

Plan du cours théorique

Historique et mise en perspective

Partie I. La projection parallèle

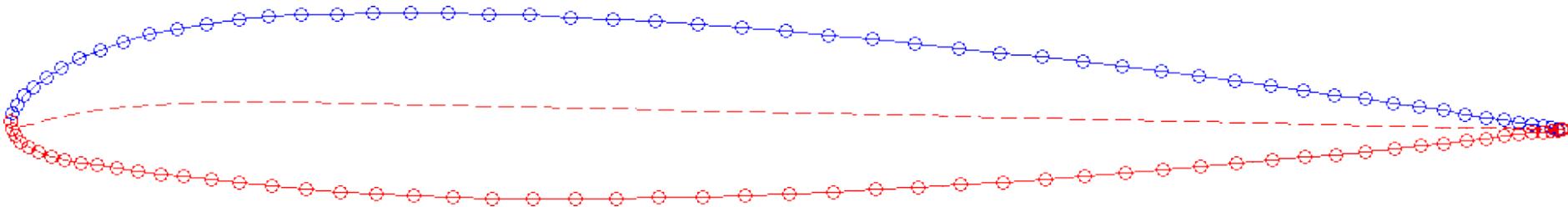
1. Le dessin multivue (dessin technique)
2. L'axonométrie

Partie II. Géométrie numérique

- 1. Courbes de Bézier**
2. Les applications affines

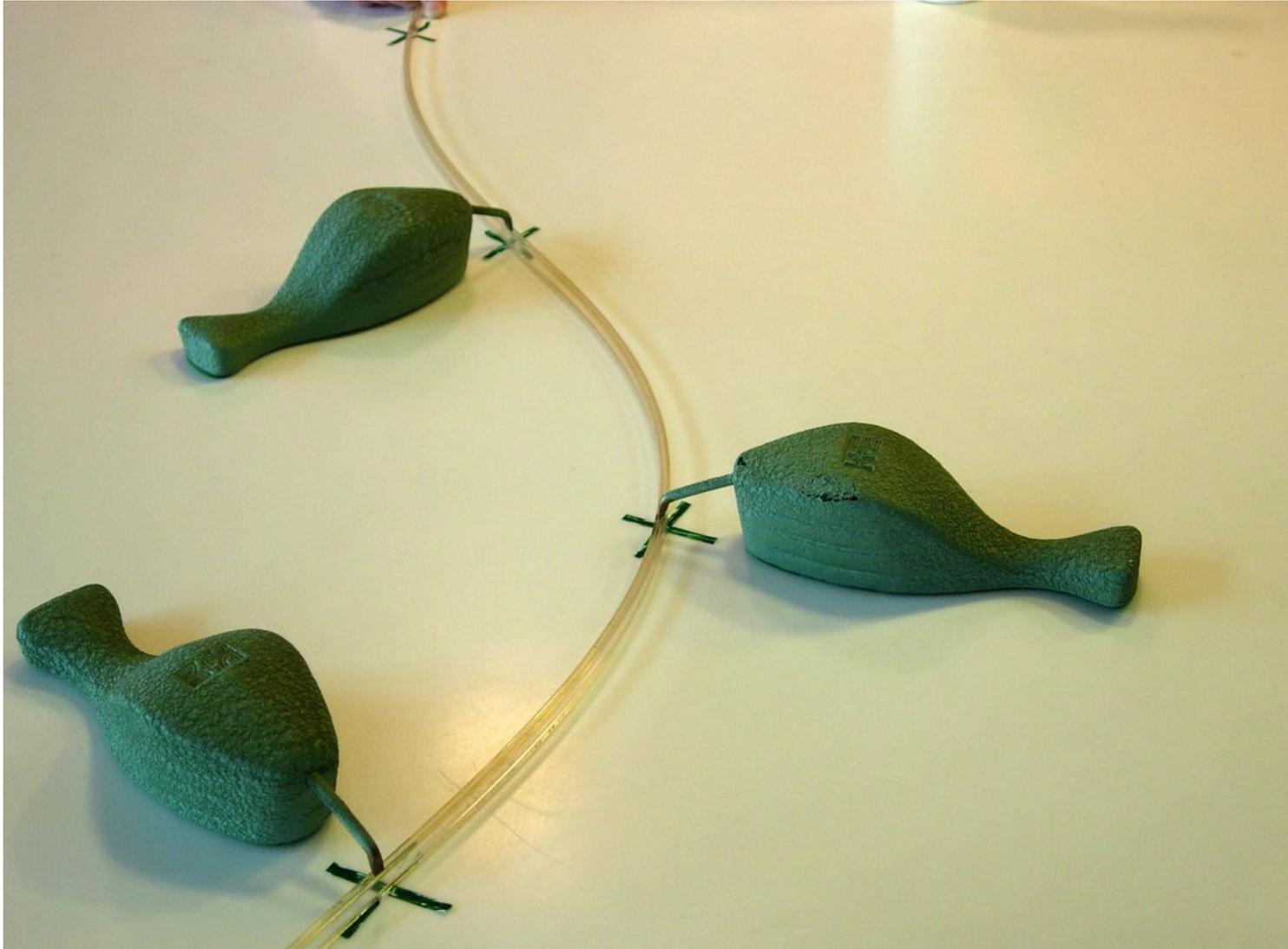
Courbes

- Représentation de courbes
 - Arcs de cercles – compas
 - Projection d'arcs de cercles – ovales → arcs de cercles
 - Autres courbes ?
 - Splines – la courbe passe par une série de points et est « lisse »
 - Construction navale et aéronautique (cambrure d'une aile d'avion p. ex.)



Profil d'aile NACA

Courbes

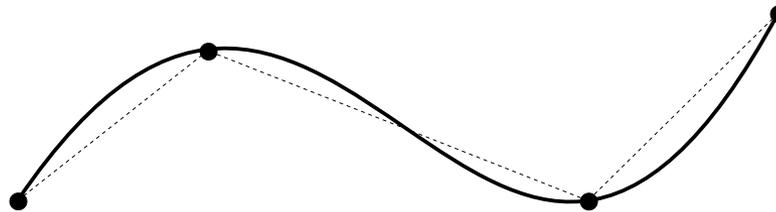


Courbes

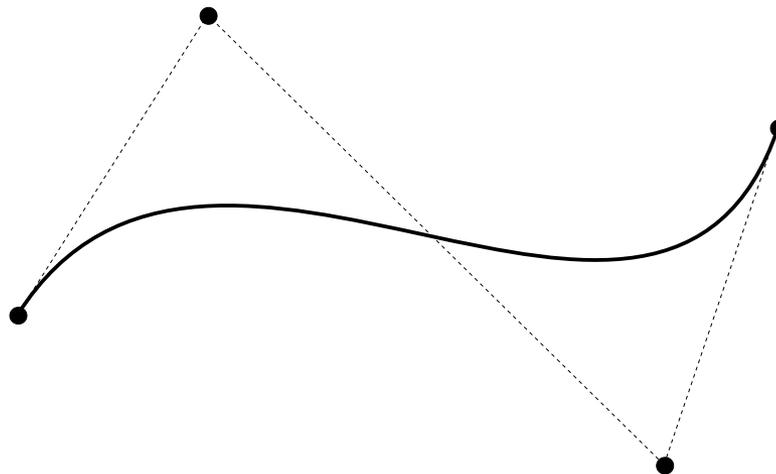


Courbes

- Courbes interpolantes (Splines, Lagrange etc...)

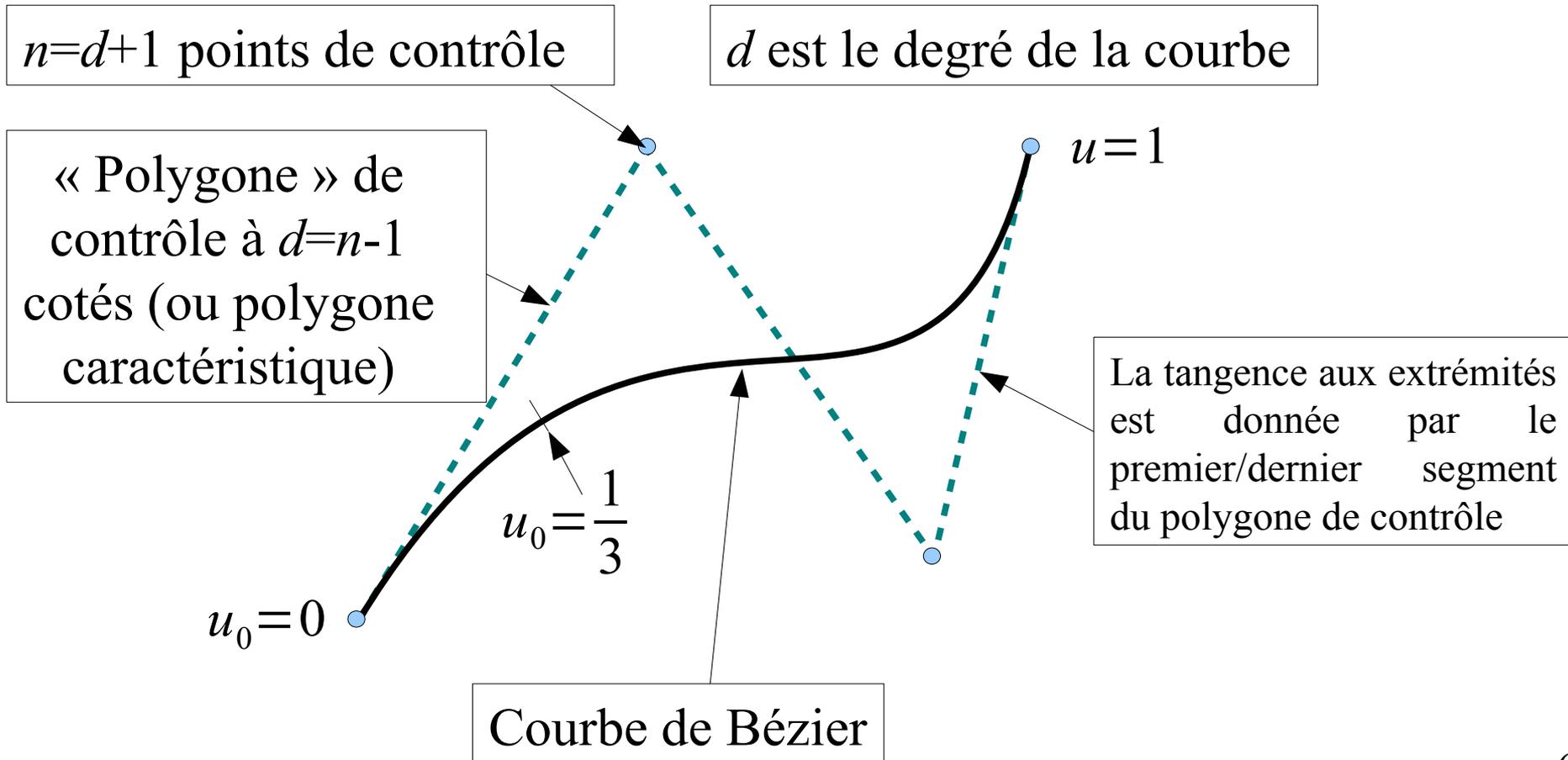


- Courbes approximantes (Bézier et d'autres)



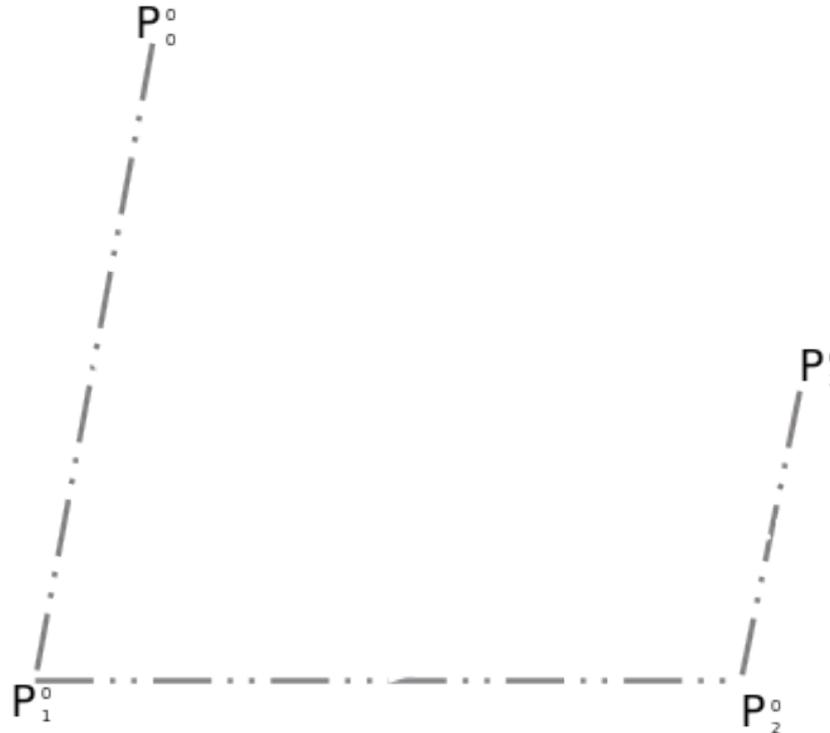
Courbes de Bézier

- Éléments d'une courbe de Bézier :



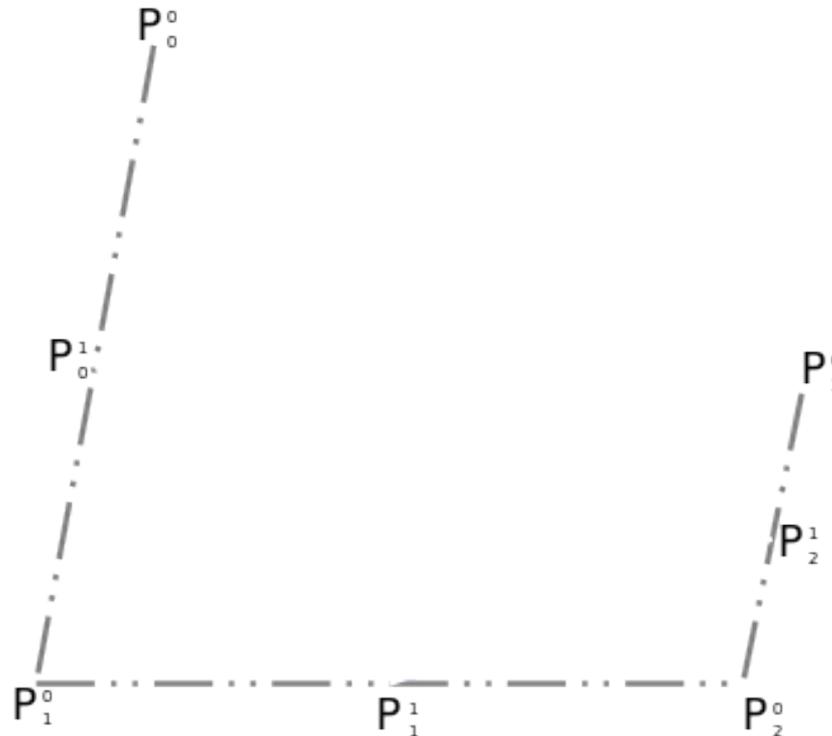
Courbes de Bézier

- Construction d'un point de la courbe de Bézier pour $u=1/2$: Algorithme de De Casteljau pour $n=4, d=3$



