TP 2 : Techniques de reconstruction de surfaces gauches

2.1. Objectifs du TP

Ce travail pratique explore les techniques de reconstruction de surfaces gauches, notamment les courbes de Bézier, les courbes à pôles, les B-Splines et les NURBS (Non-Uniform Rational B-Splines). L'objectif est de comprendre leurs principes fondamentaux, les fonctions de base et leurs propriétés afin de modéliser des surfaces non développables.

Des exercices permettent d'explorer différentes méthodes de modélisation géométrique en utilisant MATLAB pour tracer des courbes complexes. Chaque exercice est conçu pour aider à comprendre les principes sous-jacents des courbes paramétriques et de leur influence sur la forme des surfaces gauches

2.2. Définition d'une surface gauche

Une surface gauche est une surface non développable, c'est-à-dire qu'on ne peut pas l'aplatir sur un plan sans la déformer par étirement ou compression telle que celles illustrées sur la Figure 1 :

- L'hélicoïde (a) comme la vis (b) ou la rampe d'escalier (c).
- Le cylindre oblique (d) ou le cône tordu (e).
- La surface d'une hélice (f) ou d'un ruban de Möbius (g) .

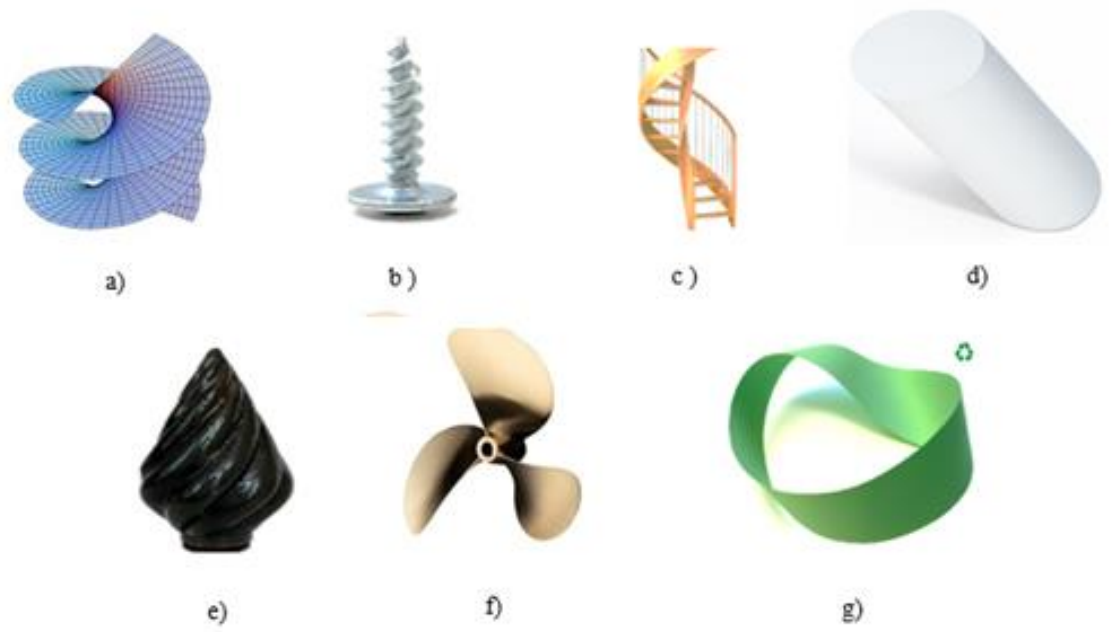


Figure 1: Exemples de surfaces gauches.

En géométrie et conception assistée par ordinateur (CAO), les surfaces gauches sont définies par des courbes génératrices et directrices, comme les courbes de Bézier, les courbes à pôles, les B-Splines et les NURBS (Non-Uniform Rational B-Splines), qui permettent de modéliser des formes complexes et fluides (carrosseries, ailes d'avion, etc.).

