➡ produit tensoriel :**
  - une façon de construire une surface paramétrique est de faire le produit tensoriel entre 2 courbes paramétriques.

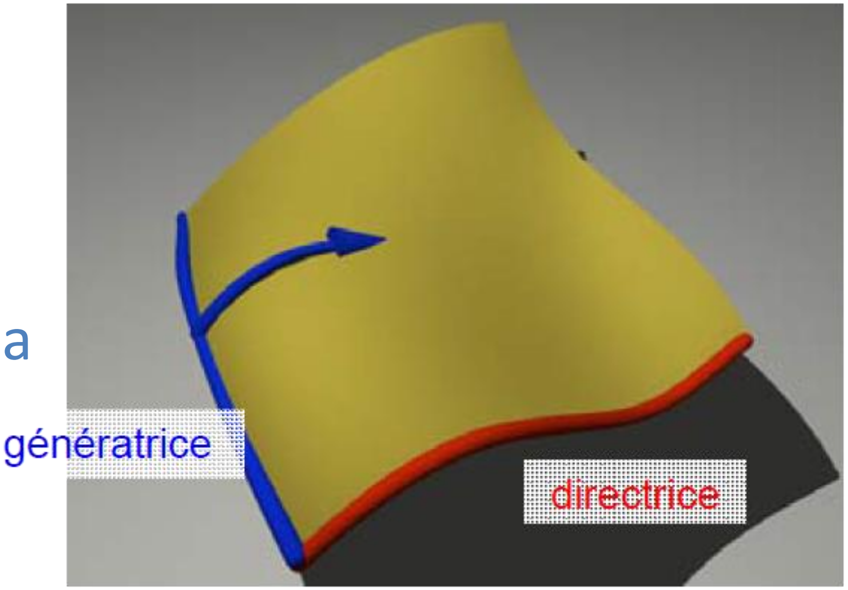
Carreaux surfaciques

- **Principe ➡ produit tensoriel :**
  - une façon de construire une surface paramétrique est de faire le produit tensoriel entre 2 courbes paramétriques.
  - Produit tensoriel : $A \otimes B$

$$A \begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix} \qquad B \begin{vmatrix} 5 & 6 \\ 7 & 8 \end{vmatrix} \Rightarrow A \otimes B \begin{vmatrix} 1x5 & 1x6 & 2x5 & 2x6 \\ 1x7 & 1x8 & 2x7 & 2x8 \\ 3x5 & 3x6 & 4x5 & 4x6 \\ 3x7 & 3x8 & 4x7 & 4x8 \end{vmatrix}$$

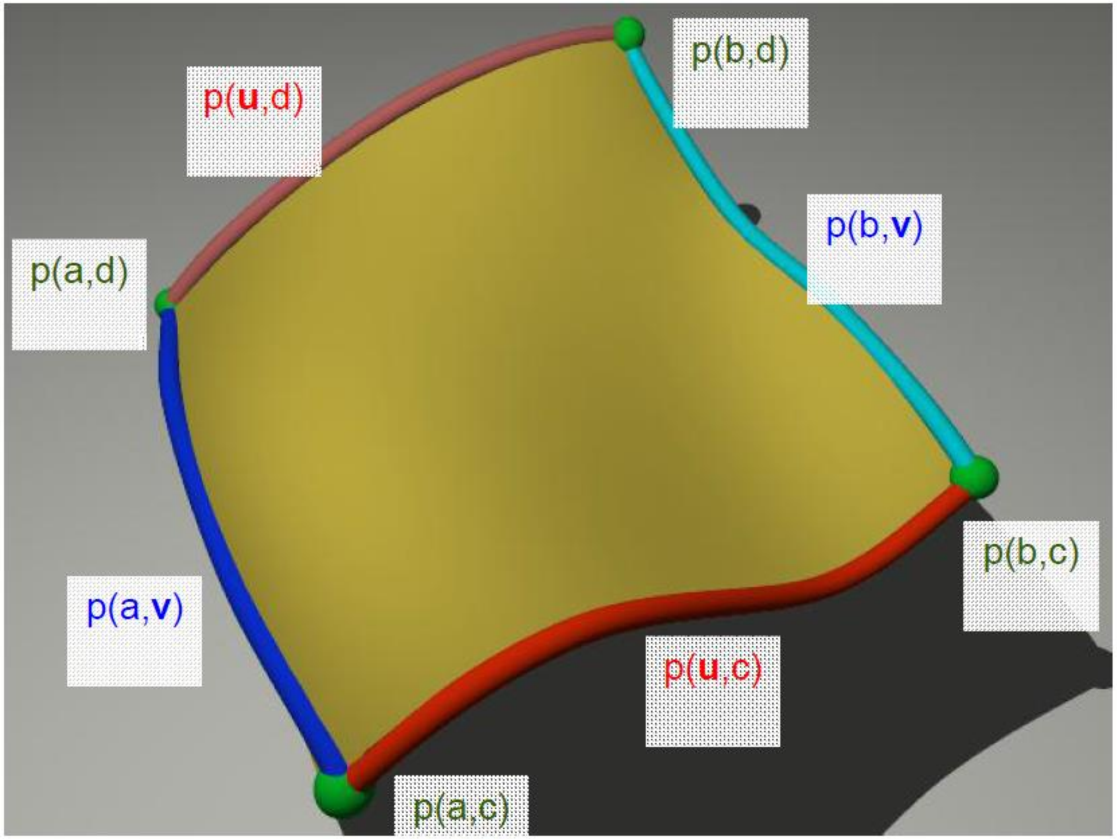
Carreaux surfaciques

- **Principe ➡ produit tensoriel :**
  - une façon de construire une surface paramétrique est de faire le produit tensoriel entre 2 courbes paramétriques.
  - la première courbe $d(u)$ est appelée la courbe directrice et la seconde $g(v)$ est appelée la courbe génératrice.
  - la surface est obtenue en ``déplaçant'' et en ``déformant'' la courbe génératrice le long de la courbe directrice.



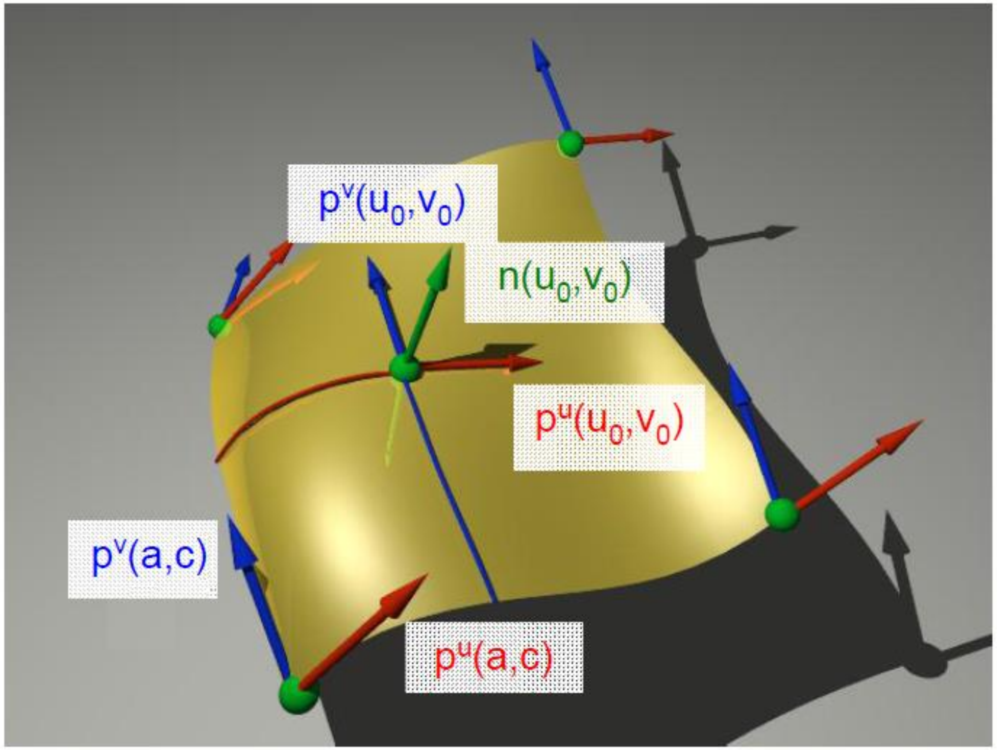
Carreaux surfaciques

- **Coins et bords :**
  - soit un carreau (u,v) dans [a,b]x[c,d]



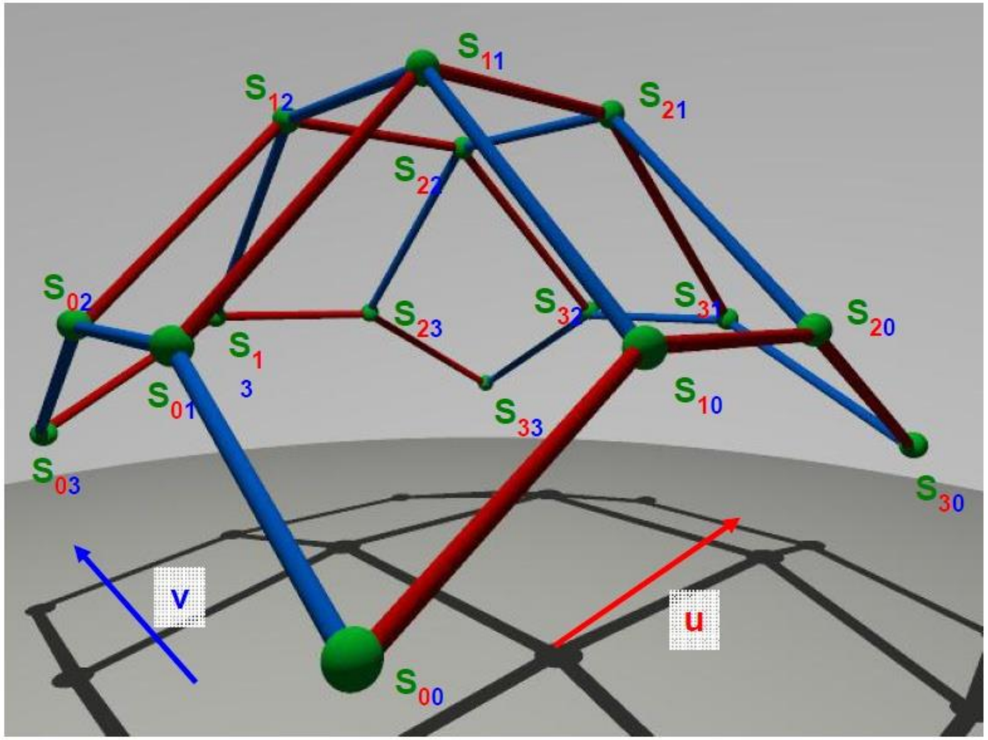
Carreaux surfaciques

- **Tangentes et normales :**
  - un plan tangent peut être défini en chaque point, en utilisant les deux tangentes aux 2 courbes iso-paramétriques du point,
  - la normale à chaque point est obtenue grâce au produit vectoriel des 2 tangentes.