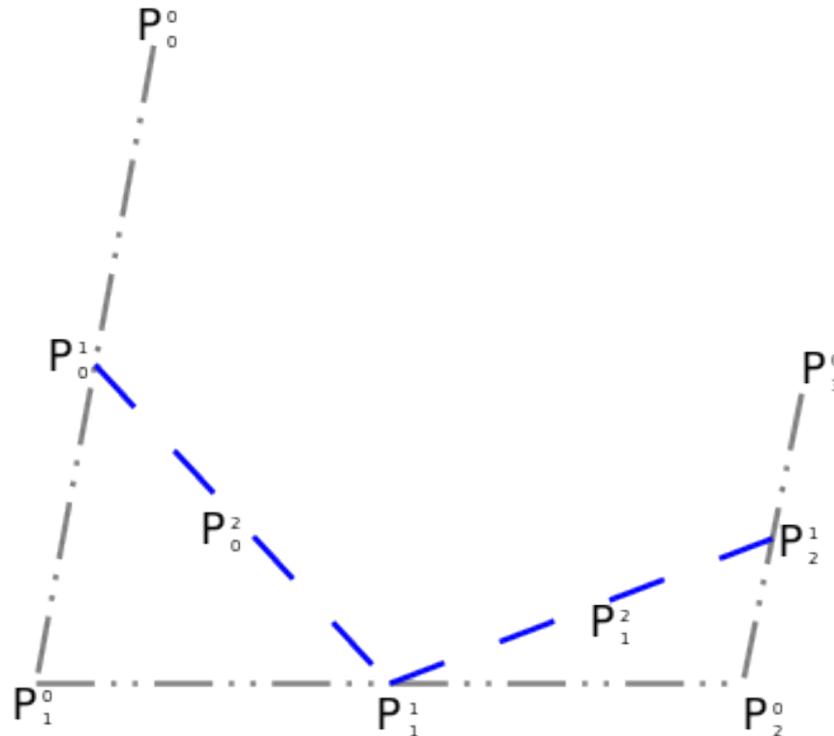
Courbes de Bézier

- Sur les segments du polygone, positionner les points milieu



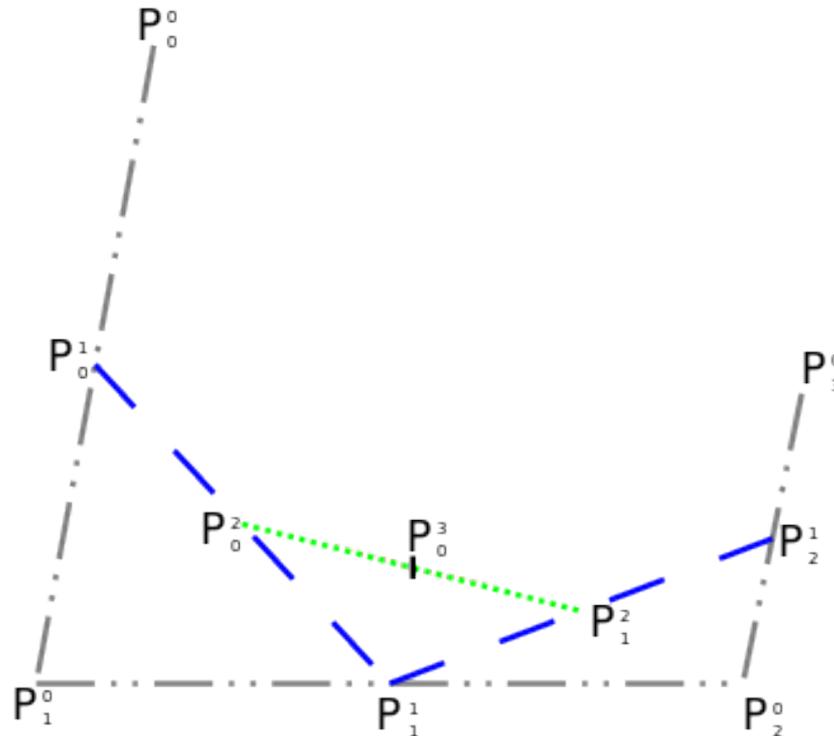
Courbes de Bézier

- Relier les points milieu et positionner les nouveaux points milieu



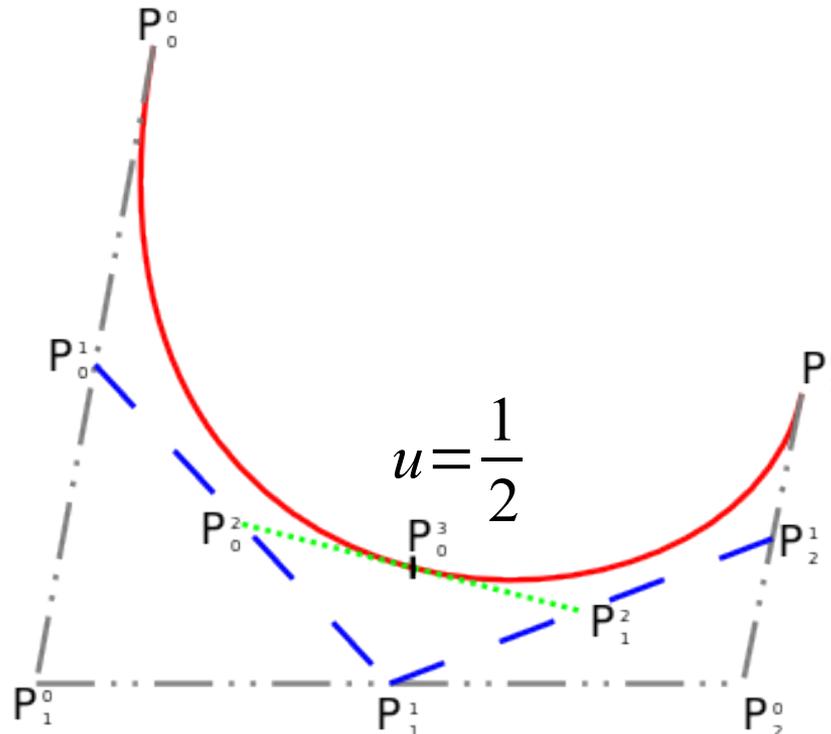
Courbes de Bézier

- Ceci, autant de fois que nécessaire...



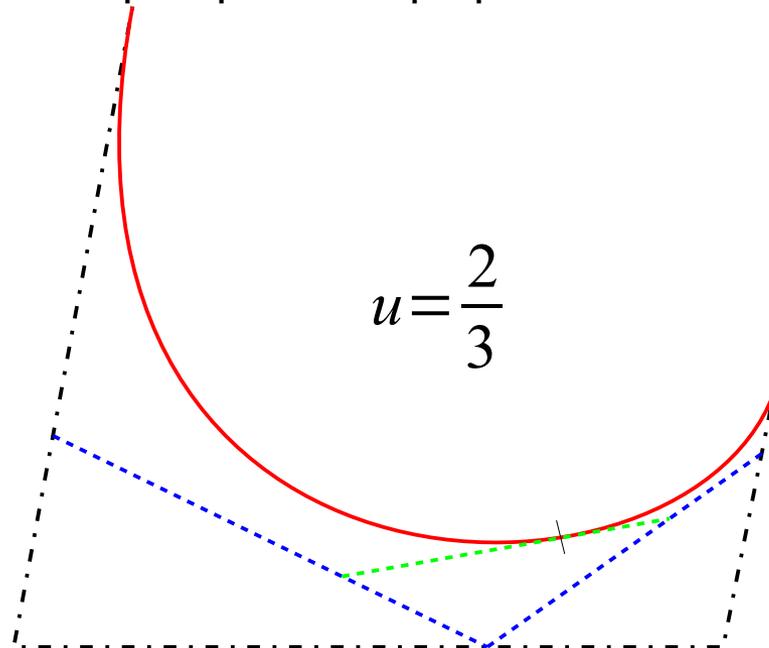
Courbes de Bézier

- Le dernier point milieu donne un point de la courbe (correspondant à $u=1/2$)



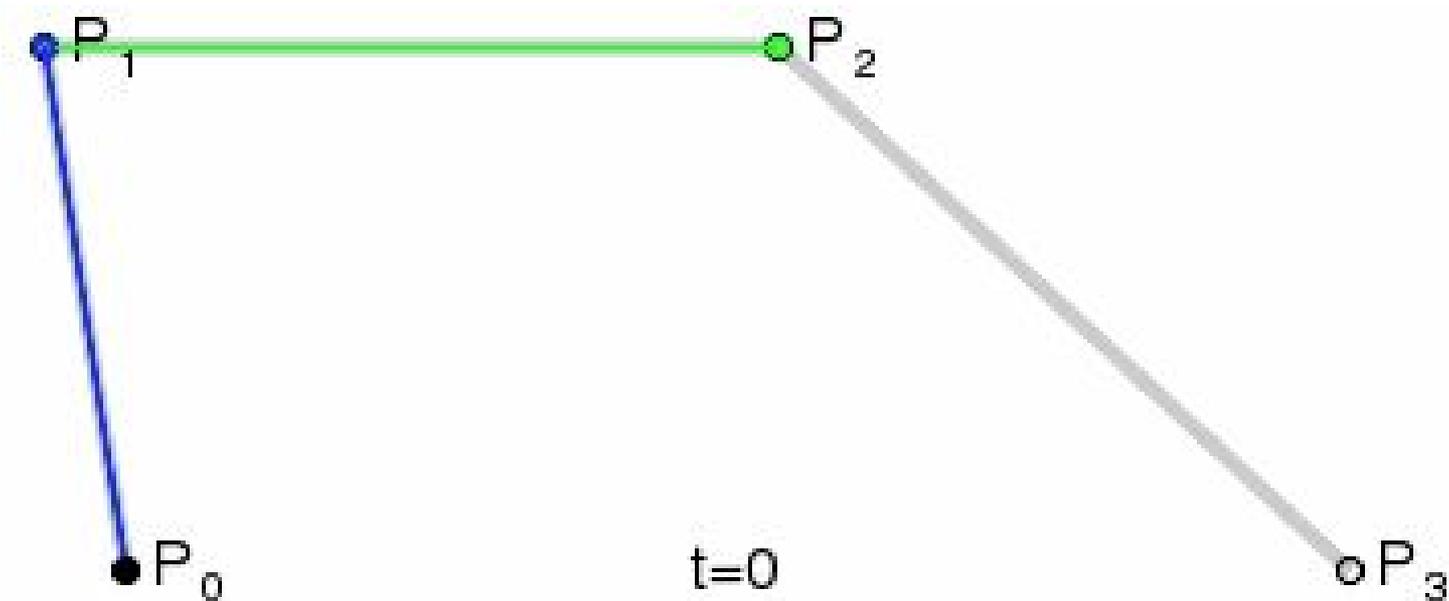
Courbes de Bézier

- La construction précédente est valable pour u quelconque (compris entre 0 et 1 !)
- Plutôt que de prendre le point milieu, on prend le point tel que : $P_j^{i+1} = (1-u)P_j^i + uP_{j+1}^i, 0 \leq u \leq 1$
 - C'est juste un peu plus compliqué à construire graphiquement



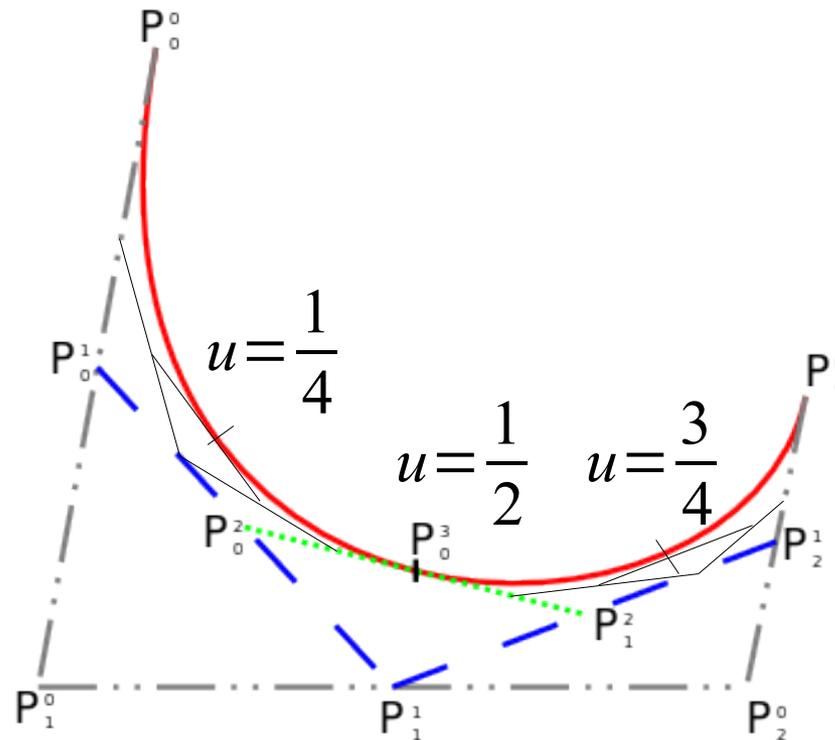
Courbes de Bézier

- En balayant u entre 0 et 1, on décrit la courbe au complet



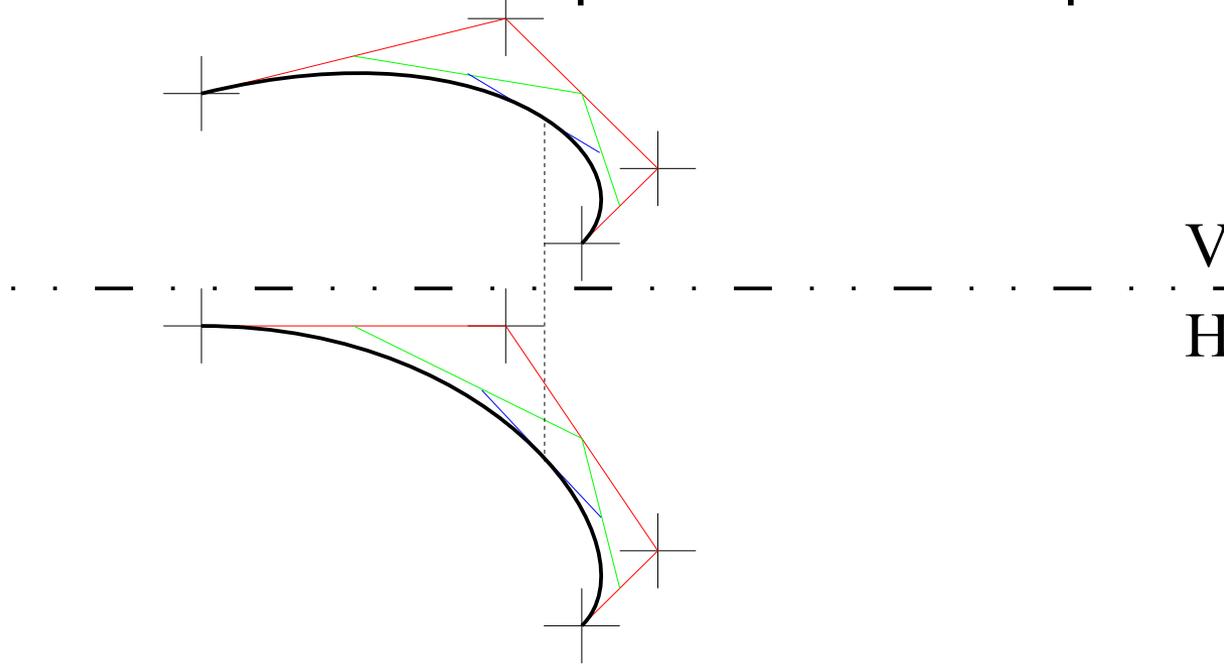
Courbes de Bézier

- Construction récursive d'autres points de la courbe de Bézier avec le polygone de contrôle



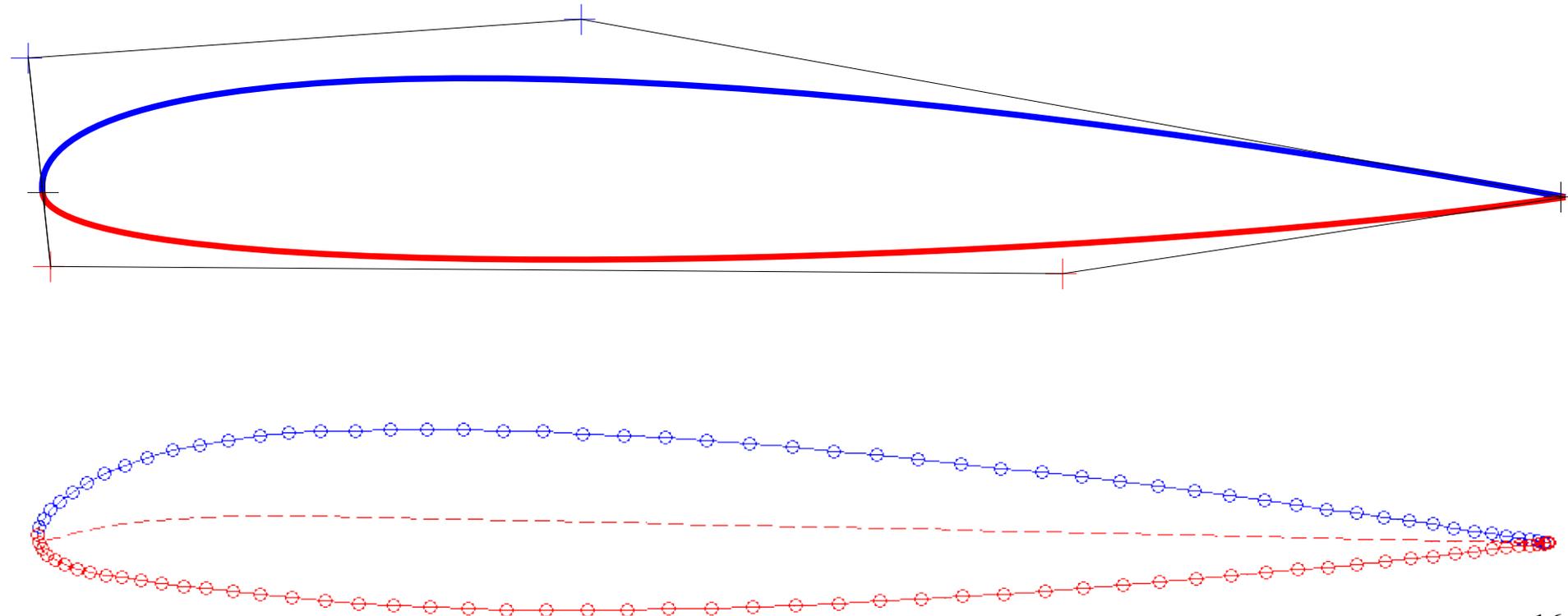
Courbes de Bézier

- Courbes tridimensionnelles?
 - La projection parallèle conserve les rapports de longueurs
 - La projection de la courbe peut être construite dans l'épure sur chacun des plans V et H indépendamment