2.3. Techniques de reconstruction des surfaces gauches

2.3.1. Courbes de Bézier

Les courbes de Bézier sont des courbes paramétriques définies par des polynômes de Bernstein $B_{i,n}(t)$ et des points de contrôle P_i. Une courbe de Bézier de degré n est donnée par l'équation suivante :

$$B(t) = \sum_{i=0}^{n} B_{i,n}(t) P_i$$

Les polynômes de Bernstein de degré n sont définis sur [0,1] comme suit :

$$B_{i,n}(t) = \begin{cases} C_i^n t^i (1-t)^{n-i} & si\ 0 \le i \le n \\ 0 & \text{Ailleurs} \end{cases}$$

avec : $C_i^n = \dfrac{n!}{i!\,(n-i)!}$; C_i^n : Coefficient binomial.

Ces polynômes vont nous servir par la suite pour définir les courbes de Bézier.

Exemple :

Dans le cas cubique ($n = 3$), les polynômes Bernstein sont donnés par :

$B_{0,3}(t) = (1-t)^3$

$B_{1,3}(t) = 3t(1-t)^2$

$B_{2,3}(t) = 3t^2(1-t)$

$B_{3,3}(t) = t^3$

On voit qu'ils sont tous positifs pour $t \in [0,1]$ et que leur somme est égale à 1.

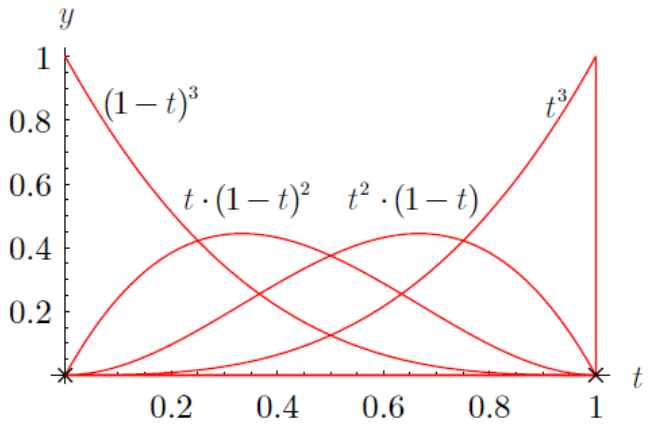


Figure 2: Polynômes de Bernstein de degré 3.

2.3.1.1. Propriétés des courbes de Bézier

- **Linéarité :** Une courbe de Bézier de degré n passe par les deux premiers et derniers points de contrôle.
- **Barycentre :** Les points de la courbe sont des combinaisons convexes des points de contrôle.
- **Degré :** Le degré de la courbe est égal au nombre de points de contrôle moins un.

2.3.1.2. Exercice 1 : Courbes de Bézier

L'objectif de cet exercice est de comprendre et de visualiser la courbe de Bézier de degré 2 en utilisant MATLAB.

1. Définissez trois points de contrôle : $P_0 = (0,0)$, $P_1 = (1,2)$, et $P_2 = (2,0)$.
2. Utilisez la formule des polynômes de Bernstein pour calculer les points sur la courbe de Bézier pour $t$allant de 0 à 1.
3. Tapez le code Matlab ci-après et tracez la courbe de Bézier et les points de contrôle.

2.3.1.3. Code MATLAB pour une courbe de Bézier (3 points de contrôle) :

```
% Définir les points de contrôle
P0 = [0, 0];
P1 = [1, 2];
P2 = [2, 0];

% Nombre de points pour la courbe
t = linspace(0, 1, 100);

% Calcul des polynômes de Bernstein
B0 = (1 - t).^2;
B1 = 2 * t .* (1 - t);
B2 = t.^2;

% Calcul des points sur la courbe
```

```
B = B0' * P0 + B1' * P1 + B2' * P2;

% Tracer la courbe de Bézier
figure;
plot(B(:,1), B(:,2), 'b-', 'LineWidth', 2);
hold on;
plot([P0(1), P1(1), P2(1)], [P0(2), P1(2), P2(2)], 'ro-', 'MarkerFaceColor', 'r');
title('Courbe de Bézier de degré 2');
xlabel('x');
ylabel('y');
grid on;
axis equal;
```