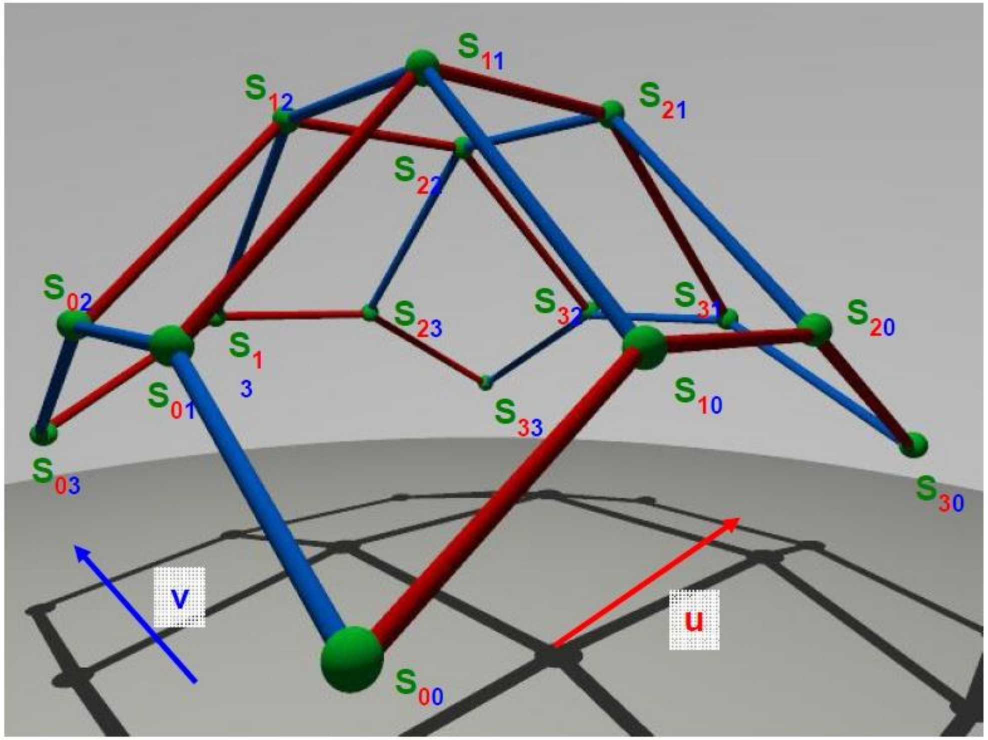
Carreaux surfaciques

- **Carreaux de Bézier :**
  - la surface est contrôlée à partir d'une ``grille'' régulière composée de quadrilatère.
  - les points de contrôles les sommets de la grille et il sont numérotés dans les directions *u* et *v*.



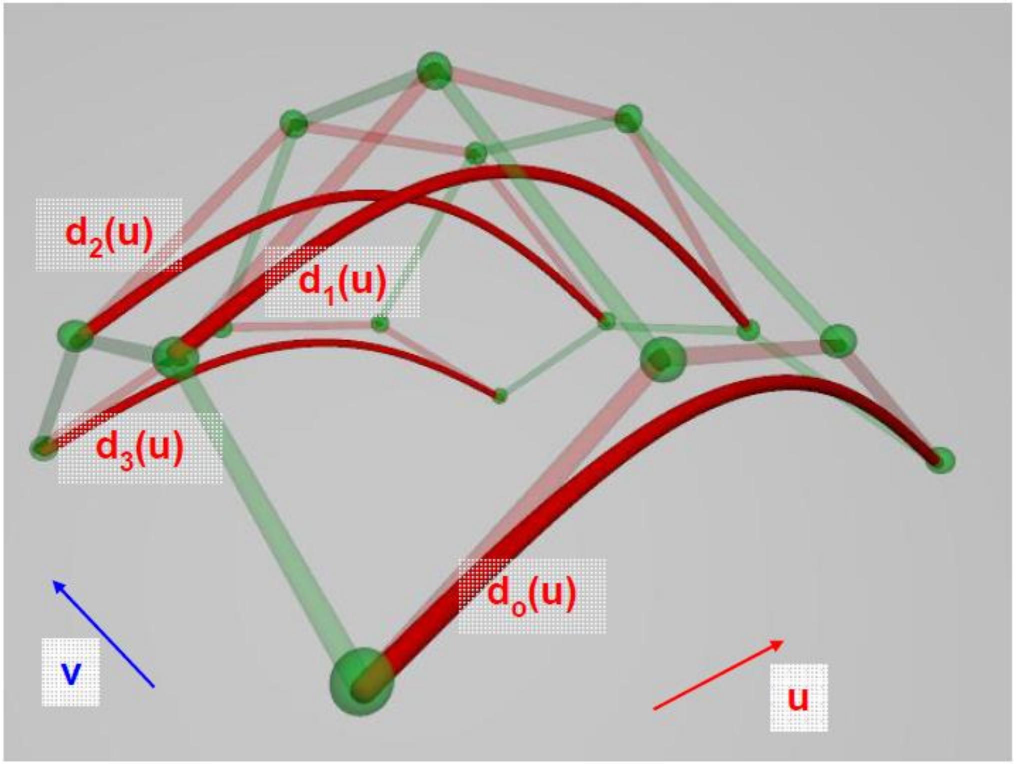
Carreaux surfaciques

- **Carreaux de Bézier :**
  - *u* et *v* sont généralement compris entre 0 et 1.
  - les (n+1) points de contrôle en *u* donnent le degré **n** des courbes *u*
    ici n = 3
  - les (m+1) points de contrôle en *v* donnent le degré **m** des courbes *v*
    ici m=3



Carreaux surfaciques

- **Carreaux de Bézier :**
  - pour calculer un point *P*(*u*,*v*), on utilise le produit tensoriel
  - si les courbes directrices sont dans le sens de *u*
    - il y a une courbe directrice par polygone dans le sens de *u*
    - la directrice est une courbe de Bézier basée sur les points de contrôle de la grille

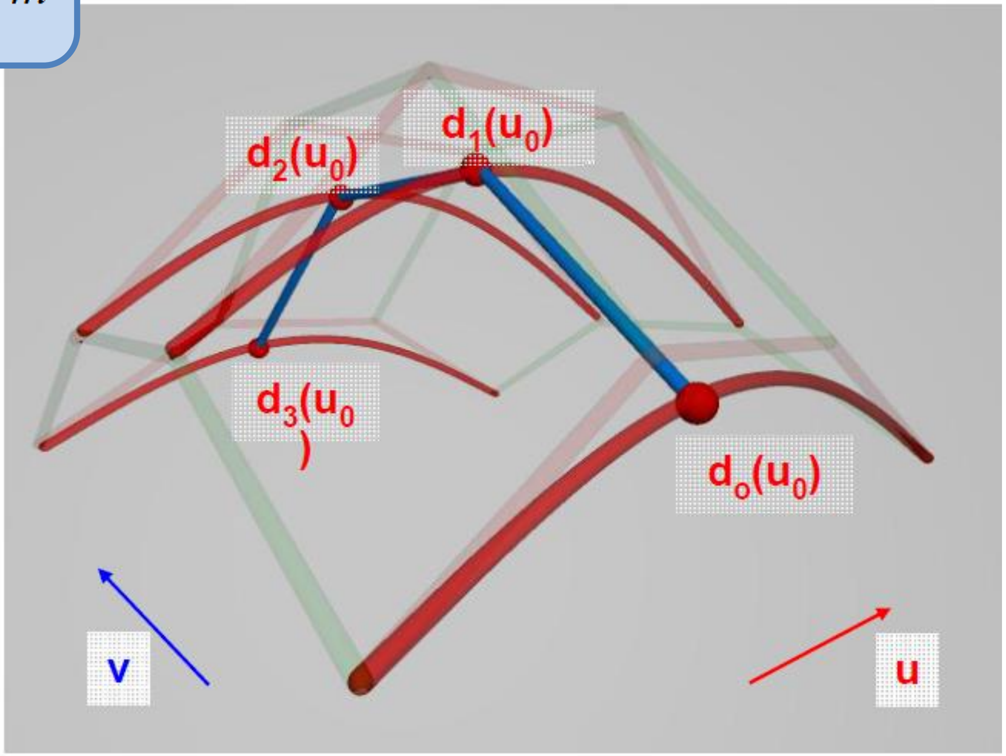


Carreaux surfaciques

- **Carreaux de Bézier :**
  - pour chaque directrice (en rouge) on évalue un point

$$d_j(u_0)=\sum_{i=0}^{n} B_i^n(u_0)\, S_{ij} \quad , \qquad j=0..m$$

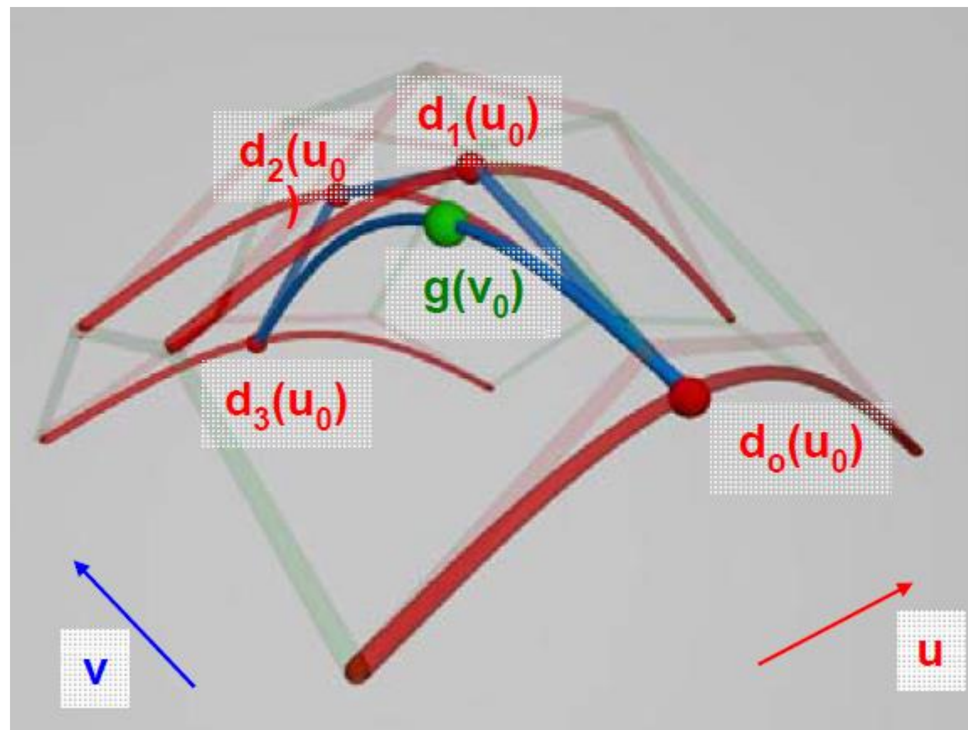
  - ces points sont les points de contrôles de la courbe génératrice (en bleu)



Carreaux surfaciques

- **Carreaux de Bézier :**
  - La génératrice est sur la surface et le point de la surface est alors celui de la courbe génératrice *g*(*v*)