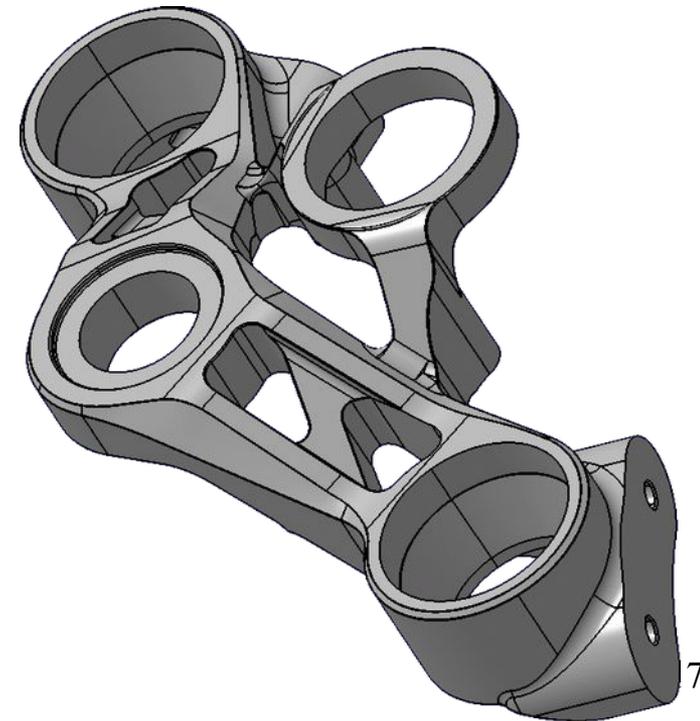
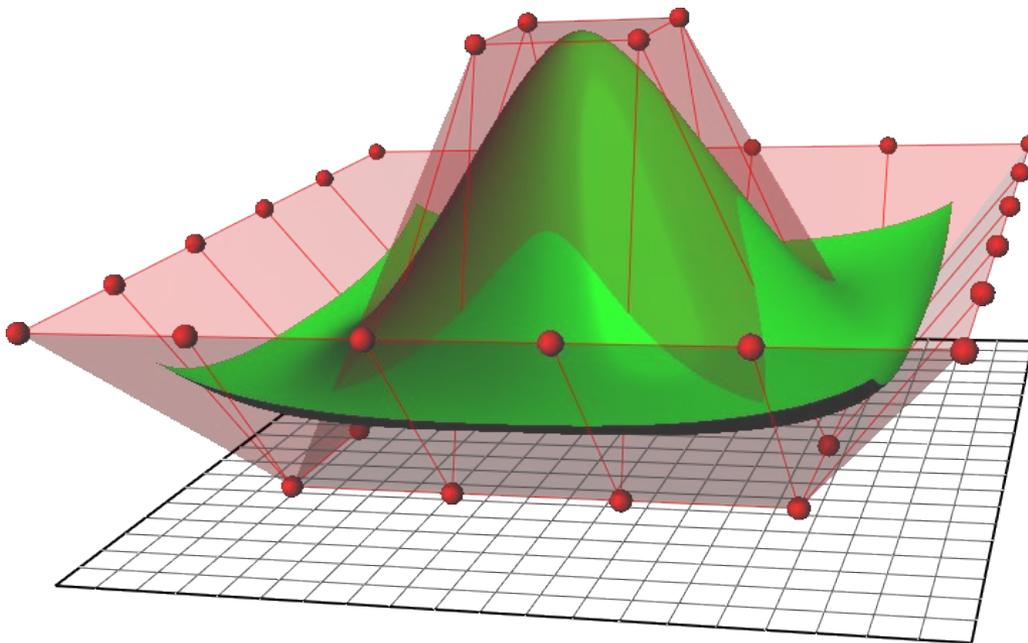
Courbes de Bézier

- Représentation d'un profil d'aile
 - Deux courbes de Bézier raboutées en tangence



Courbes de Bézier

- Surfaces de Bézier/ B-Splines / Nurbs
 - Bases de ce que l'on trouve dans les logiciels de CAO

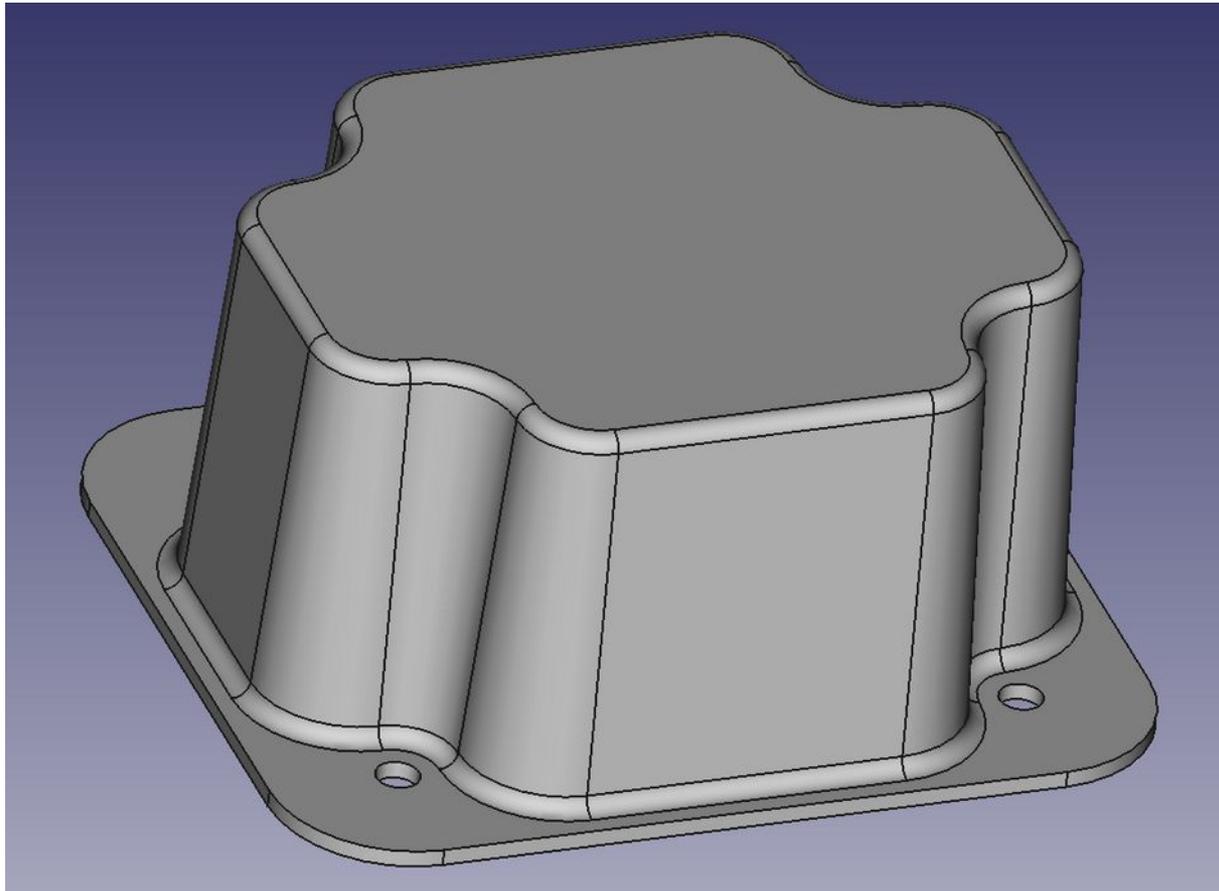


Principes de CAO

- Quelques principes courants de construction d'un modèle CAO « réutilisable »
 - Notion de modélisation paramétrique
 - Toutes les opérations sont paramétrables et donc le modèle CAO est « vivant » car ajustable à des contraintes externes
 - Un modèle complet comporte parfois des centaines de parties, toutes interdépendantes.
 - La base de la modélisation paramétrique est l'**esquisse** : dessins (en principes bidimensionnels) servant de base pour la construction d'objets.
 - On peut définir des **contraintes**, de façon à ne garder que les « degrés de liberté » pertinents.

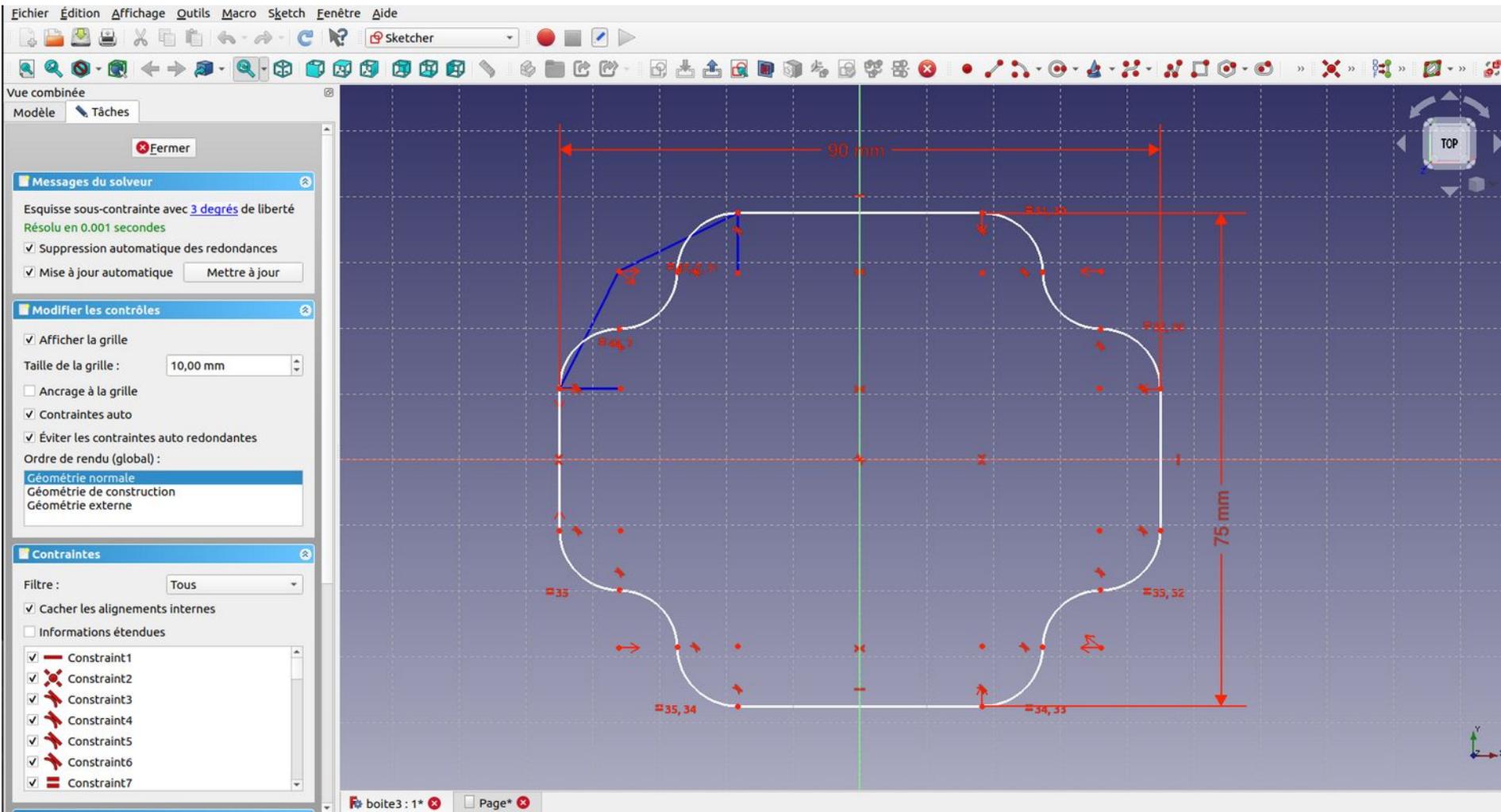
Principes de CAO

- Exemple de modèle paramétrique



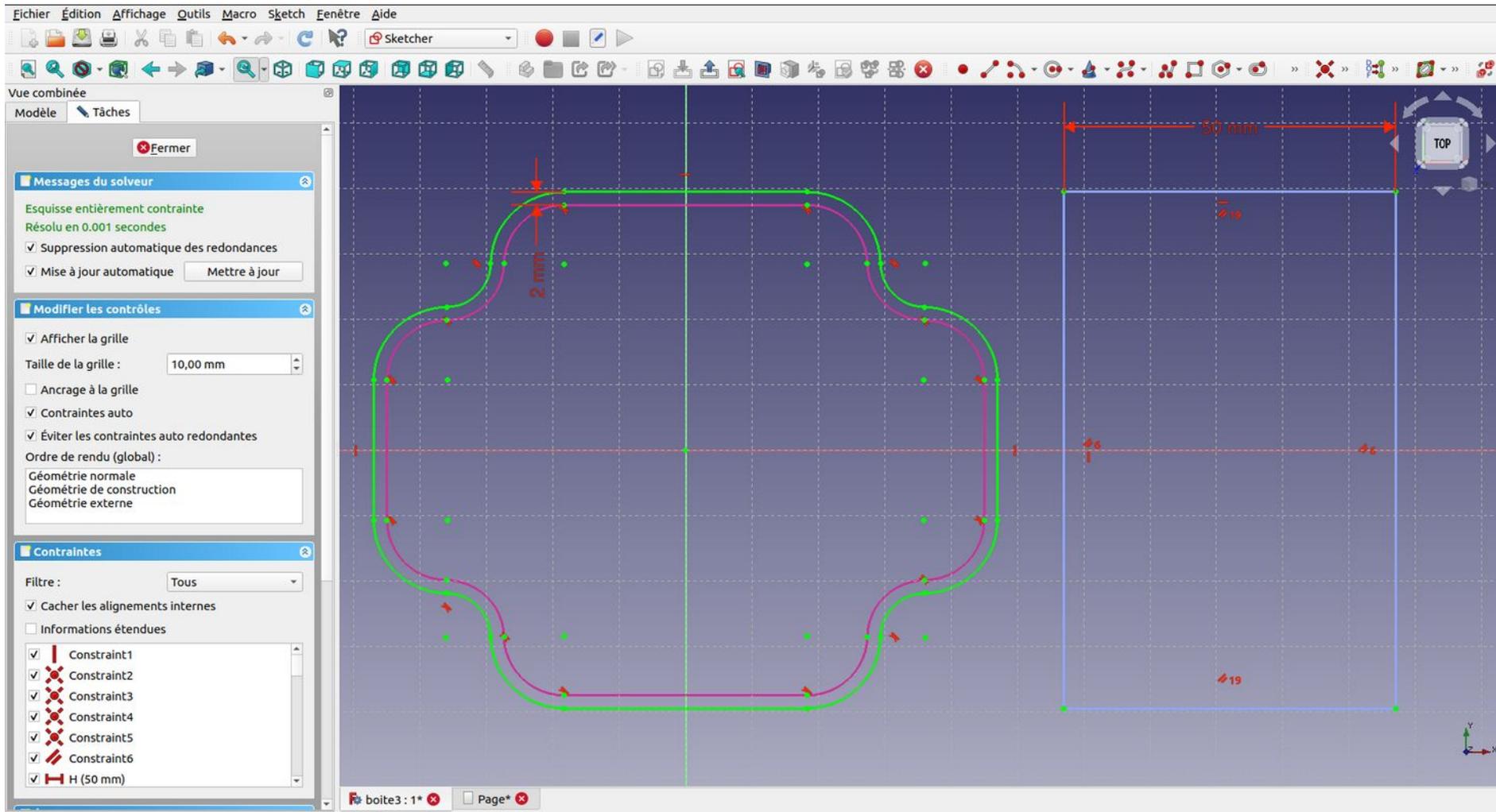
Principes de CAO

- Esquisse + contraintes ...