2.3.1.4. Questions de réflexion

- Quelles sont les propriétés des courbes de Bézier ? Comment les points de contrôle influencent-ils la forme de la courbe ?
- Que se passe-t-il si vous ajoutez un quatrième point de contrôle ?

2.3.2. Courbes à Pôles

Les courbes à pôles sont une variante des courbes de Bézier où les points de contrôle sont pondérés. Elles sont exprimées comme suit :

$$C(t) = \sum_{i=0}^{n} \lambda_i B_{i,n}(t)\, P_i$$

où λ_i sont les poids appliqués aux points de contrôle P_i

2.3.2.1. Propriétés des courbes à pôles

- **Interactivité :** Les pôles peuvent être ajustés pour modifier la courbe de manière intuitive.
- **Non-développementabilité :** Ces courbes peuvent être utilisées pour modéliser des surfaces gauches.

2.3.2.2. Exercice 2 : Courbes à Pôles

Cet exercice vise à explorer les courbes à pôles, qui sont une variante des courbes de Bézier avec des poids.

1. Définissez trois points de contrôle : $P_0 = (0,0)$, $P_1 = (1,2)$, et $P_2 = (2,0)$.
2. Attribuez des poids différents aux points de contrôle : $\lambda_0 = 1$, $\lambda_1 = 2$, et $\lambda_2 = 1$.

3. Utilisez la formule des courbes à pôles pour calculer les points sur la courbe pour t allant de 0 à 1.
4. Ecrivez un code Matlab pour tracez la courbe à pôles et les points de contrôle.

2.3.2.3. Questions de réflexion

- Comment les poids influencent-ils la forme de la courbe ?
- Que se passe-t-il si vous ajustez les poids des points de contrôle ?

2.3.3. B-Splines : Fonction de Base et Propriétés

Les B-Splines de degré p sont définies par l'équation :

$$C(t) = \sum_{i=0}^{n} N_{i,p}(t) P_i$$

où $N_{i,p}(t)$sont les fonctions de base B-Splines et P_i sont les points de contrôle.

Les **fonctions de base B-Splines** $N_{i,p}(t)$ sont calculées à partir d'un vecteur de nœuds, et elles sont définies de manière récursive. Voici comment elles sont généralement définies :

2.3.3.1. Fonctions de base B-Splines d'ordre 0 (discrètes)

Pour $p = 0$ (degré 0), les fonctions de base sont définies comme suit :

$$N_{i,p}(t) = \begin{cases} 1, & si\ t_i \le t < t_{i+1} \\ 0, & sinon \end{cases}$$

Cela signifie que pour $p = 0$, la fonction $N_{i,0}(t)$ vaut 1 uniquement lorsque t se trouve dans l'intervalle $[t_i, t_{i+1}]$, et 0 ailleurs.

2.3.3.2. Fonctions de base B-Splines d'ordre p (degré p)

Pour un degré $p > 0$, les fonctions de base B-splines sont définies de manière récursive par la formule suivante :

$$N_{i,p}(t) = \frac{t - t_i}{t_{i+p} - t_i} N_{i,p-1}(t) + \frac{t_{i+p+1} - t}{t_{i+p+1} - t_{i+1}} N_{i+1,p-1}(t)$$

Cela signifie que les fonctions de base B-Splines d'un degré p sont une combinaison linéaire de celles du degré $p - 1$, et leur forme dépend du paramètre t, ainsi que des nœuds t_i.