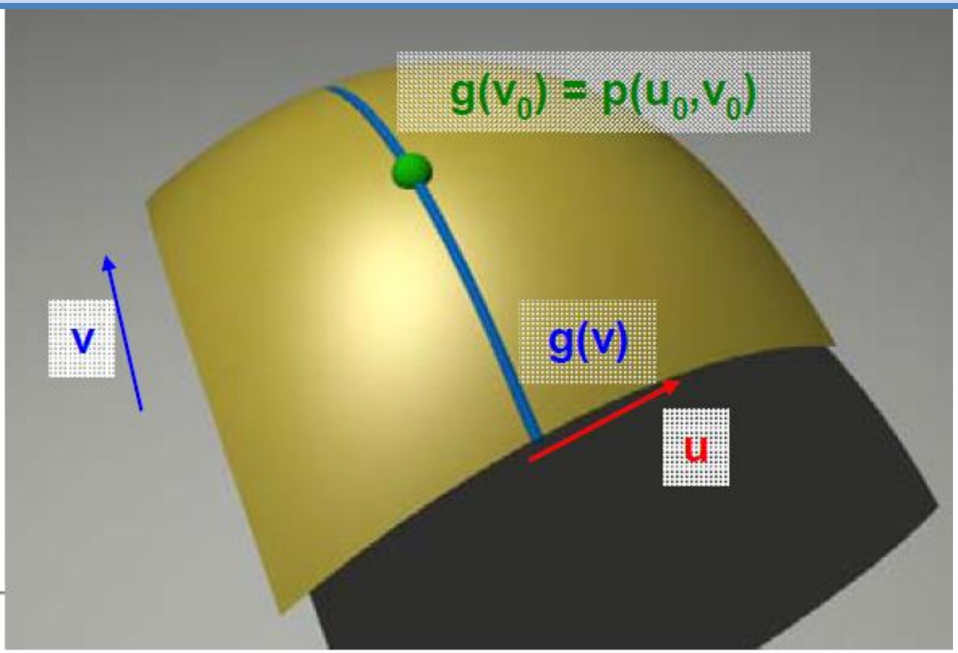
$$\boxed{g(v_0)=\sum_{j=0}^{m} B_j^m(v_0)\, d_j(u_0)}$$



Carreaux surfaciques

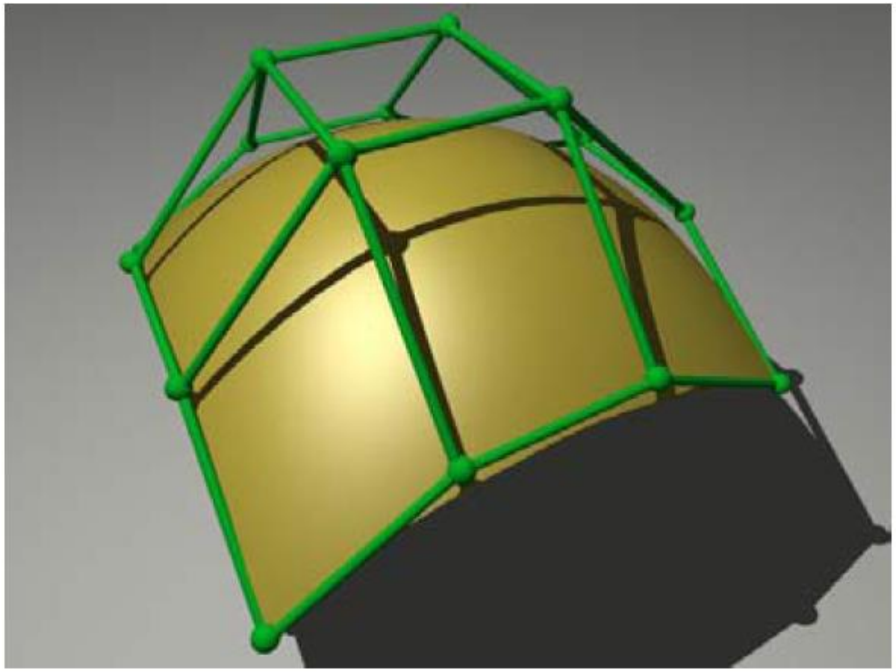
- **Carreaux de Bézier :**
  - en reportant l'équation des *dj*(*u*) dans *g*(*v*)

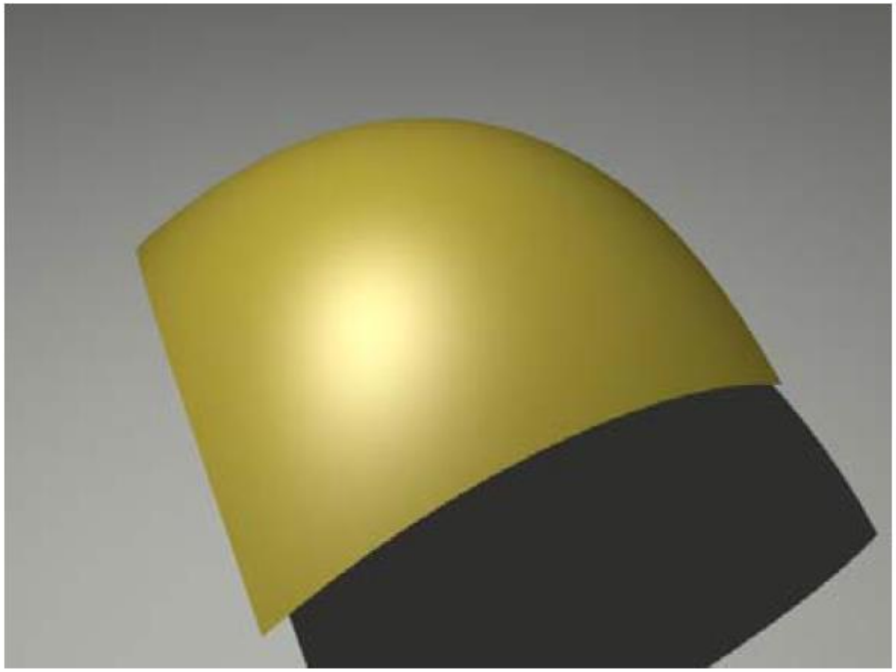
$$p(u,v)=g(v)=\sum_{i=0}^{n}\sum_{j=0}^{m}B_i^n(u)\,B_j^m(v)\,S_{ij}$$



Carreaux surfaciques

- **Carreaux de Bézier :**
  - ➢ un carreau de Bézier est donc défini par une grille de points de contrôle et les polynômes de Bernstein.

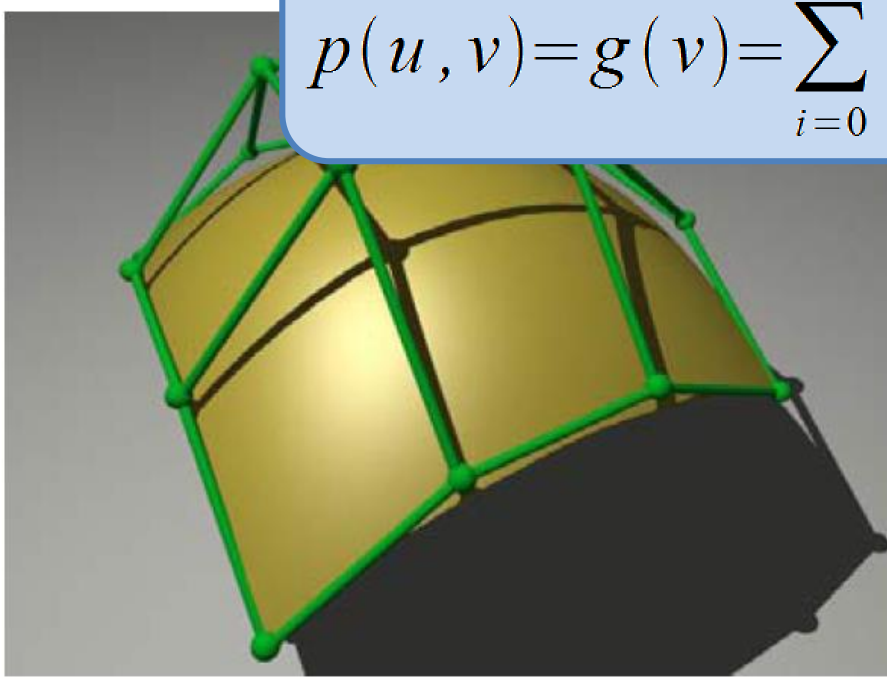


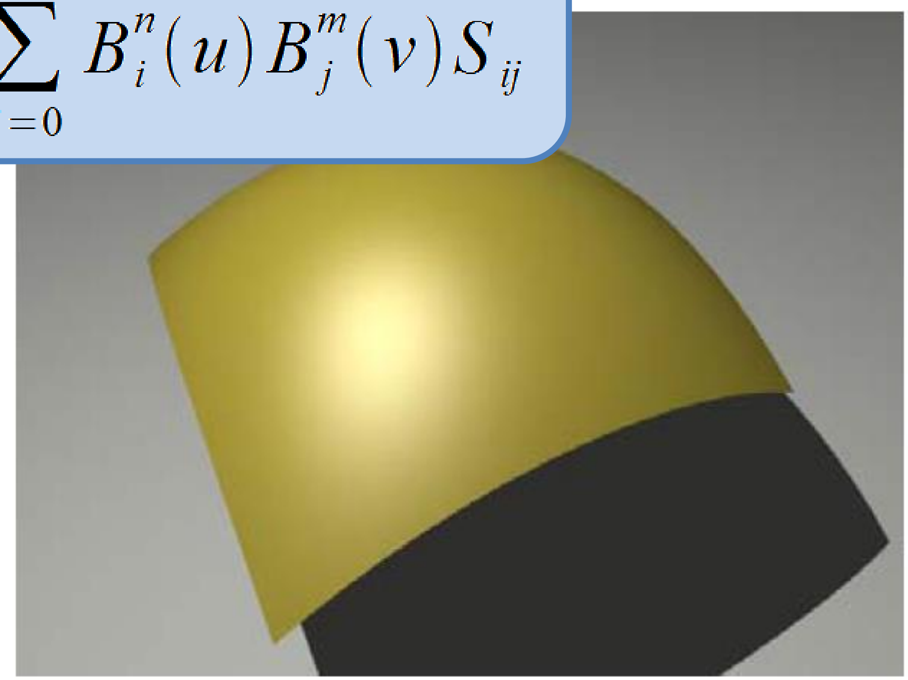


Carreaux surfaciques

- **Carreaux de Bézier :**
  - ➢ un carreau de Bézier est donc défini par une grille de points de contrôle et les polynômes de Bernstein.