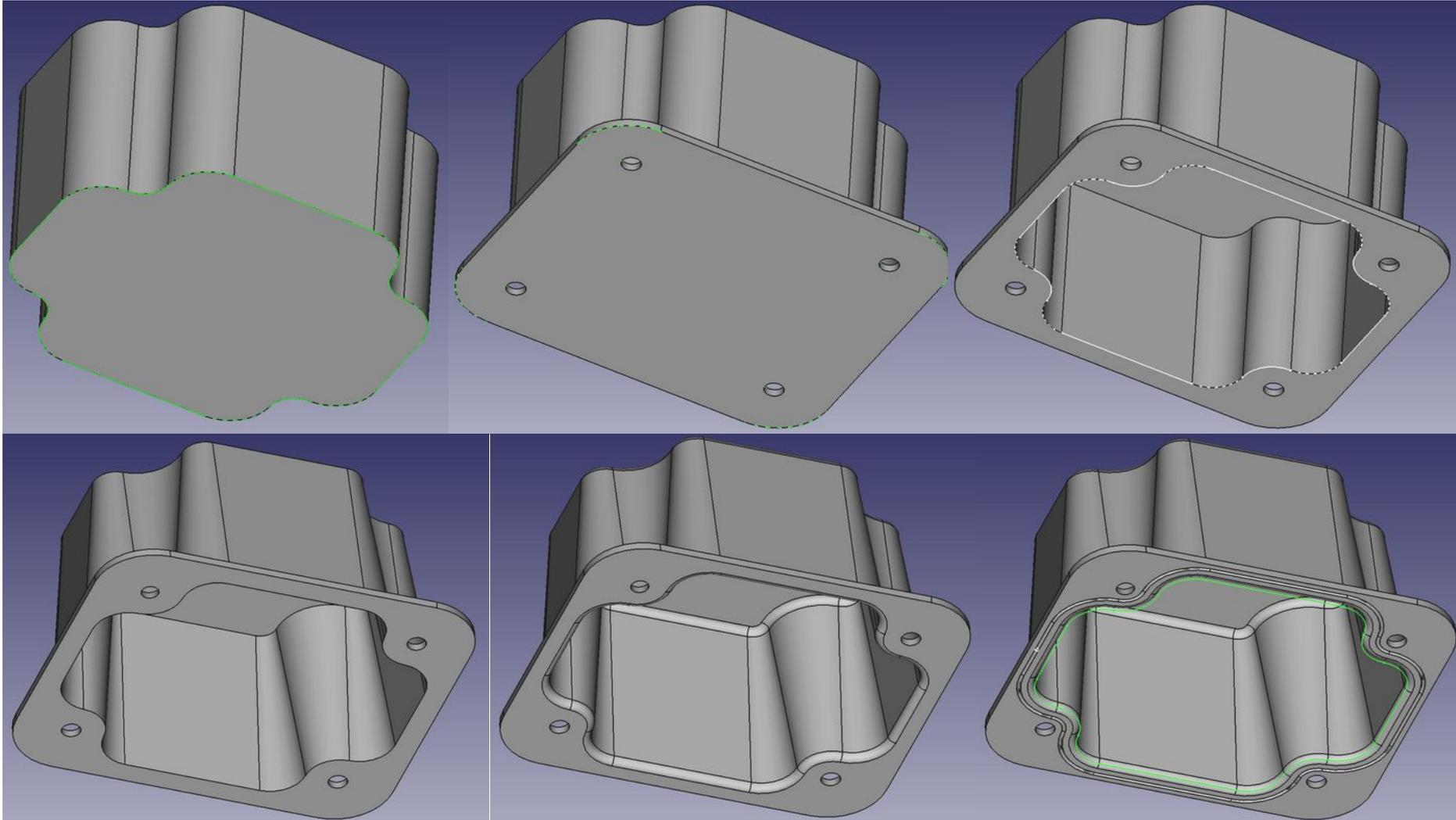


Principes de CAO

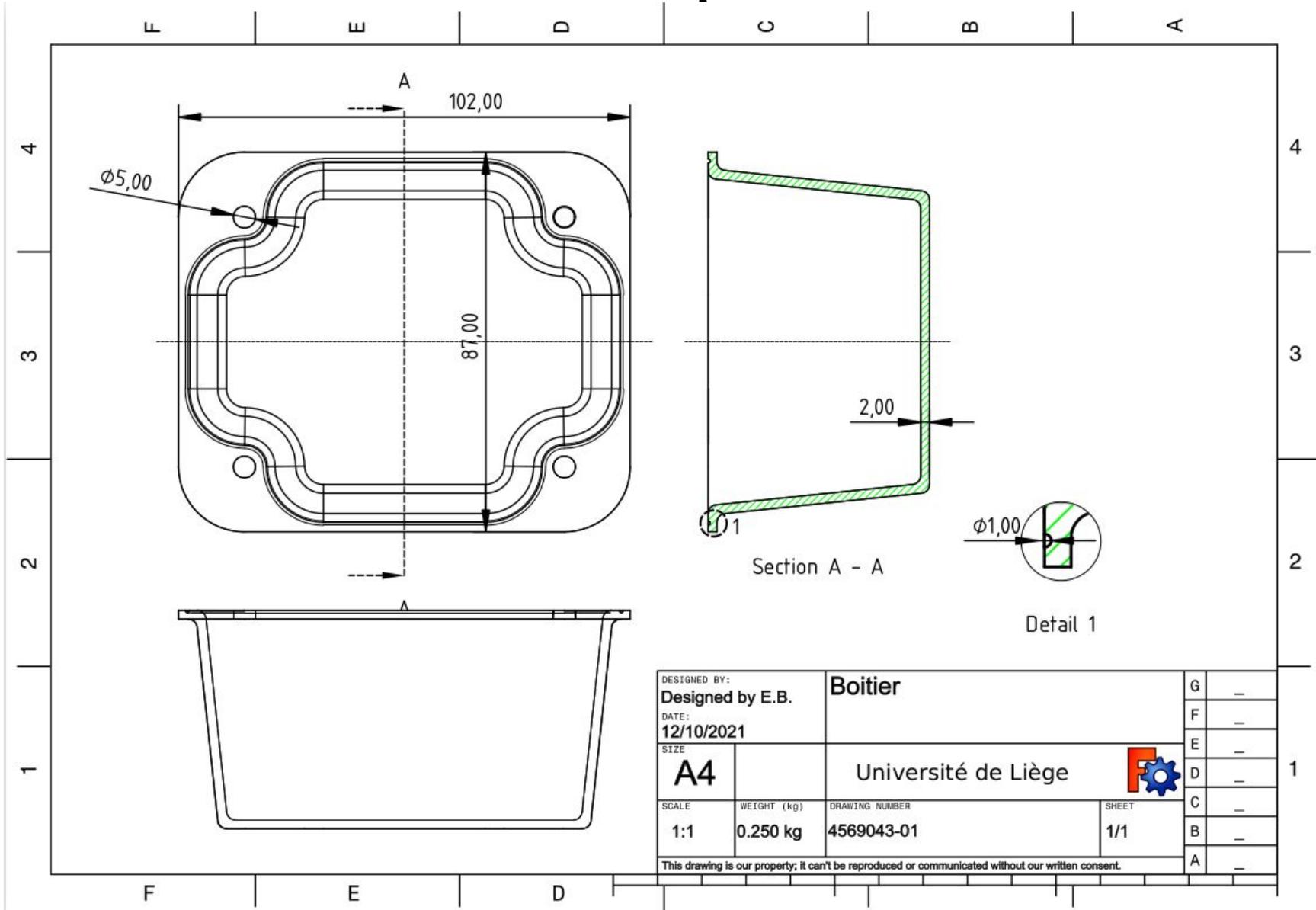
.... qui permettent de définir d'autres esquisses



Principes de CAO



Principes de CAO



DESIGNED BY: Designed by E.B.		Boitier		G	-
DATE: 12/10/2021				F	-
SIZE A4		Université de Liège		E	-
SCALE 1:1	WEIGHT (kg) 0.250 kg	DRAWING NUMBER 4569043-01	SHEET 1/1	D	-
This drawing is our property; it can't be reproduced or communicated without our written consent.				C	-
				B	-
				A	-

Plan du cours théorique

Historique et mise en perspective

Partie I. La projection parallèle

1. Le dessin multivue (dessin technique)
2. L'axonométrie

Partie II. Géométrie numérique

1. Courbes de Bézier
- 2. Les applications affines**

Géométrie numérique

- Représentation numérique des points, vecteurs
- Opérations élémentaires
 - Translations
 - Rotations
 - Mises à l'échelle
 - Cisaillements
- Traitement numérique des axonométries
- Cas des courbes de Bézier

Transformations affines

Points et vecteurs

Les points sont des éléments de l'espace euclidien tridimensionnel E^3

E^3 s'appelle l'espace affine, un point définit une position, point milieu d'une droite, centre de gravité de l'objet

Les vecteurs sont des éléments de l'espace vectoriel \mathbb{R}^3

$$P = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \quad ; \quad t = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$



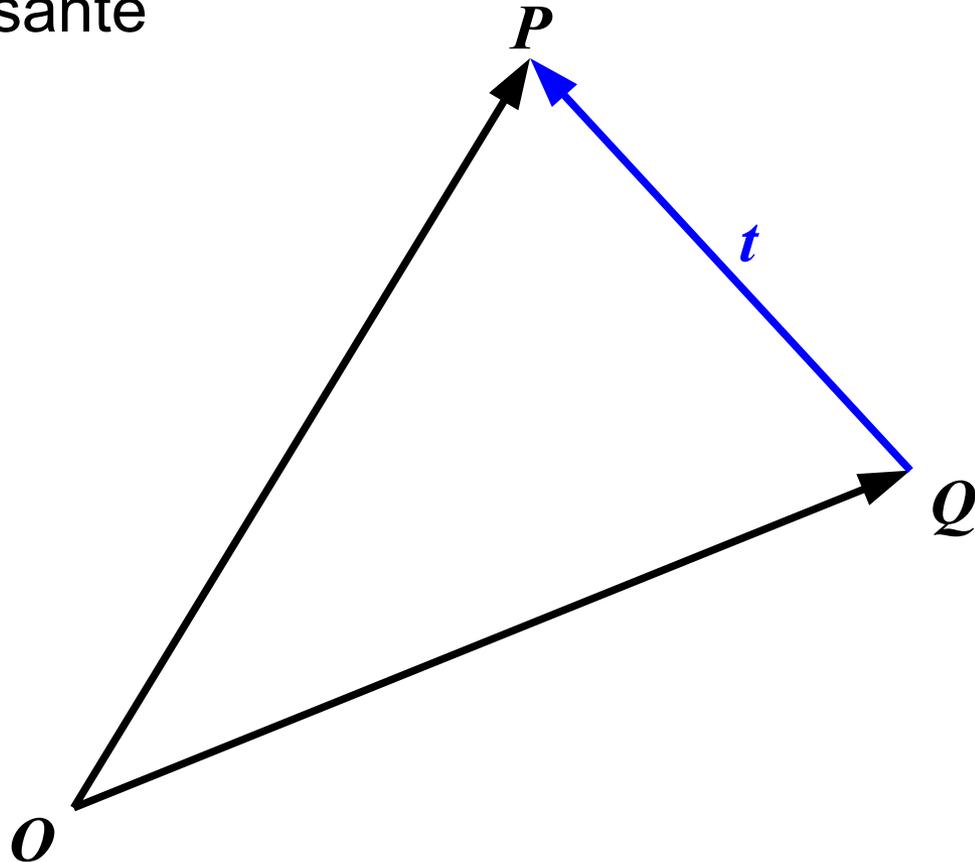
La convention utilisée dans la suite est celle des vecteurs colonne

Vecteurs

Pour tout couple de points P, Q , il existe un vecteur unique t qui pointe de Q vers P , il est calculé par leur soustraction composante par composante

$$t = P - Q$$

$$P, Q \in E^3; \quad t \in \mathbb{R}^3$$



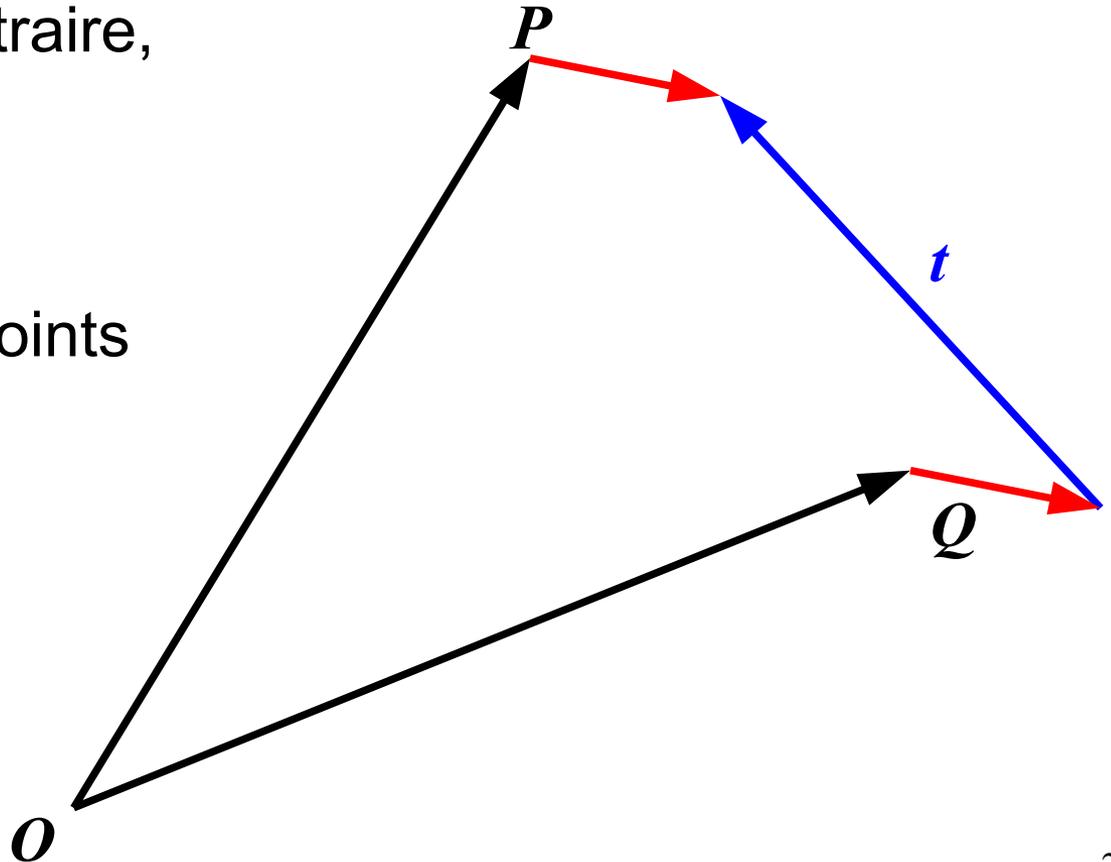
Vecteurs

Par ailleurs pour un vecteur donné t , il existe une infinité de paires de points telles que $t = P - Q$

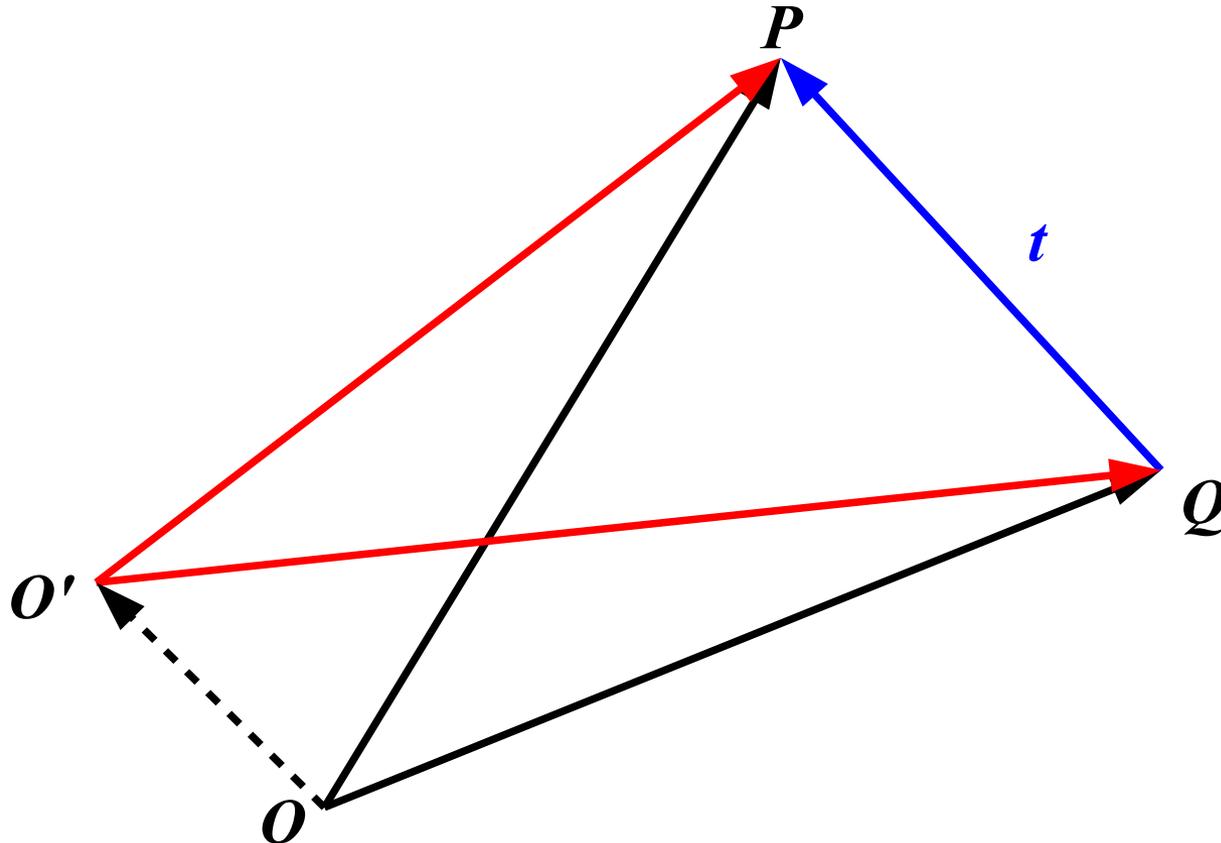
Si u est un vecteur arbitraire,

$$P + u, Q + u$$

est une autre paire de points qui satisfait la relation.