2.3.3.3. Propriétés des fonctions de base B-Splines

1) **Non-négativité :** $N_{i,p}(t) \geq 0$ pour tout t et i.
2) **Partition de l'unité :** La somme de toutes les fonctions de base $N_{i,p}(t)$ pour tous les indices i est égale à 1 pour tout t:

$$\sum_{i=0}^{n} N_{i,p}(t) = 1$$

3) **Support local :** Chaque fonction $N_{i,p}(t)$ est non nulle seulement sur un intervalle défini par les nœuds t_i et t_{i+p+1}. Cela permet un contrôle local sur la courbe ou la surface.

2.3.3.4. Propriétés des B-Splines

1) **Localité :** Chaque fonction B-Splines affecte seulement un sous-ensemble de points de contrôle.
2) **Lissité :** Les B-Splines assurent une courbure continue et lisse entre les segments de la courbe.
3) **Non-développementabilité :** Elles peuvent modéliser des surfaces gauches de manière flexible et fluide.

2.3.3.5. Exercice 3 : B-Splines

Cet exercice est destiné à explorer les B-Splines, qui permettent de modéliser des courbes lisses à partir de points de contrôle.

1. Définissez quatre points de contrôle : $P_0 = (0,0)$, $P_1 = (1,3)$, $P_2 = (2,1)$, et $P_3 = (3,4)$.
2. Utilisez un vecteur de nœuds pour une B-Splines cubique et calculez les points sur la courbe.
3. Ecrivez un code Matlab pour tracez la courbe B-Splines et les points de contrôle.

2.3.3.6. Questions de réflexion

- Quelle est la différence entre une courbe B-Splines et une courbe de Bézier ?
- Comment les nœuds influencent-ils la forme de la courbe B-spline ?

2.3.4. NURBS (Non-Uniform Rational B-Splines)

Les NURBS sont des courbes rationnelles définies par des poids. Une courbe NURBS de degré P est définie par l'équation :

$$C(t) = \frac{\sum_{i=0}^{n} N_{i,p}(t)\omega_i P_i}{\sum_{i=0}^{n} N_{i,p}(t)\omega_i}$$

où ω_i sont les poids, P_i les points de contrôle, et $N_{i,p}(t)$ les fonctions de base B-splines.

2.3.4.1. Propriétés des NURBS

1) **Rationalité :** Les NURBS utilisent des poids associés aux points de contrôle, permettant de modéliser des formes plus complexes comme des arcs ou des coniques.
2) **Flexibilité :** Elles peuvent décrire des courbes et surfaces non développables, offrant une plus grande liberté de modélisation.

2.3.4.2. Exercice 3 : NURBS (Non-Uniform Rational B-Splines)

Dans cet exercice, vous allez explorer les courbes NURBS en utilisant des poids pour modéliser des formes plus complexes.

1. Définissez trois points de contrôle : $P_0 = (0,0)$, $P_1 = (1,2)$, et $P_2 = (2,0)$.
2. Attribuez des poids différents : $\omega_0 = 1$, $\omega_1 = 2$, et $\omega_2 = 1$.
3. Utilisez les fonctions de base B-Splines pour calculer les points sur la courbe NURBS.
4. Tracez la courbe NURBS et les points de contrôle sous Matlab.

2.3.4.3. Questions de réflexion

- Quel est l'impact des poids sur la forme de la courbe NURBS par rapport à une courbe de Bézier classique ?
- Que se passe-t-il si vous modifiez les poids ?

2.4. Conclusion

Les techniques de reconstruction de surfaces gauches à l'aide des courbes de Bézier, des courbes à pôles, des B-Splines et des NURBS sont des outils puissants pour la modélisation de géométries complexes. Les codes MATLAB permettent de visualiser ces courbes et surfaces, facilitant leur application dans des domaines comme l'architecture, la modélisation 3D, et l'ingénierie. Ces courbes sont essentielles pour travailler avec des surfaces non développables, notamment dans les domaines de la géométrie architecturale et industrielle.