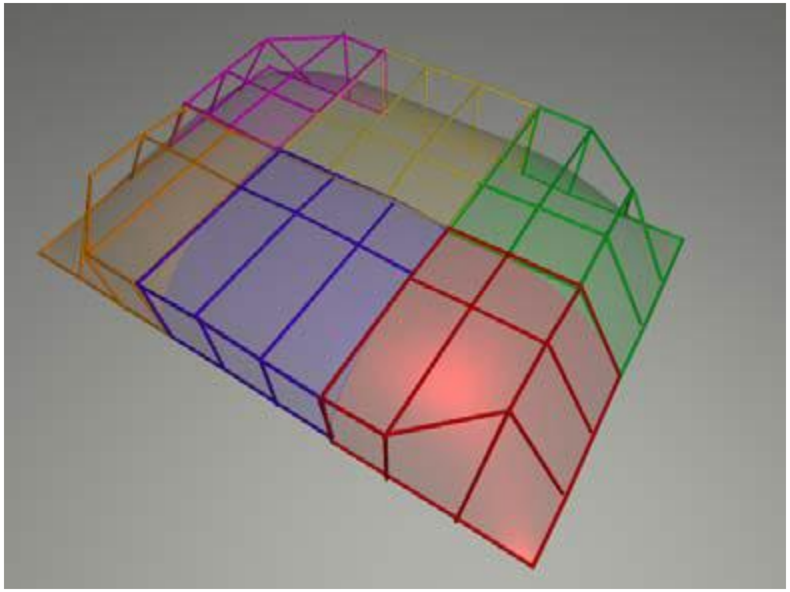
$$p(u,v)=g(v)=\sum_{i=0}^{n}\sum_{j=0}^{m} B_i^n(u)\,B_j^m(v)\,S_{ij}$$





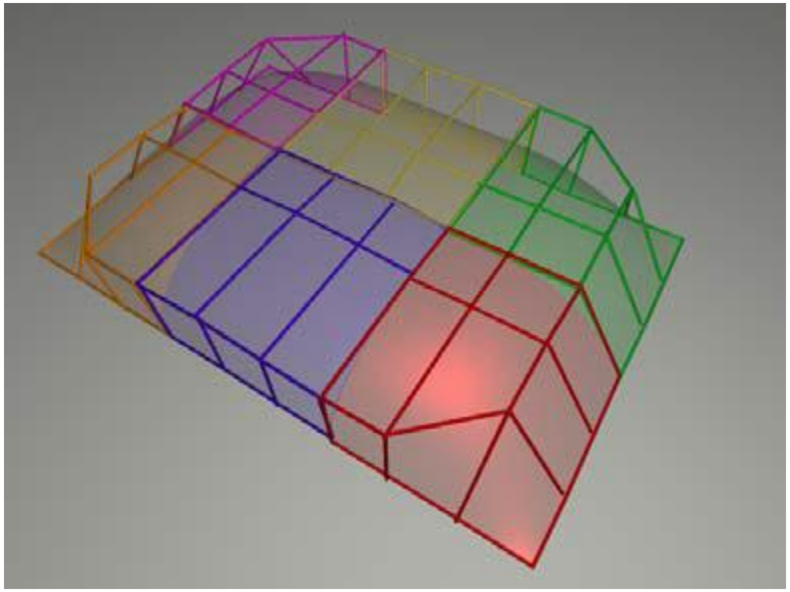
Carreaux surfaciques

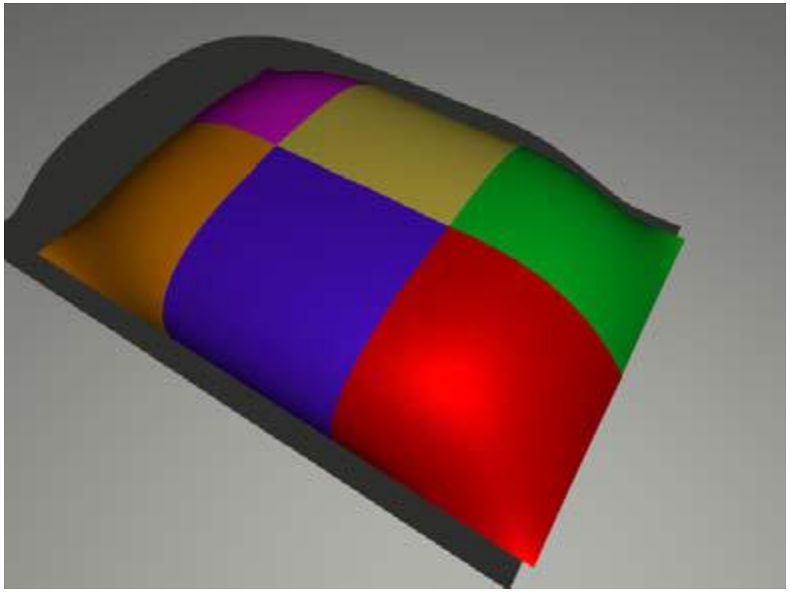
- **Modélisation par carreaux :**
  - modélisation d'un objet par ``couture'' de carreaux,
  - il faut jouer sur la position des points de contrôle pour obtenir de ``belles'' continuités.



Carreaux surfaciques

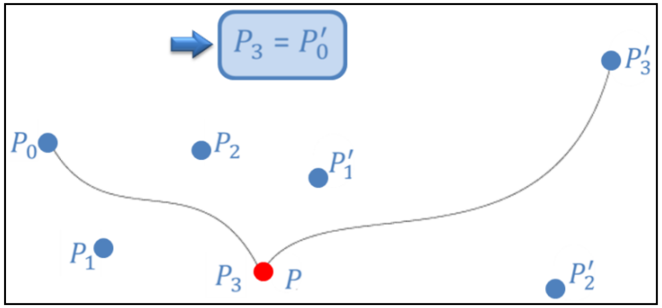
- **Modélisation par carreaux :**
  - ➢ modélisation d'un objet par ``couture'' de carreaux,
  - ➢ il faut jouer sur la position des points de contrôle pour obtenir de ``belles'' continuités.





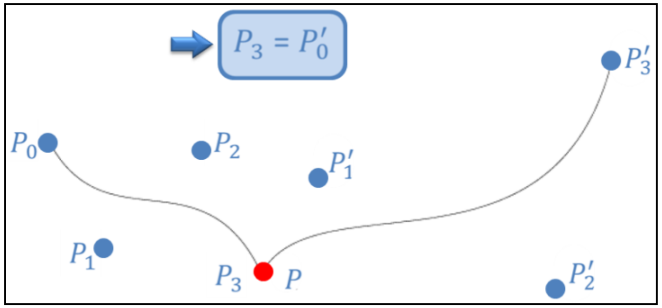
Carreaux surfaciques

- Modélisation par carreaux ➡ continuité
  - Courbes 3D :
    - C_0 ou G_0

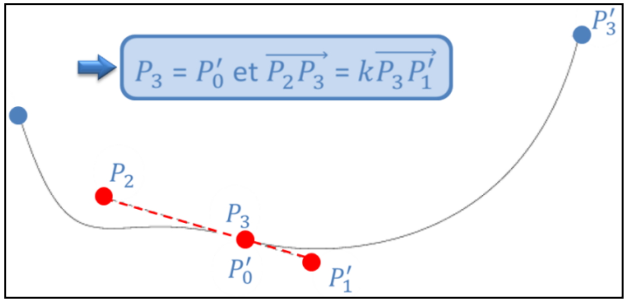


Carreaux surfaciques

- Modélisation par carreaux ➡ continuité
  - ➢ Courbes 3D :
    - C_0 ou G_0

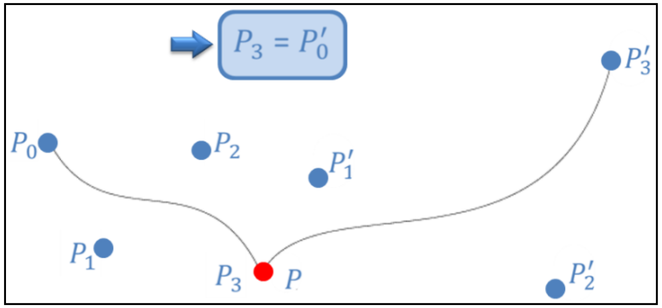


    - G_1

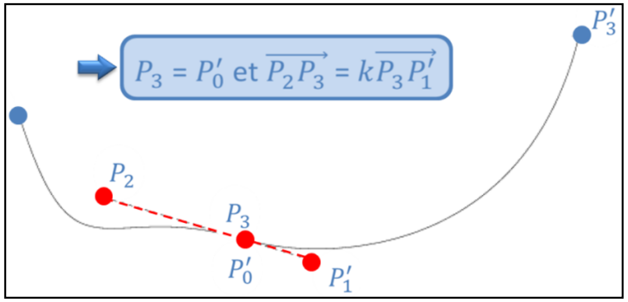


Carreaux surfaciques

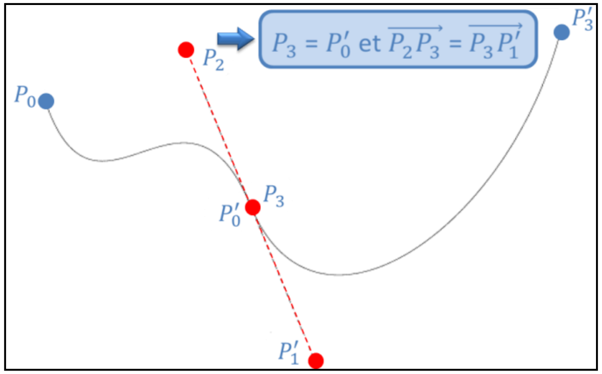
- Modélisation par carreaux ➡ continuité
  - Courbes 3D :
    - C_0 ou G_0