Vecteurs et Points



les **vecteurs** sont **invariants** par rapport aux translations
tandis que les points ne le sont pas

Combinaison barycentrique

Combinaison barycentrique souvent appelée combinaison affine

$$X = \sum_{i=0}^n \alpha_i P_i \quad X, P_i \in E^3, \alpha_0 + \dots + \alpha_n = 1$$

$$X = P_0 + \sum_{i=1}^n \beta_i (P_i - P_0)$$

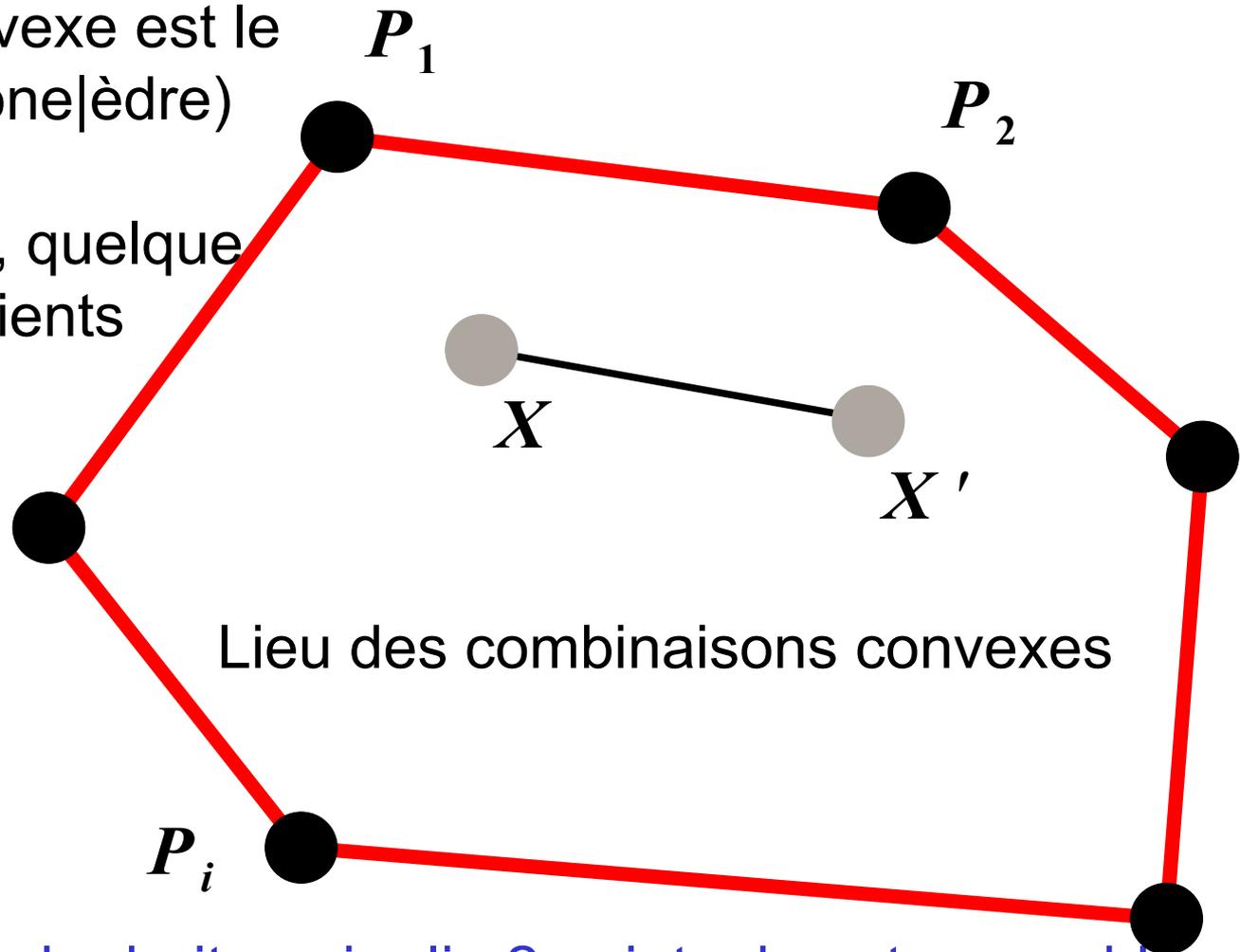
c'est aussi une somme d'un point et de vecteurs

La combinaison convexe est une combinaison barycentrique où tous les coefficients α_i sont non négatifs, leur somme restant égale à 1

La combinaison convexe de points est toujours à « l'intérieur » de l'ensemble des points, ce qui conduit à la définition de l'enveloppe convexe d'un ensemble de points...

L'enveloppe convexe est le plus petit poly(gone|èdre) convexe incluant tous les points X , quelque soient les coefficients α_i positifs et de somme unité :

$$X = \sum_{i=0}^n \alpha_i P_i$$



Tout segment de droite qui relie 2 points de cet ensemble est entièrement situé à l'intérieur de l'enveloppe convexe

Pour définir un **vecteur** à partir de **points** :

$$\mathbf{u} = \sum_{i=0}^n \beta_i \mathbf{P}_i \quad \beta_0 + \dots + \beta_n = 0$$

On a **invariance** par rapport à la translation...

Pour définir un **point** à partir d'autres **points**:

$$\mathbf{X} = \sum_{i=0}^n \alpha_i \mathbf{P}_i \quad \alpha_0 + \dots + \alpha_n = 1$$

Une transformation ϕ est **affine** si elle laisse invariantes les combinaisons affines :

$$X = \sum_{i=0}^n \alpha_i P_i \quad ; \quad X, P_i \in E^3, \alpha_0 + \dots + \alpha_n = 1$$

$$\phi(X) = \sum_{i=0}^n \alpha_i \phi(P_i)$$

$$\phi(X) : X \in E^3 \rightarrow E^3$$

De manière concrète cela signifie par exemple, que le point milieu d'un segment de droite a son image au milieu de l'image du segment de droite...

Transformation affine sous forme matricielle (dans l'espace E^3):

$$\phi(\mathbf{P}) \equiv \mathbf{A} \cdot \mathbf{P} + \mathbf{u}, \quad \mathbf{u} \in \mathbb{R}^3, \quad \mathbf{A} \text{ matrice } 3 \times 3$$

$$\begin{aligned} \phi\left(\sum_0^n \alpha_i \mathbf{P}_i\right) &= \mathbf{A} \cdot \left(\sum_0^n \alpha_i \mathbf{P}_i\right) + \mathbf{u} \\ &= \sum \mathbf{A} \cdot \alpha_i \mathbf{P}_i + \sum \alpha_i \mathbf{u} \\ &= \sum \alpha_i (\mathbf{A} \cdot \mathbf{P}_i + \mathbf{u}) \\ &= \sum \alpha_i \phi(\mathbf{P}_i) \end{aligned}$$

Quelques transformations affines

identité : $u = 0$, $A = I$, I est la matrice identité,

$$u=0 ; A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

translation :

u est le vecteur de translation, $A = I$,

$$u = \begin{bmatrix} a \\ b \\ c \end{bmatrix} ; A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

mise à échelle

$u = 0$, A est une matrice diagonale dont les termes définissent les échelles selon les axes,

$$u=0 ; A = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix}$$

rotation :

$u = 0$, A est une matrice de rotation,

$$u=0 ; A = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Mouvement de corps rigide

Cas particulier important :

Si la matrice \mathbf{A} est orthogonale : $\mathbf{A}^T \mathbf{A} = \mathbf{A}^{-1} \mathbf{A} = \mathbf{I}$

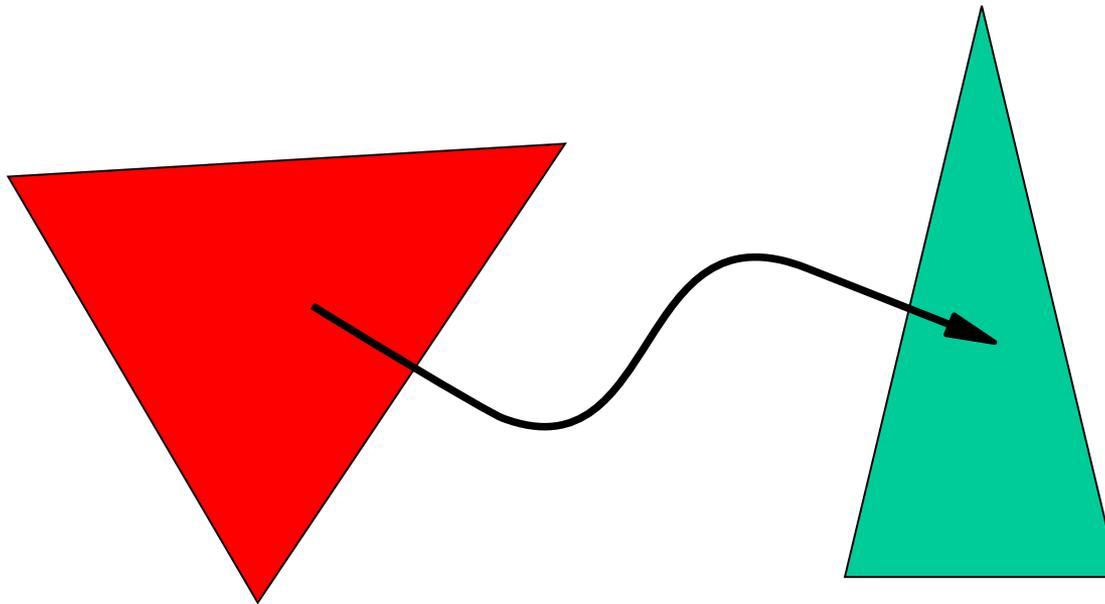
alors **cette transformation conserve les angles et les longueurs.**

Exemple : Translation, Rotation

N.B.: Toutes les transformations affines conservent les **rappports de sections**, mais pas forcément les longueurs et les angles.

Une autre définition...

Une transformation affine de E^2 vers E^2 est univoquement définie par un triangle non dégénéré et son image.

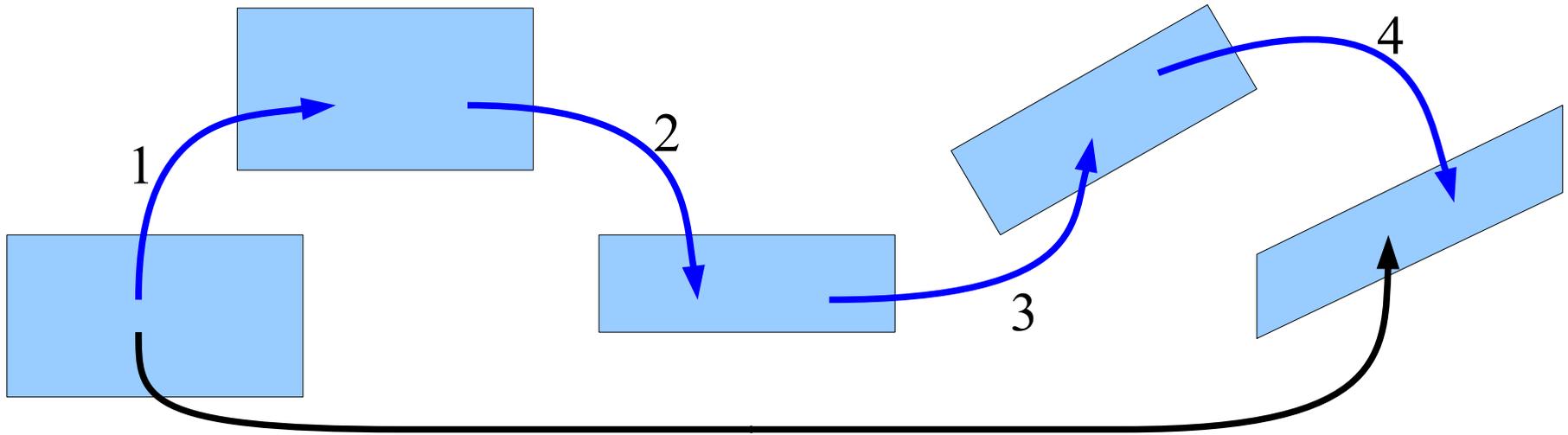


(6 paramètres)

Dans E^3 , la transformation affine est univoquement définie par un tétraèdre non dégénéré et son image.

(12 paramètres)₈

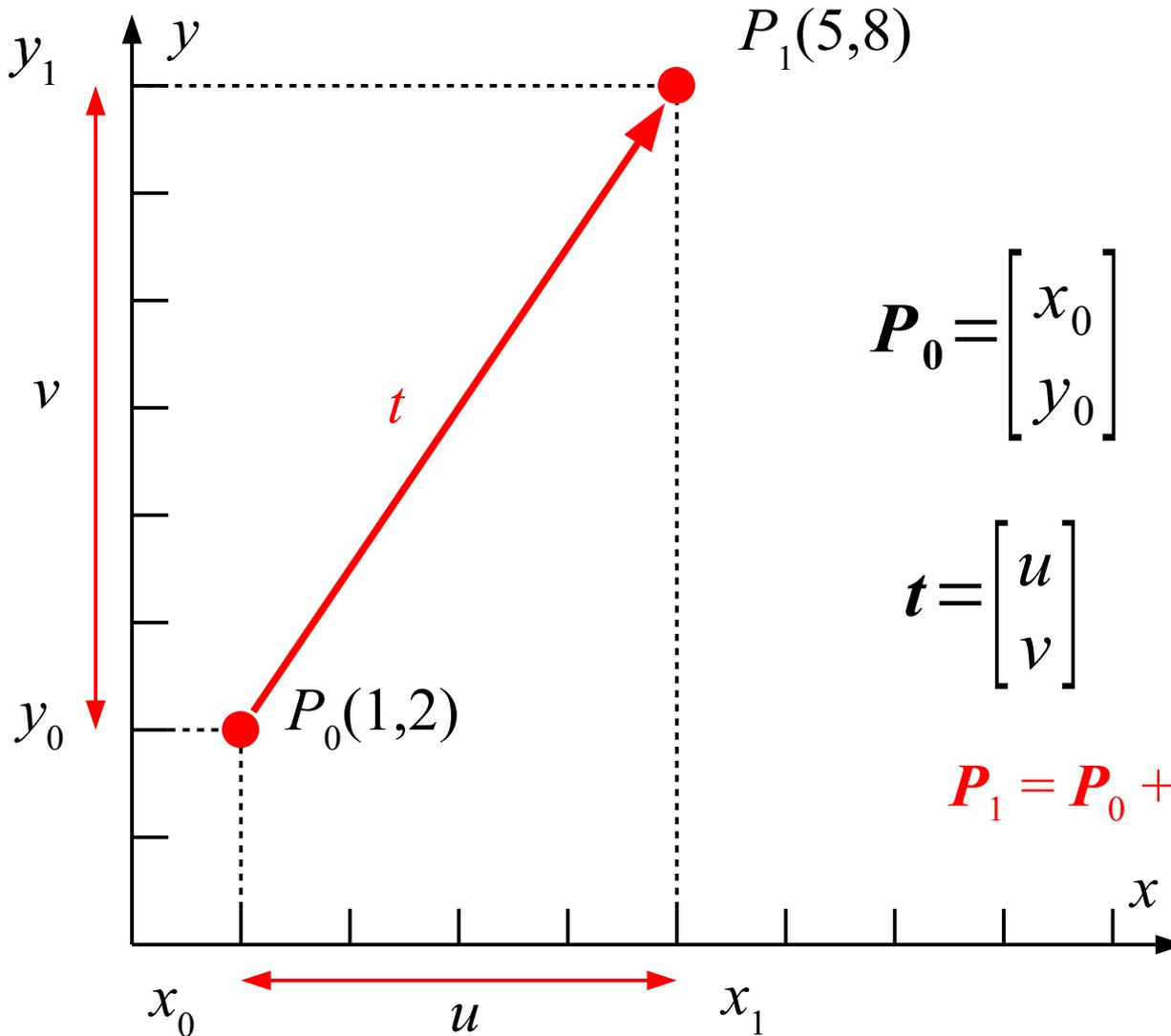
Décomposition d'une transformation affine



Toute transformation affine peut être décomposée en
 translations, mises à échelle, rotations et cisaillements