    - G_1

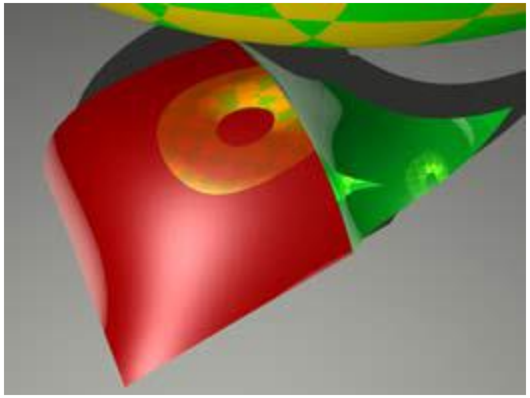


    - C_1



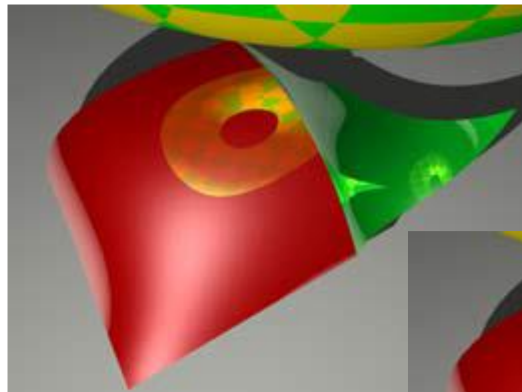
Carreaux surfaciques

- **Modélisation par carreaux ➡ continuité**
  - Surfaces 3D :
    - C_0 ou G_0

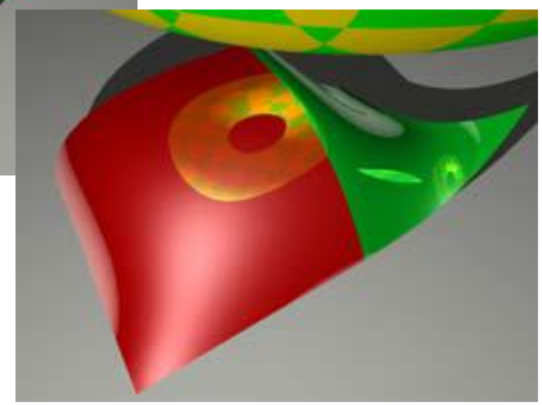


Carreaux surfaciques

- Modélisation par carreaux ➡ continuité
  - Surfaces 3D :
    - C_0 ou G_0

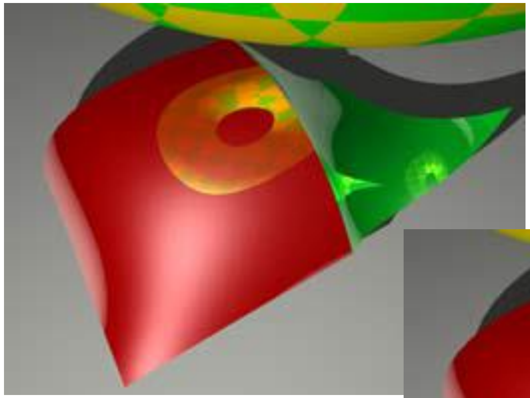


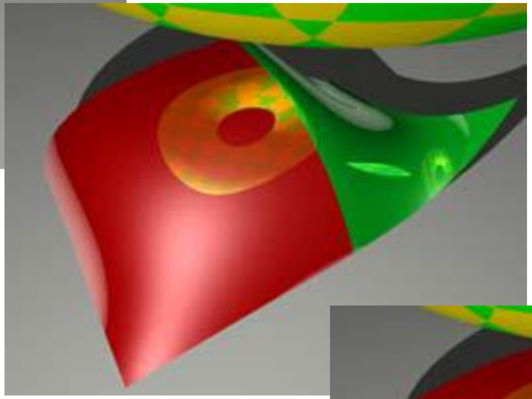
    - G_1

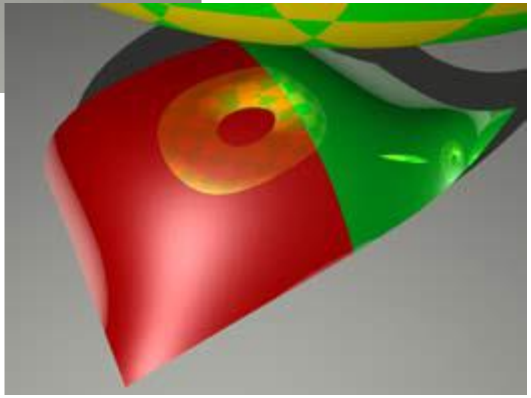


Carreaux surfaciques

- Modélisation par carreaux ➡ continuité
  - Surfaces 3D :
    - C_0 ou G_0
    - G_1
    - C_1

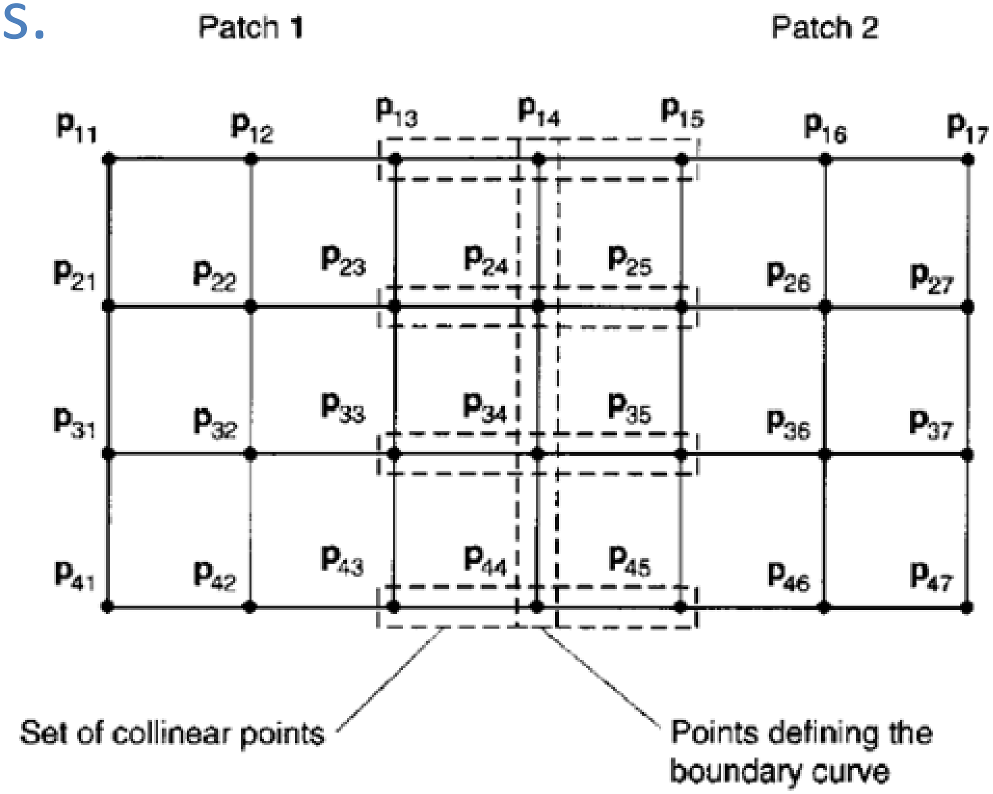






Carreaux surfaciques

- **Modélisation par carreaux ➡ continuité**
  - ➢ Surfaces 3D : même principe points commun et tangentes identiques.

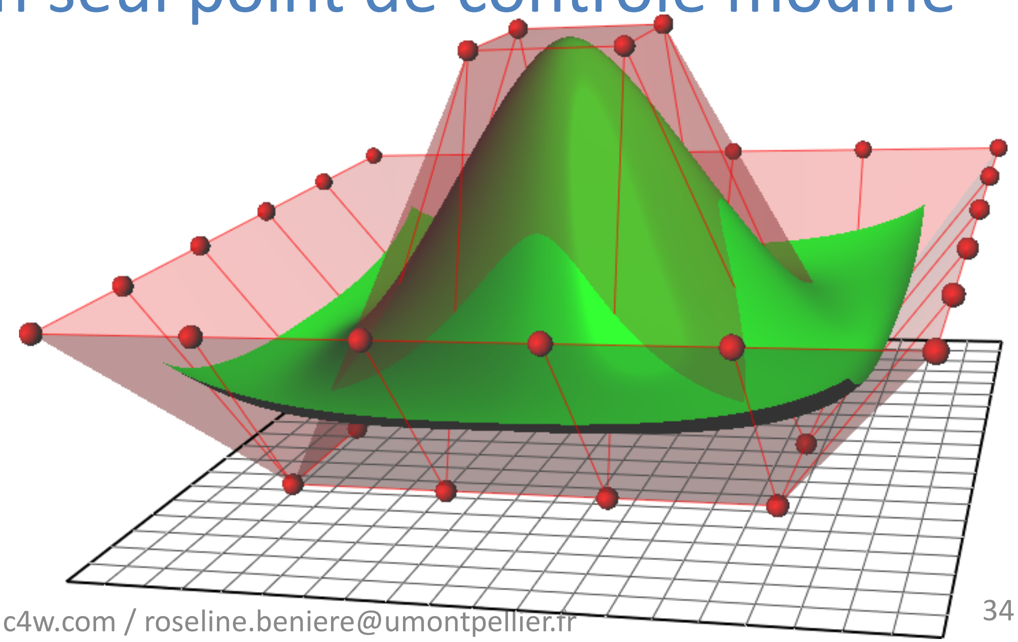


Carreaux surfaciques

- **Carreaux Béziers ➡ inconvénients**
  - les mêmes que pour les courbes,
  - les degrés des courbes sont liés aux nombres de points de contrôles,
  - le déplacement d’un seul point de contrôle modifie toute la surface.

Carreaux surfaciques

- Carreaux Béziers ➡ inconvénients
  - ➢ les mêmes que pour les courbes,
  - ➢ les degrés des courbes sont liés aux nombres de points de contrôles,
  - ➢ le déplacement d'un seul point de contrôle modifie toute la surface.



- ➢ même solution

➡ **surfaces B-Splines**

Carreaux surfaciques

- **Construction par l'algorithme De Casteljau :**
  - pour une courbe de Bézier

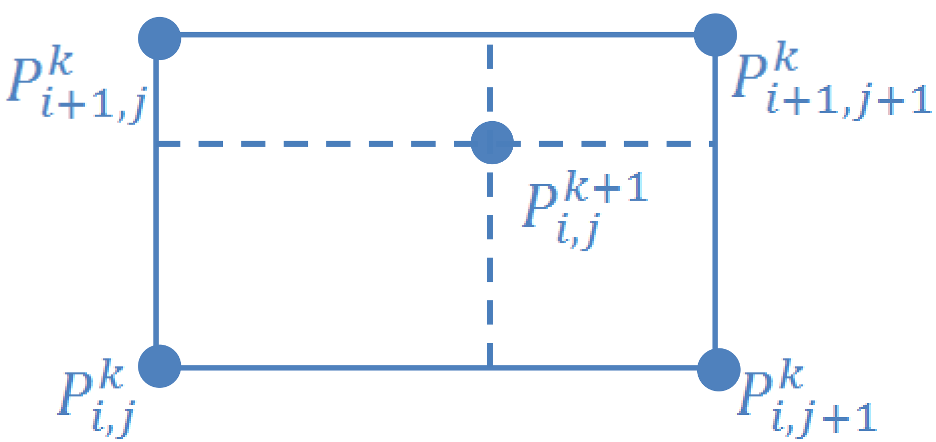
$$\boxed{P_i^{k+1} = (1\text{-}u)P_i^k + uP_{i+1}^k}$$

Carreaux surfaciques

- **Construction par l'algorithme De Casteljau :**
  - pour une courbe de Bézier

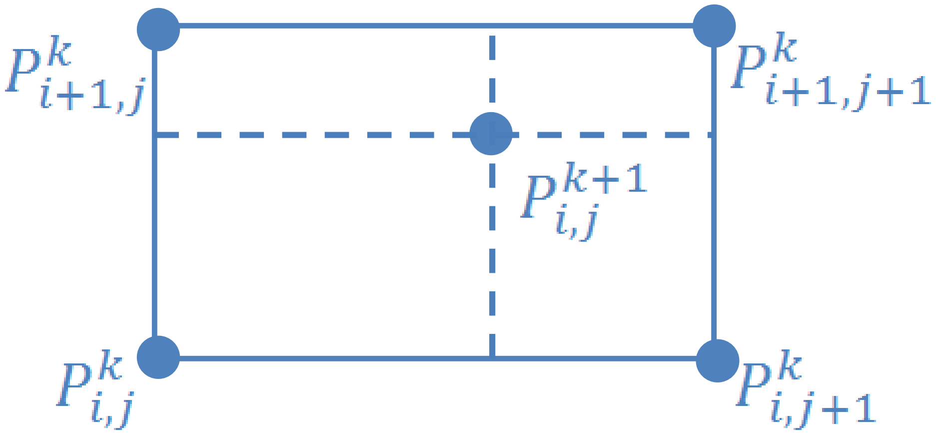
$$\boxed{P_i^{k+1} = (1\text{-}u)P_i^k + uP_{i+1}^k}$$

  - pour une surface de Bézier, on utilise les 4 points d'un carreau pour obtenir le point à l'itération suivante.



Carreaux surfaciques

- **Construction par l'algorithme De Casteljau :**
  - ➢ on commence par calculer 2 points selon *u*

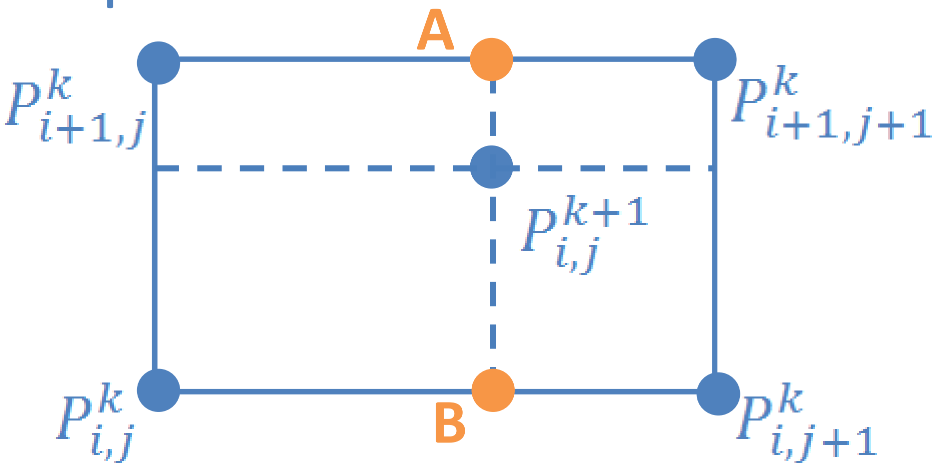


Carreaux surfaciques

- **Construction par l'algorithme De Casteljau :**
  - on commence par calculer 2 points selon *u*

$$A = (1-u)P^k_{i,j+1} + uP^k_{i+1,j+1}$$

$$B = (1-u)P^k_{i,j} + uP^k_{i+1,j}$$

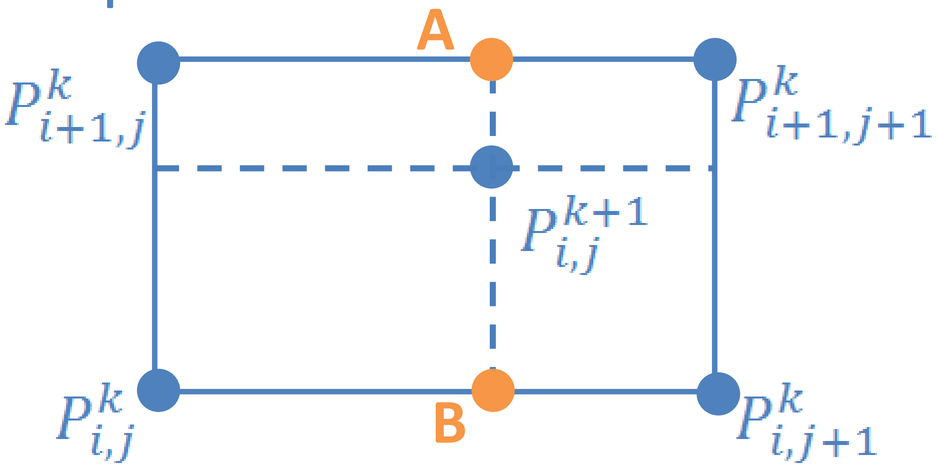


Carreaux surfaciques

- **Construction par l'algorithme De Casteljau :**
  - on commence par calculer 2 points selon *u*

$$A = (1-u)P^k_{i,j+1} + uP^k_{i+1,j+1}$$

$$B = (1-u)P^k_{i,j} + uP^k_{i+1,j}$$

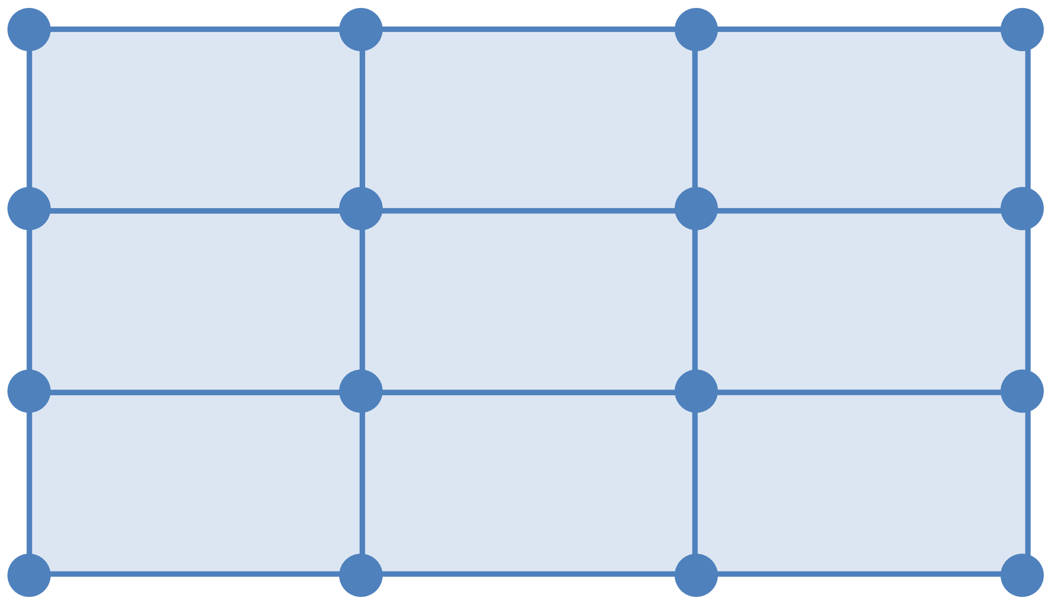


  - puis on obtient le point de l'itération, en utilisant *v*

$$P^{k+1}_{i,j} = (1-v)B + vA$$

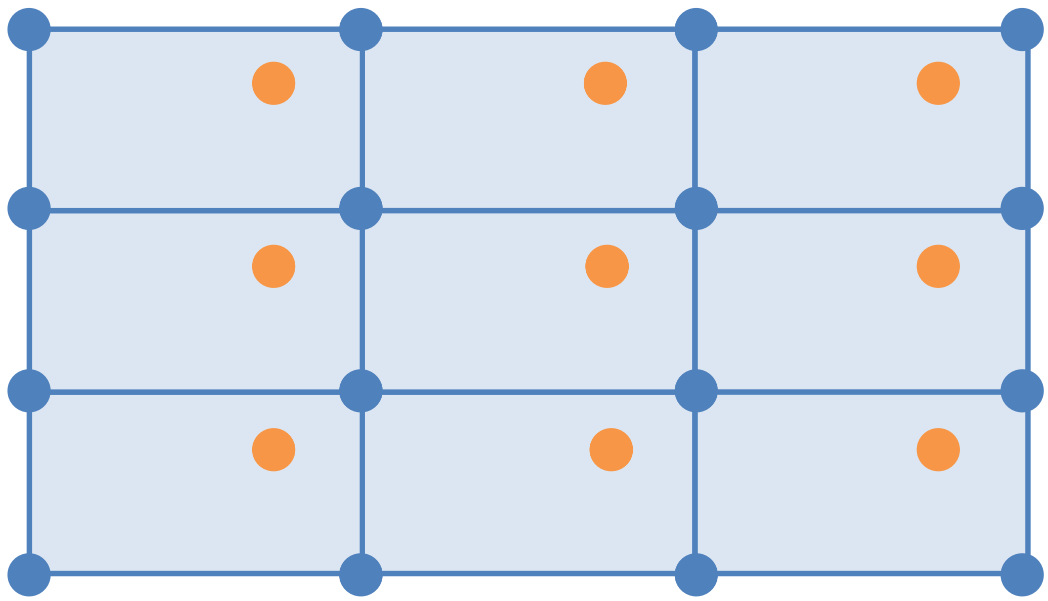
Carreaux surfaciques

- **Construction par l'algorithme De Casteljau :**
  - calcul récursive de la position du plan



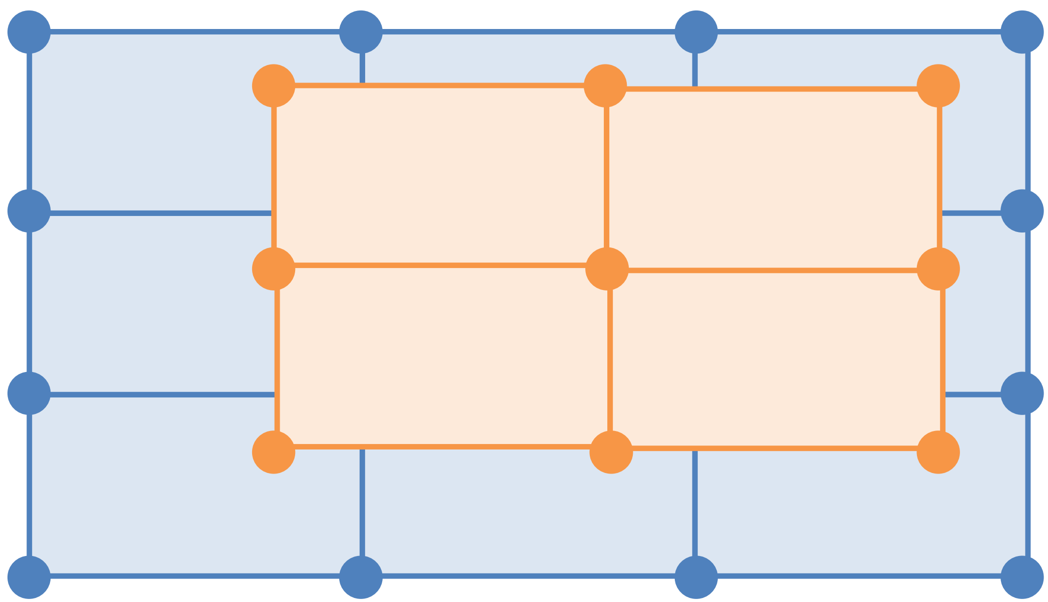
Carreaux surfaciques

- **Construction par l'algorithme De Casteljau :**
  - ➢ calcul récursive de la position du plan



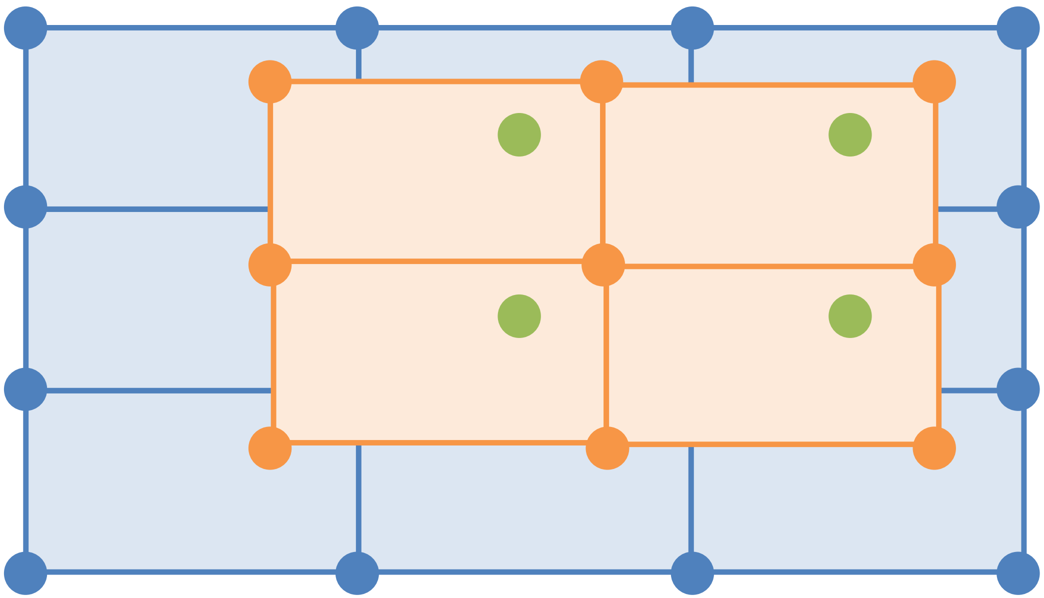
Carreaux surfaciques

- **Construction par l'algorithme De Casteljau :**
  - ➢ calcul récursive de la position du plan