2D :	2	2	1	1
3D :	3	3	3	3

Transformations en 2 dimensions



Translation

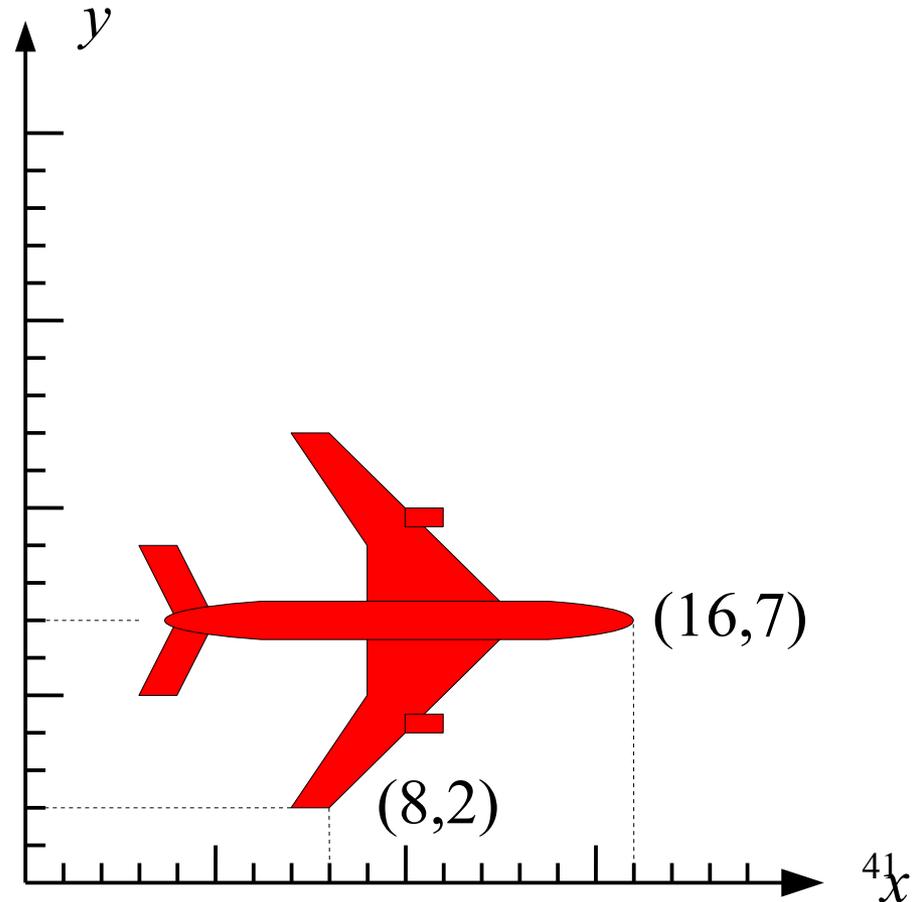
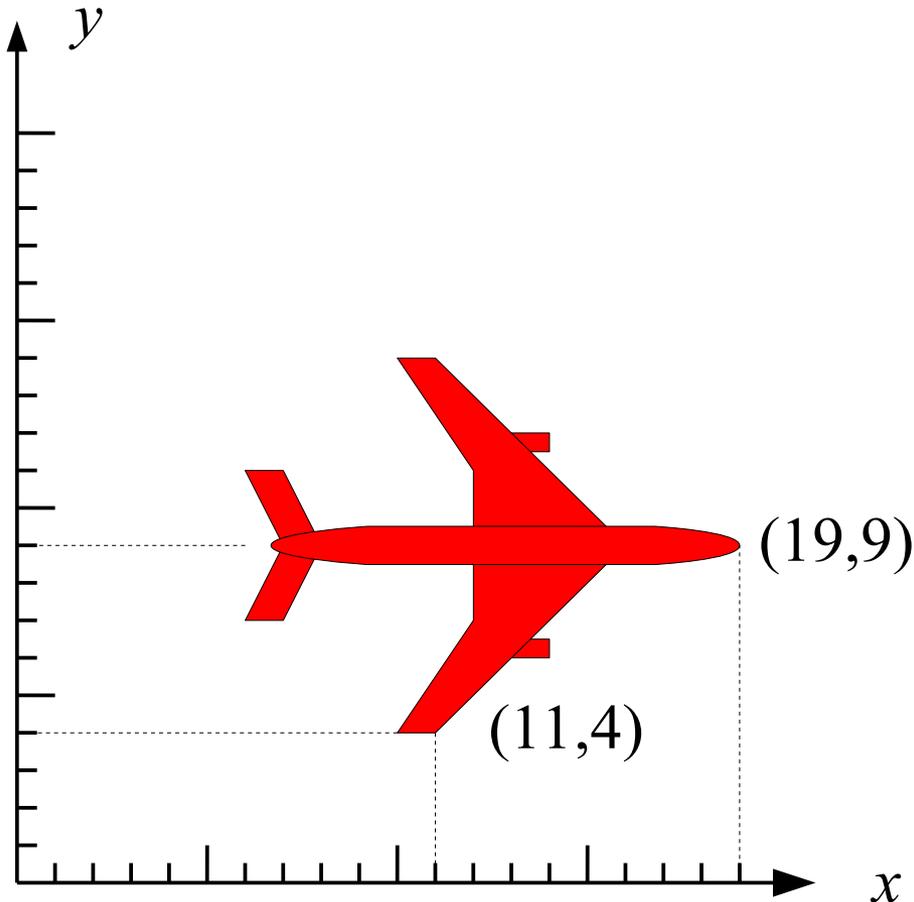
$$P_0 = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad P_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

$$t = \begin{bmatrix} u \\ v \end{bmatrix} \quad \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} u \\ v \end{bmatrix}$$

$$P_1 = P_0 + t$$

Translation

Translation d'un objet : effectuer l'opération sur tous les sommets des polygones qui le définissent



Mise à l'échelle

La mise à l'échelle

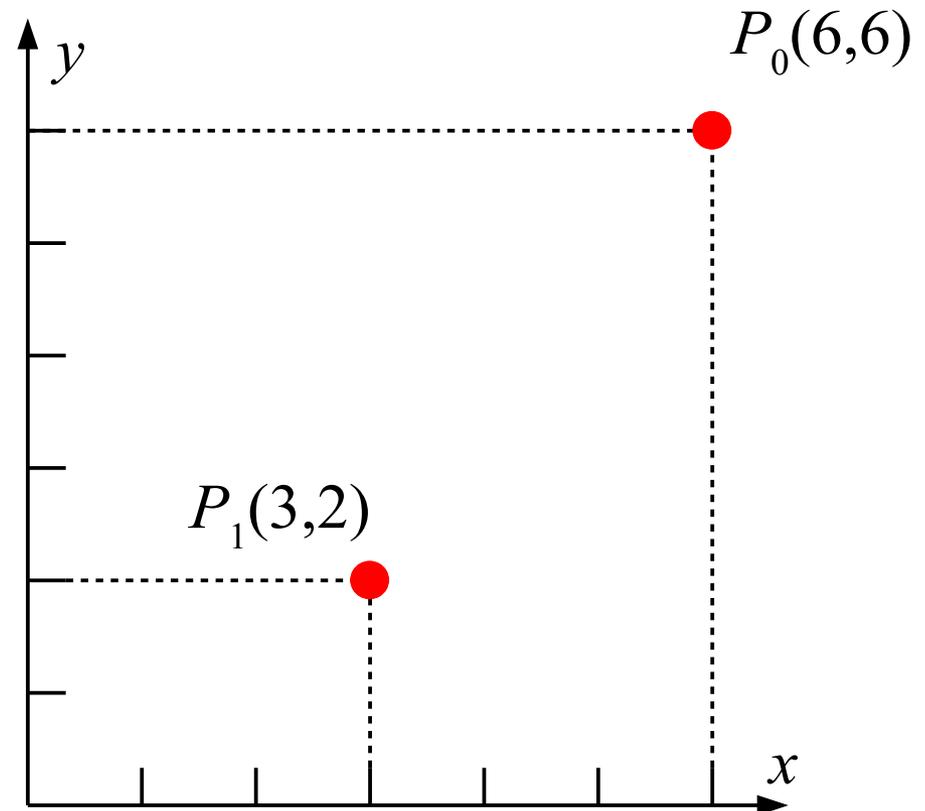
s'effectue par rapport à l'origine.

$$\mathbf{x}_1 = \mathbf{x}_0 \mathbf{p}$$

$$\mathbf{y}_1 = \mathbf{y}_0 \mathbf{q}$$

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} p & 0 \\ 0 & q \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

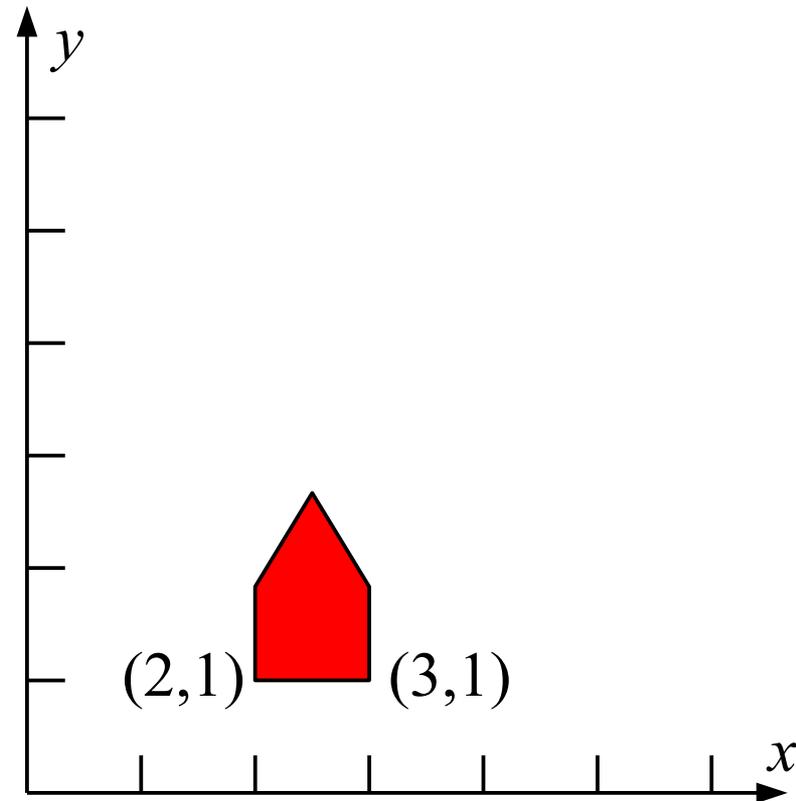
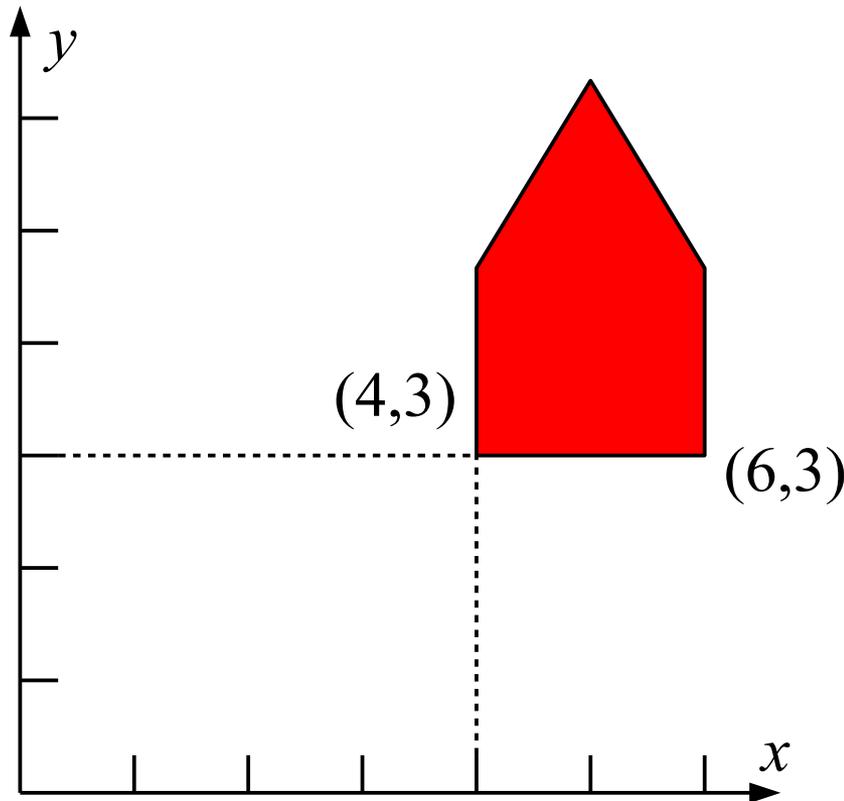
$$\mathbf{P}_1 = \mathbf{S} \mathbf{P}_0$$



Mise à l'échelle

Mise à échelle d'un objet : effectuer l'opération sur tous les sommets des polygones qui le définissent.

Résultat : changement de taille, de proportions et de position

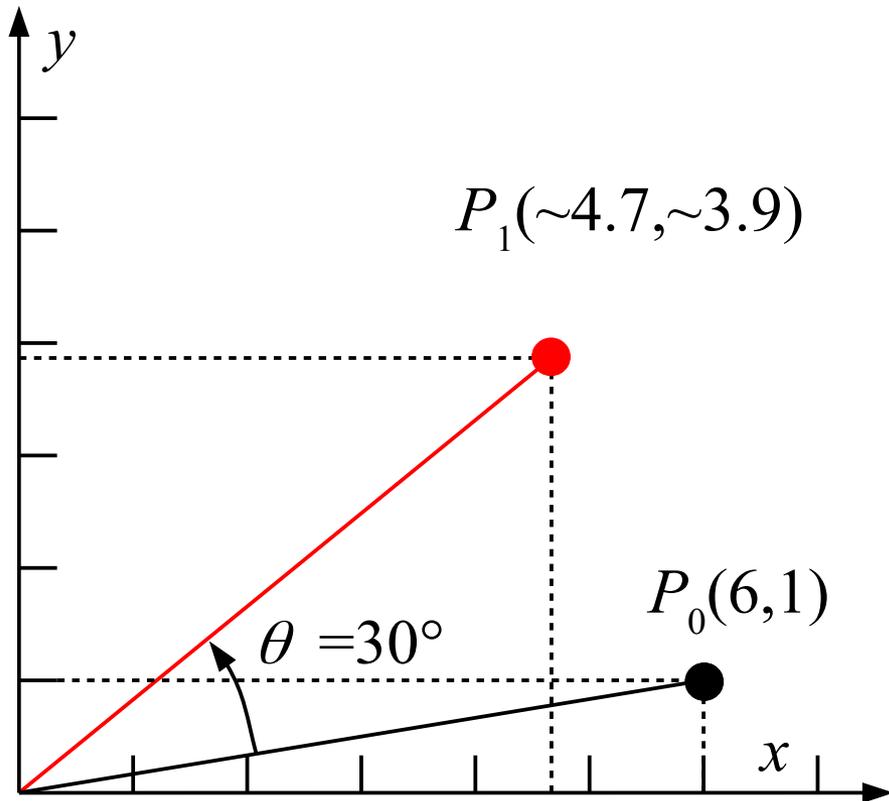


Rotation

La rotation s'effectue par rapport à l'origine

$$x_1 = x_0 \cos \theta - y_0 \sin \theta$$

$$y_1 = x_0 \sin \theta + y_0 \cos \theta$$



$$P_1(\sim 4.7, \sim 3.9)$$

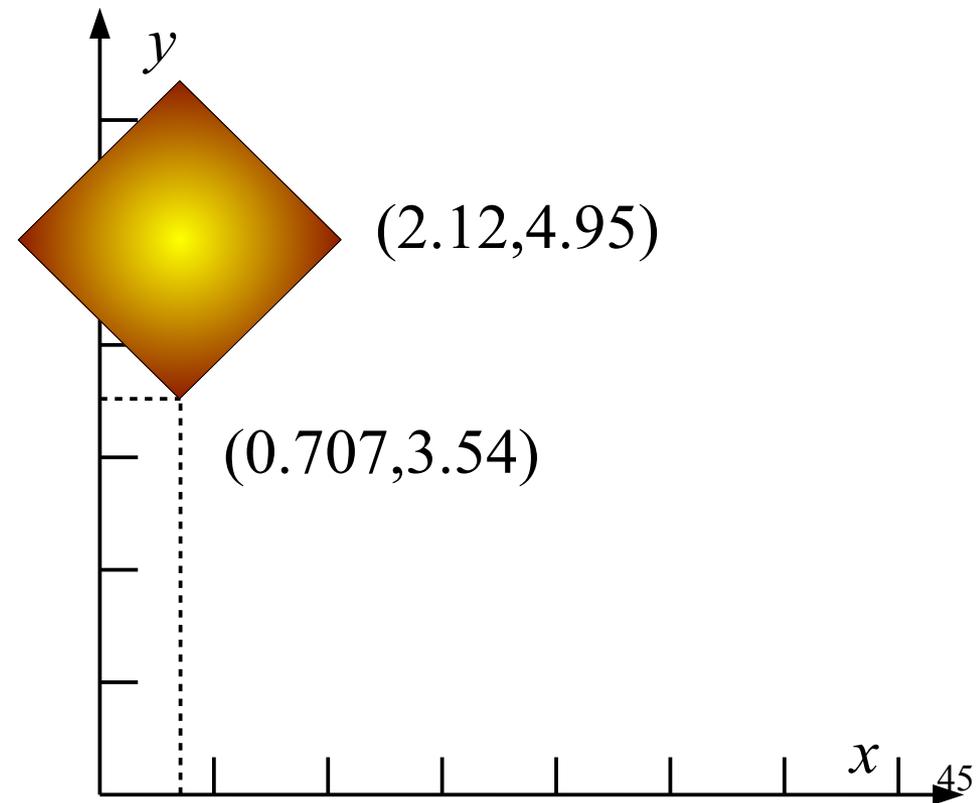
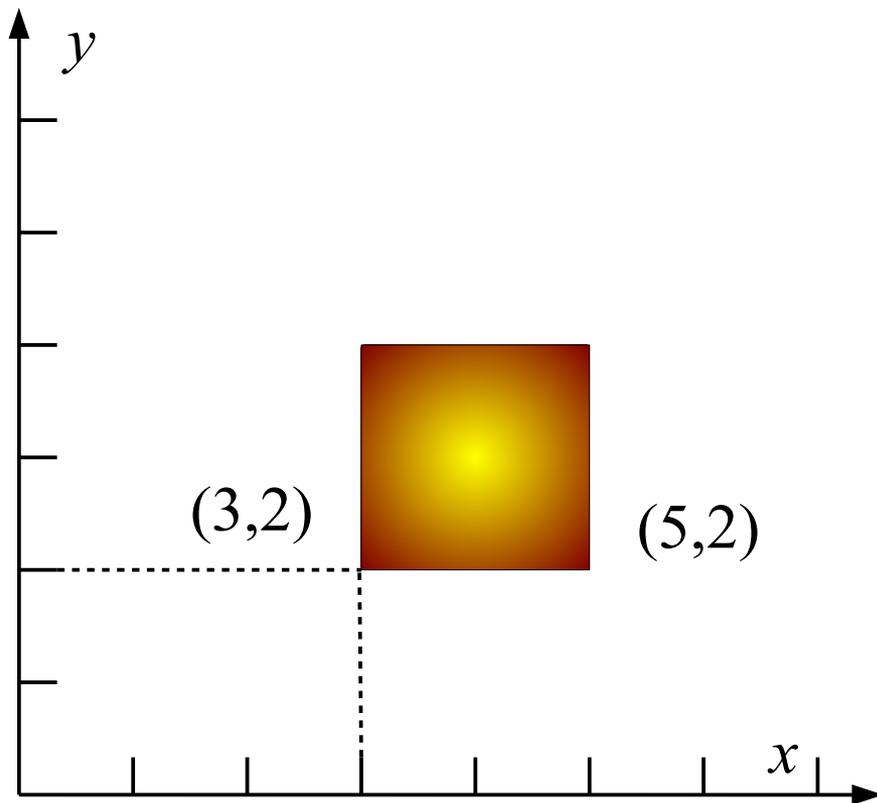
$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

$$P_1 = R P_0$$

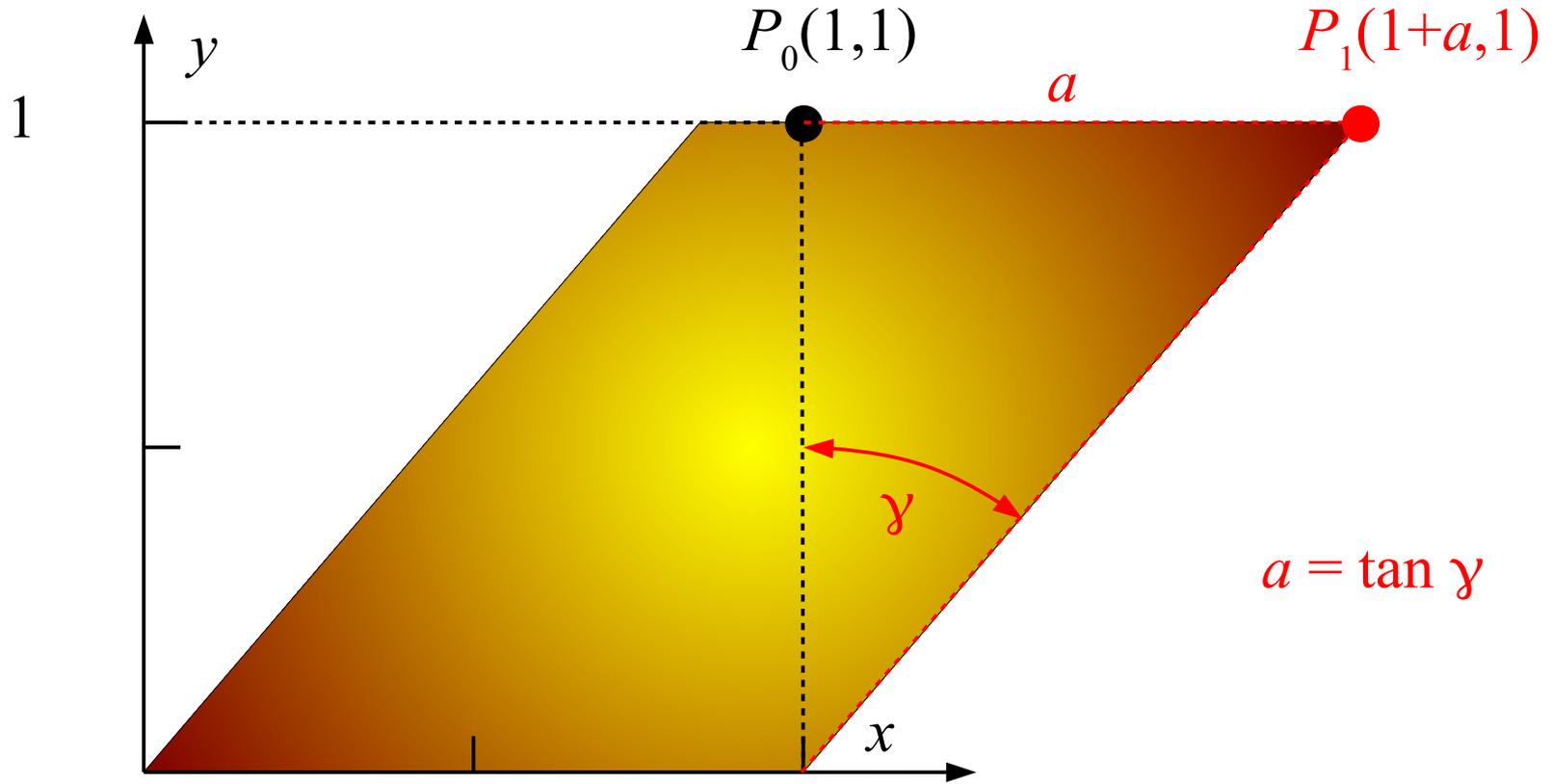
Rotation

Rotation d'un objet : effectuer l'opération sur tous les sommets des polygones qui le définissent.

Résultat : changement d'orientation et de position



Cisaillement



Cisaillement

$$x_1 = x_0 + a y_0$$

$$y_1 = y_0$$

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \mathbf{C} \cdot \mathbf{P}_0$$

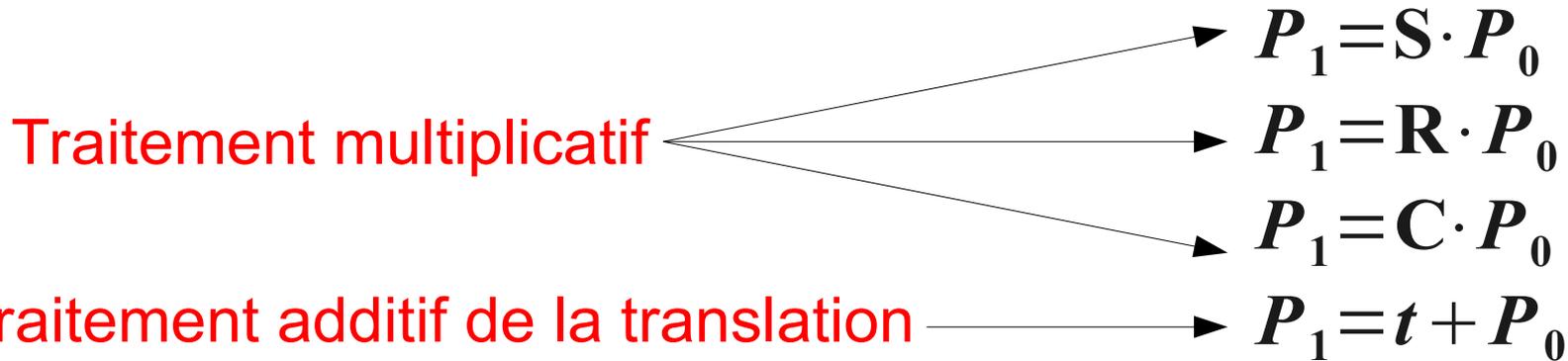
Pour la valeur $y_0 = 1$, le point est simplement translaté de la valeur a .

Cette opération fait glisser les abscisses d'une valeur égale au produit de a par la distance à l'axe des x .

La variable a est la tangente de l'angle γ de glissement

$$a = \tan \gamma$$

Traitement matriciel

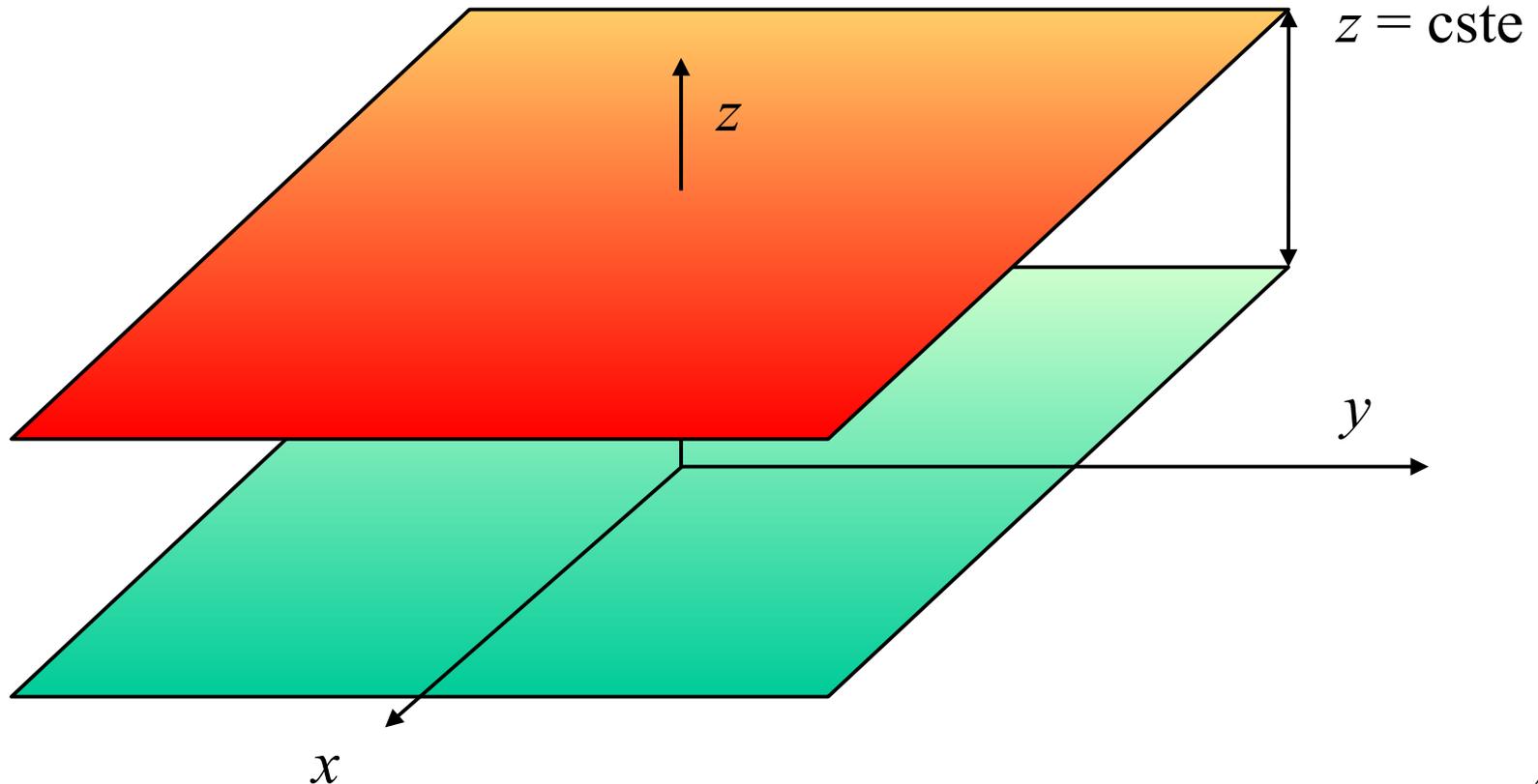


Le traitement n'est pas le même pour toutes les opérations...

Coordonnées homogènes

Au lieu de situer les objets dans un espace à 2 dimensions, plaçons les dans un espace à 3 dimensions.

Toutes les opérations sont inchangées si on met les points dans le plan $x - y$ ou dans un plan parallèle, $z = \text{constante}$



Coordonnées homogènes

Ce changement implique seulement d'ajouter aux points une coordonnée qui ne sera pas modifiée et qui est celle du plan dans lequel nous allons désormais effectuer les opérations.

On l'appelle souvent coordonnée 'poids'. On dit que les points ainsi définis sont exprimés en **coordonnées homogènes**.

Les matrices de transformations ont une dimension **(3 x 3)**, la dernière ligne étant constituées de **zéros** à l'exception du terme diagonal qui vaut **1**.

Cela signifie qu'on ne modifie jamais cette coordonnée $z = \text{cste}$

Nouvelle représentation de la translation

Soit un cisaillement parallèle aux plans $z = \text{cste}$:

$$\begin{bmatrix} 1 & 0 & u \\ 0 & 1 & v \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x + uz \\ y + vz \\ z \end{bmatrix}$$

Le point glisse dans un plan perpendiculaire à z d'une quantité proportionnelle à cette coordonnée et, aux termes u ou v .

Les points du plan $z = 1$ subissent une translation $(u, v, 0)$!

Cette nouvelle formulation peut donc remplacer l'ancienne.

Elle correspond à un cisaillement du plan $z = 1$.

Nouvelle représentation de la translation

Avec la nouvelle formulation, l'équation de translation s'écrit :

$$\begin{bmatrix} x_1 \\ y_1 \\ \textcircled{1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & u \\ 0 & 1 & v \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ y_0 \\ \textcircled{1} \end{bmatrix}$$

$$P_1 = \mathbf{D}(t) \cdot P_0 \quad ; \quad t = \begin{bmatrix} u \\ v \\ 0 \end{bmatrix} \quad \mathbf{D}(t) = \begin{bmatrix} 1 & 0 & u \\ 0 & 1 & v \\ 0 & 0 & 1 \end{bmatrix}$$

L'opération de translation est additive : si le point P_0 est déplacé de (u_1, v_1) en P_1 et ensuite, de (u_2, v_2) en P_2 , c'est comme s'il subissait une translation totale de $(u_1 + u_2, v_1 + v_2)$

Autres transformations

Mise à échelle $\mathbf{S}(p,q)$

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} p & 0 & 0 \\ 0 & q & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

Rotation $\mathbf{R}(\theta)$

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

Cisaillement $\mathbf{C}(a)$

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

Composition de transformations

Le but des développements qui suivent est de montrer comment une combinaison des opérateurs fondamentaux **D**, **S**, **R** et **C** permet d'obtenir le résultat souhaité.