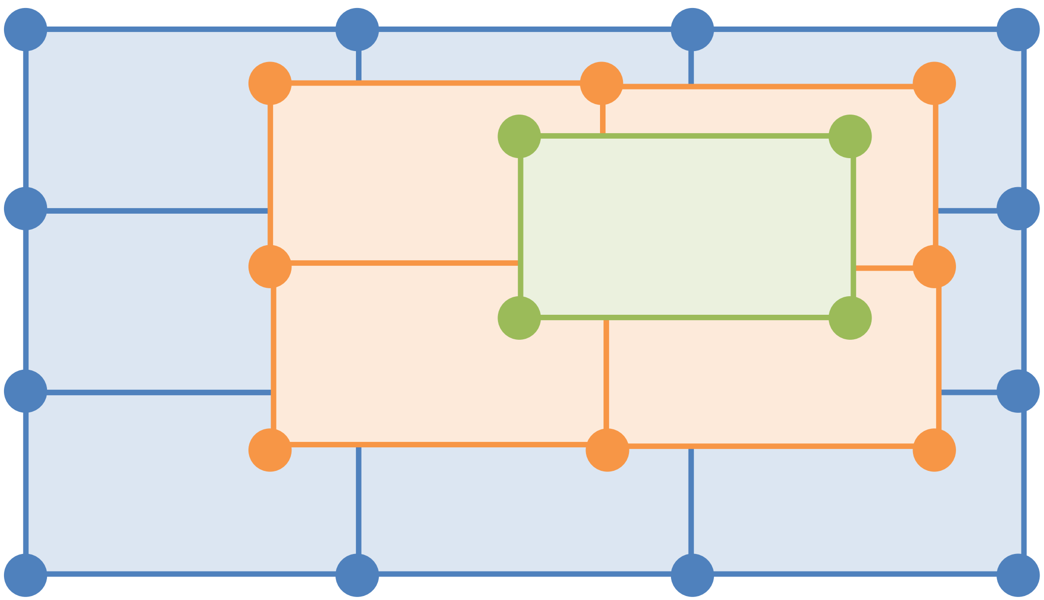
Carreaux surfaciques

- **Construction par l'algorithme De Casteljau :**
  - ➢calcul récursive de la position du plan



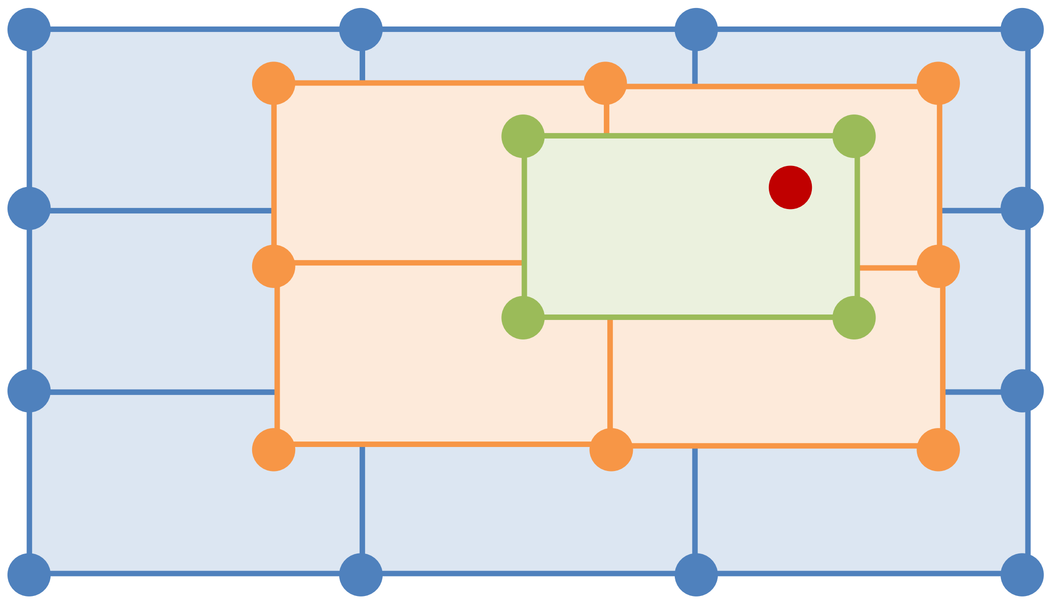
Carreaux surfaciques

- **Construction par l'algorithme De Casteljau :**
  - ➢ calcul récursive de la position du plan



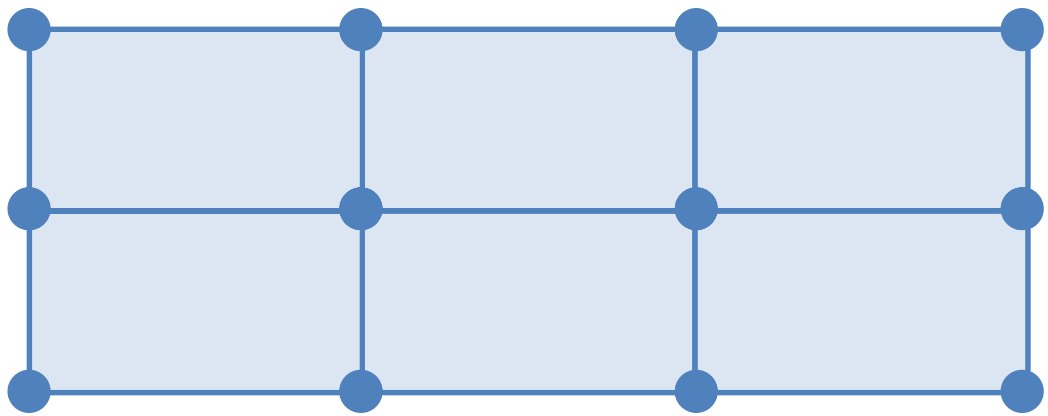
Carreaux surfaciques

- **Construction par l'algorithme De Casteljau :**
  - ➢calcul récursive de la position du plan



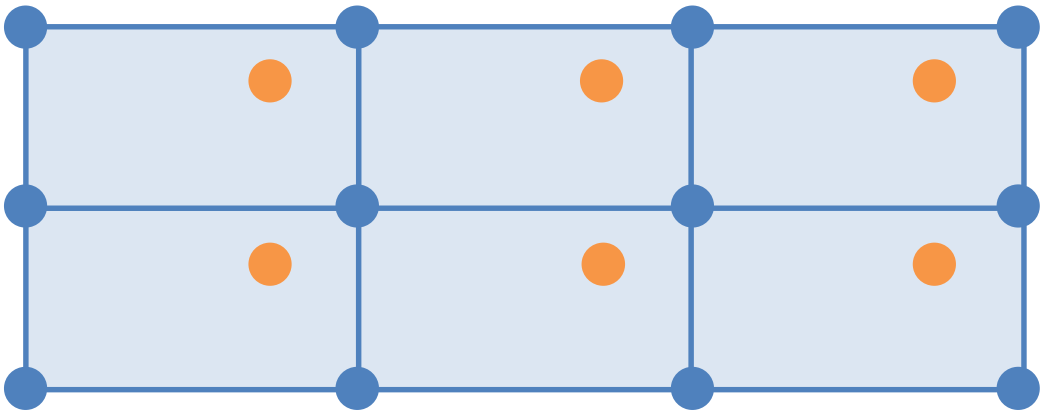
Carreaux surfaciques

- **Construction par l'algorithme De Casteljau :**
  - Attention si le degré est différent en *u* et en *v* :



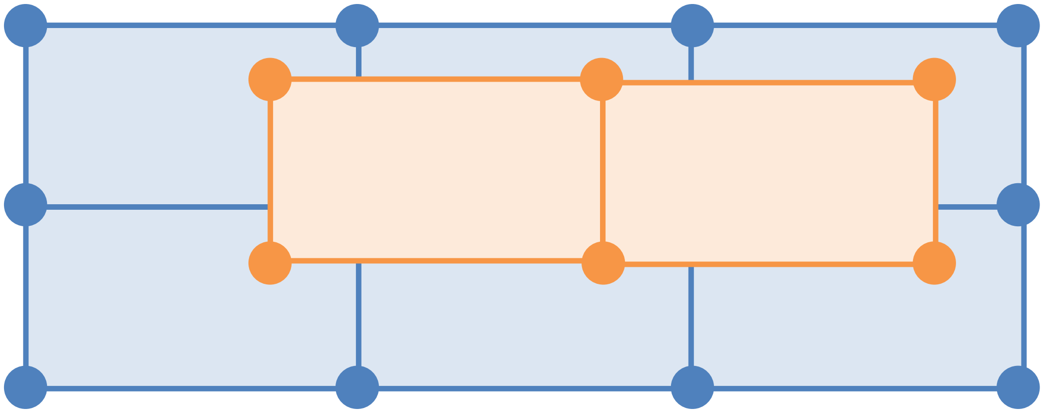
Carreaux surfaciques

- **Construction par l'algorithme De Casteljau :**
  - ➢ Attention si le degré est différent en *u* et en *v* :



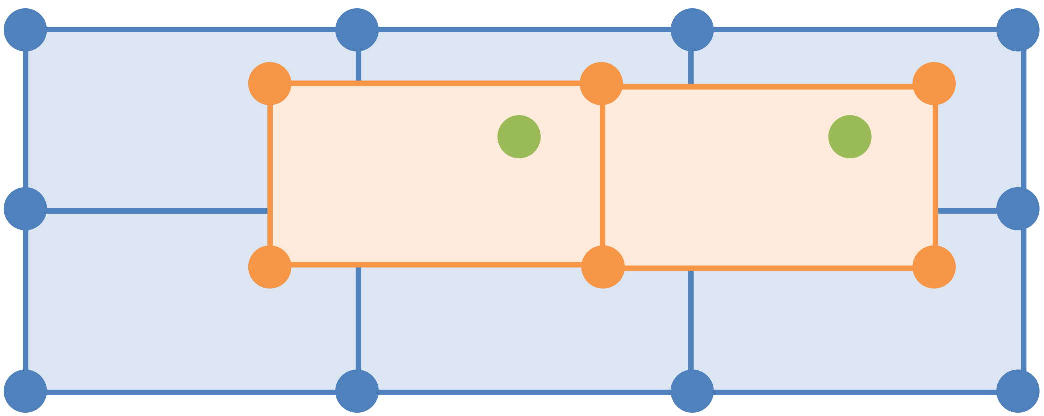
Carreaux surfaciques

- **Construction par l'algorithme De Casteljau :**
  - Attention si le degré est différent en *u* et en *v* :