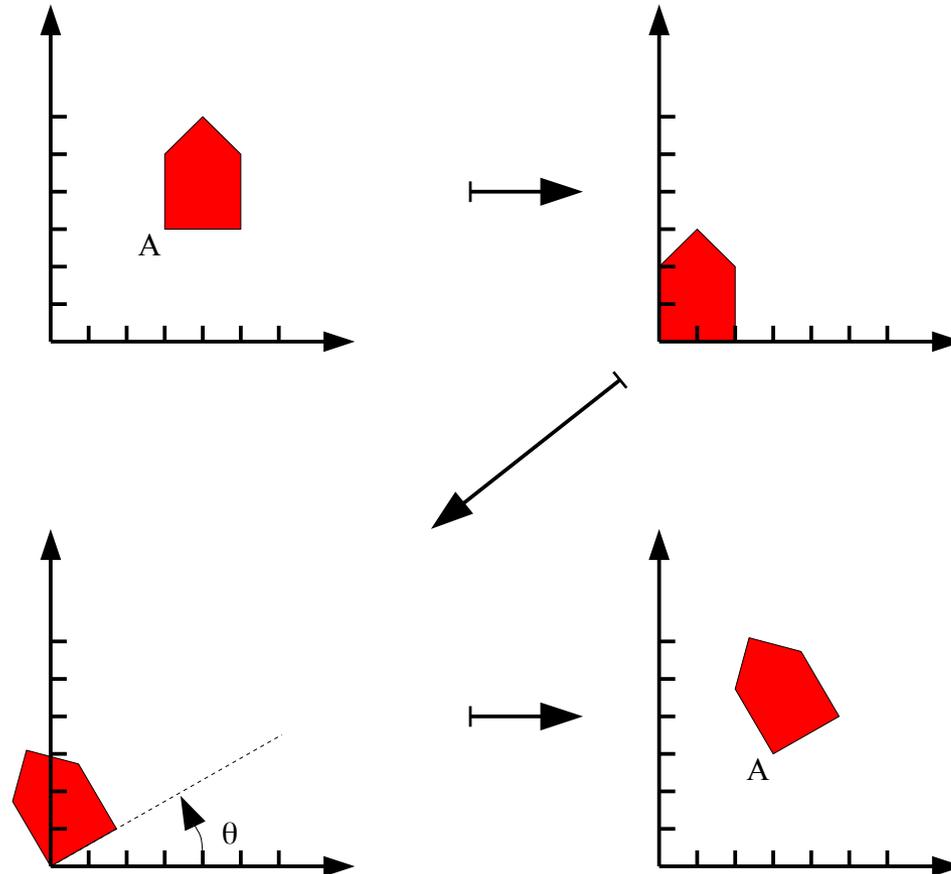
La raison pour laquelle on combine les transformations est qu'il est plus efficace d'appliquer une transformation composée unique à un grand nombre de points que de leur appliquer une série de transformations successives.

La composition se fait par simple **multiplication matricielle**, y compris pour la translation !

Rotation autour d'un point arbitraire

Considérons la rotation d'un objet autour d'un point arbitraire **A**.

- Translation pour amener **A** à l'origine;
- Rotation autour de l'origine;
- Translation pour que le point à l'origine retourne en **A**.



Rotation autour d'un point arbitraire

Rotation par rapport au point **A** différent de l'origine

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -x_A \\ 0 & 1 & -y_A \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \longrightarrow \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_3 \\ y_3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_A \\ 0 & 1 & y_A \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} =$$

$$\begin{bmatrix} 1 & 0 & x_A \\ 0 & 1 & y_A \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_A \\ 0 & 1 & -y_A \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

Rotation autour d'un point arbitraire

Rotation par rapport au point **A** différent de l'origine

$$\begin{bmatrix} 1 & 0 & x_A \\ 0 & 1 & y_A \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_A \\ 0 & 1 & -y_A \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos \theta & -\sin \theta & x_A(1 - \cos \theta) + y_A \sin \theta \\ \sin \theta & \cos \theta & y_A(1 - \cos \theta) + x_A \sin \theta \\ 0 & 0 & 1 \end{bmatrix}$$

Mise à l'échelle autour d'un point arbitraire

Mise à l'échelle par rapport au point **A** différent de l'origine

$$\begin{bmatrix} 1 & 0 & x_A \\ 0 & 1 & y_A \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} p & 0 & 0 \\ 0 & q & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_A \\ 0 & 1 & -y_A \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} p & 0 & x_A(1-p) \\ 0 & q & y_A(1-q) \\ 0 & 0 & 1 \end{bmatrix}$$

Symétries

Les symétries orthogonales d'axes **O-x** ou **O-y** s'écrivent respectivement

S(1,-1) et **S(-1,1)**

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{et} \quad \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

et la symétrie centrale de centre **O**,

S(-1,-1).

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(non) Commutativité

En général si l'on a deux matrices de transformation \mathbf{M}_1 et \mathbf{M}_2 : appliquer \mathbf{M}_1 puis \mathbf{M}_2 est différent d'appliquer \mathbf{M}_2 puis \mathbf{M}_1
C'est à dire que l'on a en général pas commutativité...
sauf dans les cas suivants :

Commutativité ($\mathbf{M}_1\mathbf{M}_2 = \mathbf{M}_2\mathbf{M}_1$) si :

\mathbf{M}_1

translation

mise à l'échelle

rotation autour de \mathbf{O}_z

mise à l'échelle ($p=q$)

\mathbf{M}_2

translation

mise à l'échelle

rotation autour de \mathbf{O}_z

rotation autour de \mathbf{O}_z

Transformations en 3D

- Même idée qu'en 2D: On rajoute une coordonnée « homogène » aux points, ce qui mène à des points exprimés en 4 dimensions...
- Les matrices de transformation sont 4x4.

$$\mathbf{P}_{3D} = \begin{bmatrix} x_P \\ y_P \\ z_P \end{bmatrix} \rightarrow \mathbf{P}_{\text{hom}} = \begin{bmatrix} x_P \\ y_P \\ z_P \\ 1 \end{bmatrix} \quad \mathbf{P}_{\text{hom}}^* = \begin{bmatrix} x_P^* \\ y_P^* \\ z_P^* \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c & u \\ d & e & f & v \\ g & h & i & w \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_P \\ y_P \\ z_P \\ 1 \end{bmatrix}$$

Transformations en 3D

- Translation : $\mathbf{D}(\mathbf{t}) = \begin{bmatrix} 1 & 0 & 0 & u \\ 0 & 1 & 0 & v \\ 0 & 0 & 1 & w \\ 0 & 0 & 0 & 1 \end{bmatrix} ; \mathbf{t} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$

- Mise à échelle : $\mathbf{S}(p, q, r) = \begin{bmatrix} p & 0 & 0 & 0 \\ 0 & q & 0 & 0 \\ 0 & 0 & r & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Transformations en 3D

- Rotation autour de l'axe z : $\mathbf{R}_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

- Rotation autour de l'axe x : $\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

- Rotation autour de l'axe y : $\mathbf{R}_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Transformations en 3D

- Cisaillement (cas général): $\mathbf{C}(a, b, c) = \begin{bmatrix} 1 & a & b & 0 \\ 0 & 1 & c & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Projections : 3D->2D (ou 1D)

- Projections orthogonales d'un point $(x \ y \ z)$

- sur un plan perpendiculaire à $z : z = k$

$$(x \ y \ z) \Rightarrow (x \ y \ k)$$

$$\mathbf{P}_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & k \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- sur un plan perpendiculaire à $y : y = l$

$$(x \ y \ z) \Rightarrow (x \ l \ z)$$

- sur un plan perpendiculaire à $x : x = m$

$$(x \ y \ z) \Rightarrow (m \ y \ z)$$

$$\mathbf{P}_x = \begin{bmatrix} 0 & 0 & 0 & m \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

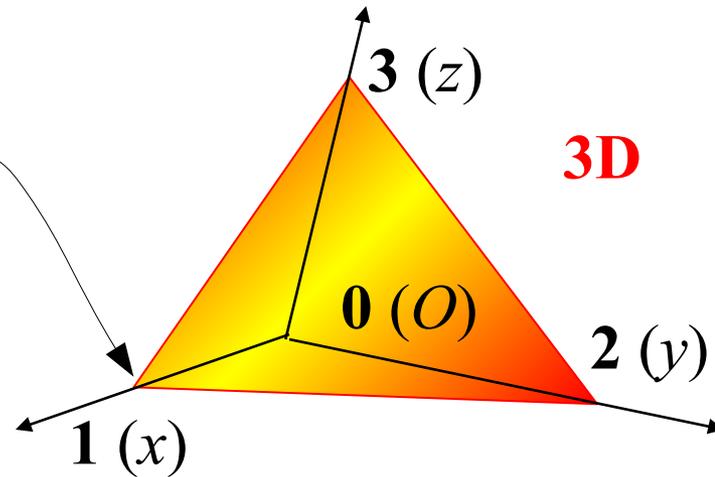
Important de noter que tout ceci est indépendant du point (ou de l'objet).

Calcul des axonométries

- Déterminer la matrice de transformation pour une axonométrie quelconque
 - On exprime les coordonnées 3D initiales et finales d'un ensemble de points
 - On dispose de 12 inconnues, il faut donc 4 points
 - Le trièdre (O,x,y,z) fait l'affaire ...

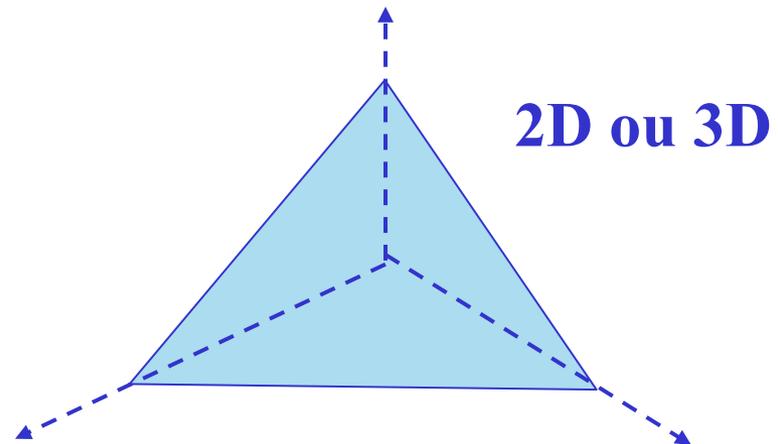
Calcul des axonométries

$$P_0 = \begin{matrix} & \mathbf{0} & \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \begin{bmatrix} 0 & \mathbf{1} & 0 & 0 \\ 0 & \mathbf{0} & 1 & 0 \\ 0 & \mathbf{0} & 0 & 1 \\ 1 & \mathbf{1} & 1 & 1 \end{bmatrix} \end{matrix}$$



$$P_1 = T \cdot P_0$$

$$P_1 = \begin{bmatrix} x_0 & x_1 & x_2 & x_3 \\ y_0 & y_1 & y_2 & y_3 \\ \mathbf{z_0} & \mathbf{z_1} & \mathbf{z_2} & \mathbf{z_3} \\ 1 & 1 & 1 & 1 \end{bmatrix}$$



Calcul des axonométries

$$P_1 = T \cdot P_0$$

$$T = P_1 \cdot P_0^{-1}$$

$$P_0^{-1} = \begin{bmatrix} -1 & -1 & -1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$P_1 = \begin{bmatrix} x_0 & x_1 & x_2 & x_3 \\ y_0 & y_1 & y_2 & y_3 \\ z_0 & z_1 & z_2 & z_3 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

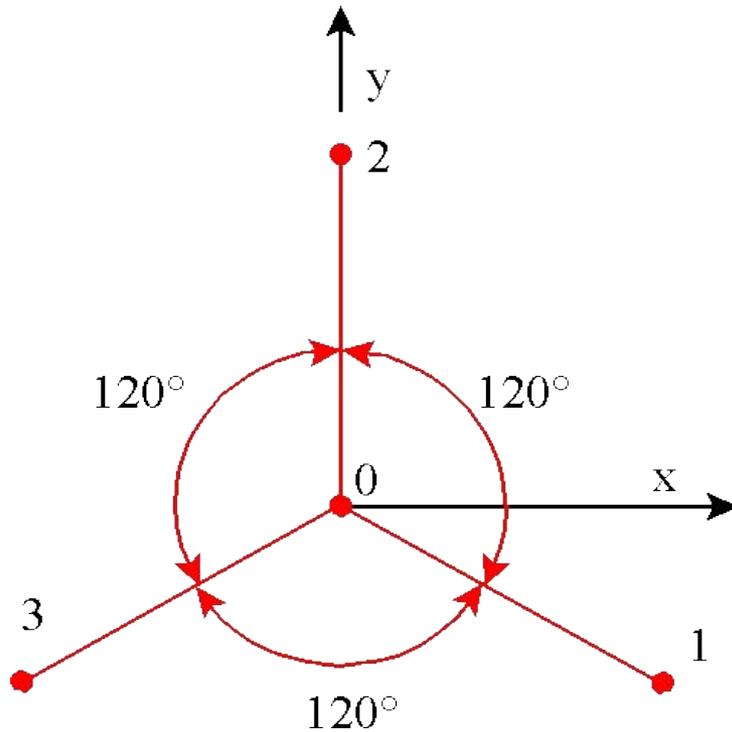
$$T = \begin{bmatrix} x_1 - x_0 & x_2 - x_0 & x_3 - x_0 & x_0 \\ y_1 - y_0 & y_2 - y_0 & y_3 - y_0 & y_0 \\ z_1 - z_0 & z_2 - z_0 & z_3 - z_0 & z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = T^{-1} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Cette matrice permet de projeter les coordonnées dans un système d'axes quelconques

(L'utilisation de son inverse* permet d'exprimer les coordonnées d'origine dans le nouveau repère, mais ce n'est pas ce que l'on veut ici !)

Perspective isométrique



$$\mathbf{T} = \begin{bmatrix} \frac{\sqrt{3}}{2} & 0 & -\frac{\sqrt{3}}{2} & 0 \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Calcul des axonométries

$$\mathbf{T} = \begin{bmatrix} \frac{\sqrt{3}}{2} & 0 & -\frac{\sqrt{3}}{2} & 0 \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \leftarrow$$

Inverse* : Ici T n'est pas inversible, car le repère de destination n'est pas constitué de vecteurs linéairement indépendants. On prend un pseudo-inverse ou on complète le repère...

Perspective cavalière

Pour une perspective cavalière :

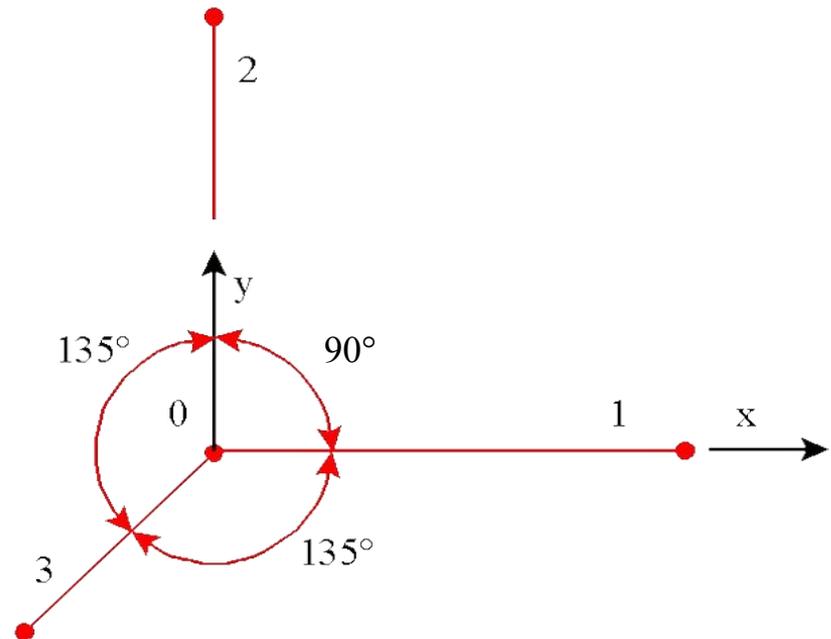
$$x_0=0 \qquad y_0=0$$

$$x_1=1 \qquad y_1=0$$

$$x_2=0 \qquad y_2=1$$

$$x_3 = -0.6 \frac{\sqrt{2}}{2} \qquad y_3 = -0.6 \frac{\sqrt{2}}{2}$$

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & -3 \frac{\sqrt{2}}{10} & 0 \\ 0 & 1 & -3 \frac{\sqrt{2}}{10} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Bézier & transformations affines

- Et pour les courbes de Bézier ?

$$P_j^{i+1} = (1-u)P_j^i + uP_{j+1}^i, 0 \leq u \leq 1$$

À chaque étape, il s'agit bien d'une combinaison affine, à poids positifs et de somme unité :

$$X = \sum_{i=0}^n \alpha_i P_i \quad X, P_i \in E^3, \alpha_0 + \dots + \alpha_n = 1, \alpha_i \geq 0$$

- Il s'agit donc d'une combinaison convexe.
les courbes de Bézier sont donc contenues dans l'enveloppe convexe des points de contrôle.
- Cette propriété est utile en CAO...

Bézier & transformations affines

- De plus, toute transformation affine ϕ laisse invariante toute combinaison convexe.

$$X = \sum_{i=0}^n \alpha_i P_i \quad X, P_i \in E^3, \alpha_0 + \dots + \alpha_n = 1, \alpha_i \geq 0$$

- Cela signifie que la transformation affine des points d'une courbe de Bézier s'obtient simplement en faisant la même transformation pour les points de contrôle qui la définissent...

$$\phi(X) = \phi\left(\sum_{i=0}^n \alpha_i P_i\right) = \sum_{i=0}^n \alpha_i \phi(P_i)$$

- Cela a d'énormes implications en CAO : inutile de mémoriser l'historique des transformations...

Projet

Projet

- Échéance : 23/12/2023
- L'énoncé sera mis en ligne d'ici peu
- Nous vous conseillons de commencer à travailler dessus dès que possible, et ne pas attendre la dernière minute
- Travail **personnel**
- De façon générale, les forums (sur e-campus) sont une bonne place pour poser des questions, que ce soit pour le projet ou les Tps.