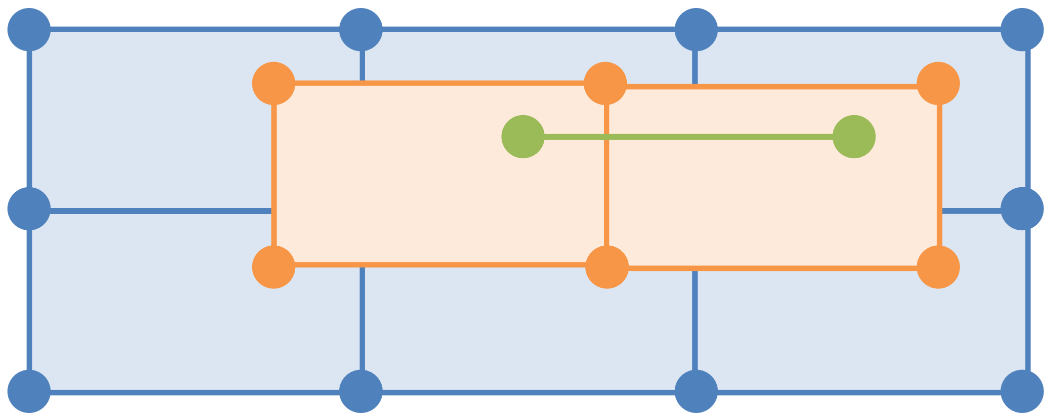
Carreaux surfaciques

- **Construction par l'algorithme De Casteljau :**
  - ➢ Attention si le degré est différent en *u* et en *v* :



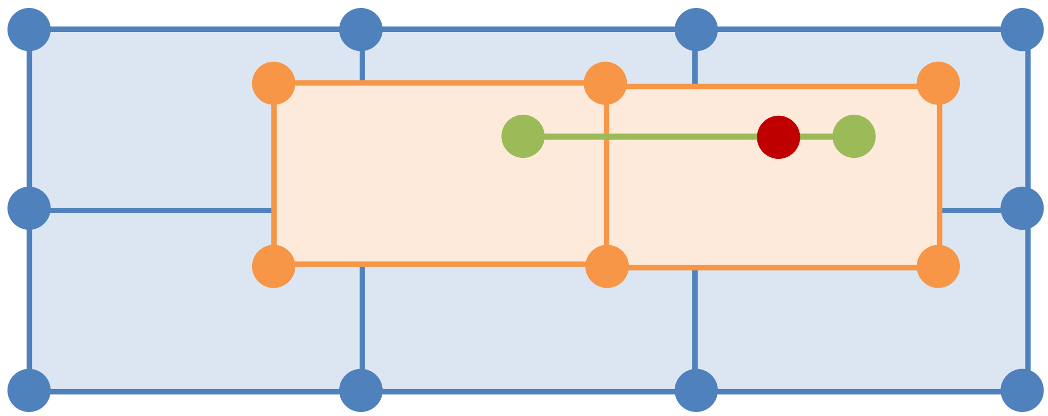
Carreaux surfaciques

- **Construction par l'algorithme De Casteljau :**
  - ➢ Attention si le degré est différent en *u* et en *v* :



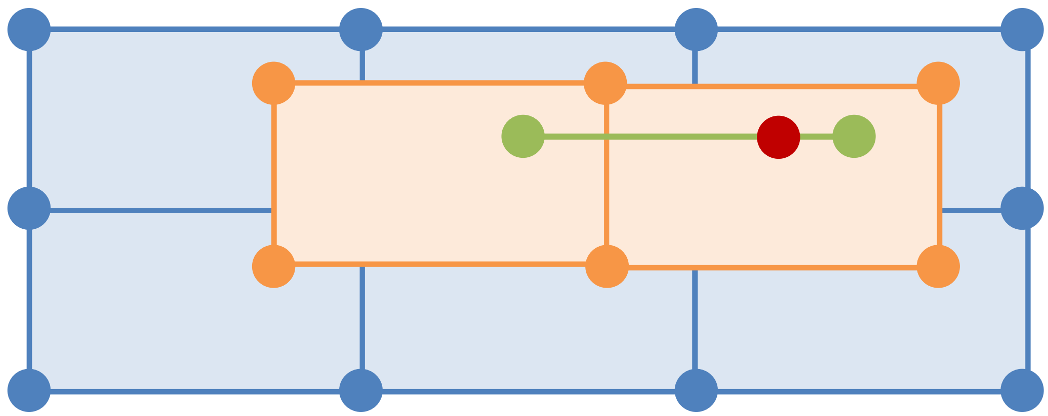
Carreaux surfaciques

- **Construction par l'algorithme De Casteljau :**
  - ➢ Attention si le degré est différent en *u* et en *v* :



Carreaux surfaciques

- **Construction par l'algorithme De Casteljau :**
  - Attention si le degré est différent en *u* et en *v* :



  - m différent de n, il y aura :
    - **|m-n|** itérations de l'algorithme De Casteljau **surface**
    - **Max(m,n) - |m-n|** itérations de De Casteljau **courbe**

Plan

- Introduction
- Surfaces balayées
- Carreaux surfaciques
- Visualisation OpenGL

Visualisation OpenGL

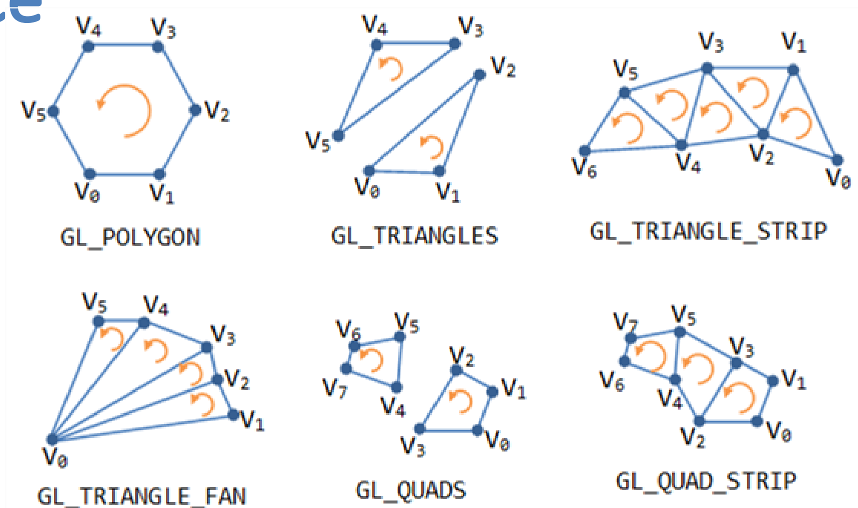
- **On ne peut pas donner à OpenGL une surface continue :**
  - calculer un ensemble de points
  - définir autant de *u* et de *v* que l'on veut de point sur la surface
  - plus, il y aura de points plus la surface apparaitra comme lisse.

Visualisation OpenGL

- Pour tracer une surface :
  - soit par un ensemble de courbes : les iso-paramétriques (voir le cours sur les courbes paramétriques)

Visualisation OpenGL

- Pour tracer une surface :
  - ➢ soit par un ensemble de courbes : les iso-paramétriques (voir le cours sur les courbes paramétriques)
  - ➢ soit en définissant un ensemble de facette correspondant à la surface
    - Type de face :
      - ○ triangles : *GL_TRIANGLES*
      - ○ quadrangles : *GL_QUADS*
      - ○ polygones : *GL_POLYGON*
      - ○ …



Conclusion

- **Surface paramétrique :**
  - défini par un ensemble de fonction
  - 2 paramètres pour décrire 2 ``déplacements’’ sur les courbes définissant la surface

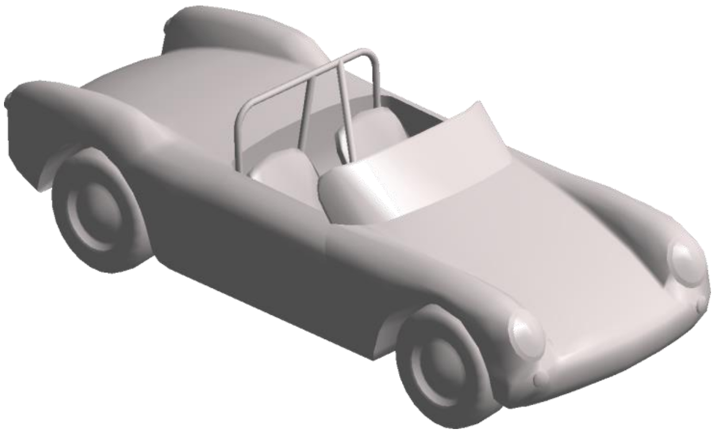
Conclusion

- **Surface paramétrique :**
  - ➢ défini par un ensemble de fonction
  - ➢ 2 paramètres pour décrire 2 ``déplacements’’ sur les courbes définissant la surface
- **Plusieurs types de surfaces :**
  - ➢ plan / sphère …
  - ➢ surfaces balayées :
    - ○ surfaces de révolution,
    - ○ surfaces cylindriques,
    - ○ …
  - ➢ surfaces de Bézier

FIN

Représentation surfacique

lundi 06/02

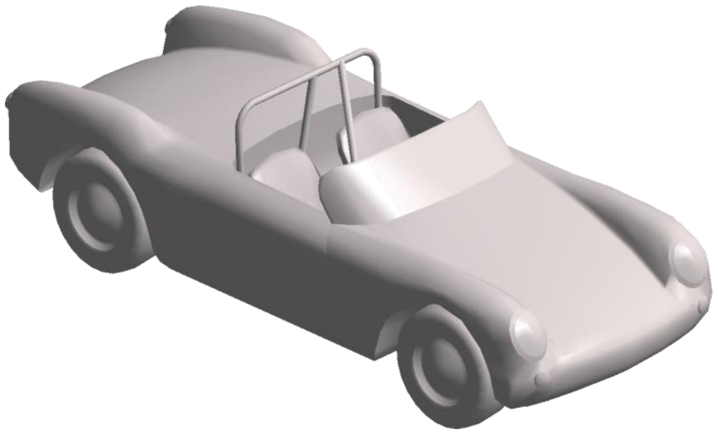


Pour récupérer les cours et le TD/TP:
http://www.lirmm.fr/~beniere/Enseignements.php

FIN

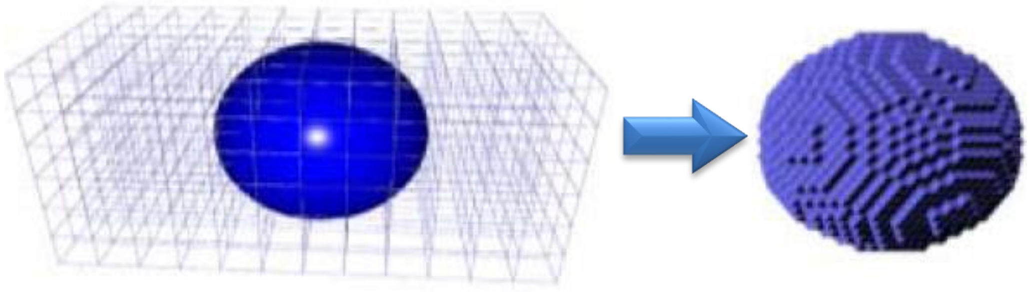
Représentation surfacique

lundi 06/02



Modèles volumiques

lundi 20/02



Pour récupérer les cours et le TD/TP:
http://www.lirmm.fr/~beniere/Enseignements.php

Sources

- **Cours utilisés pour ce support :**
  - Gilles Gesquière (Gamagora Lyon)
  - Loïc Barthe (IRIT-UPS Toulouse)