**ENSIMAG 3ème année** **2018-19**
**Modélisation Surfacique**

---

# TP — Surface Interrogation (surfaces de Bézier)

durée 5-7h

Le but du TP est d'implémenter 2 méthodes d'interrogation de surfaces et de les tester sur des surfaces de Bézier.

Un programme Matlab est à votre disposition pour ce TP. C'est un programme graphique qui, à partir d'un ensemble de $k \cdot 16$ points de contrôle $\boldsymbol{b}_{ij}^k \in \mathbb{R}^3$, $i, j = 0, \ldots, 3$ calcule et dessine $k$ surfaces de Bézier bicubiques

$$X^k(u,v) = \sum_{i=0}^{3} \sum_{j=0}^{3} \boldsymbol{b}_{ij}^k B_i^3(u) B_j^3(v) \ , \quad (u,v) \in [a,b] \times [c,d].$$

Vous pouvez faire ce TP avec le langage de votre choix: Matlab (ou Scilab), C/C++ ou Python avec OpenGL, etc. Si vous souhaitez utiliser Matlab, vous pouvez télécharger sur Chamilo l'archive *BEZIER-SURF.zip*, démarrer *matlab* et exécuter le programme principal *mainBezier.m* avec la commande *mainBezier*. Ensuite vous pouvez utiliser ce programme comme cadre pour votre TP.

### 1. Génération de surfaces

La fonction *load* permet de lire des données à partir d'un fichier ascii. Nous utilisons cette fonction pour lire les points de contrôle de la surface à partir d'un fichier. Dans le répertoire se trouve un fichier exemple, nommé *surface1*. Il contient les 16 points de contrôle d'un patch de Bézier.
Générez vous-même au moins 3 autres exemples de surfaces (au moins une avec $k > 1$, i.e. plusieurs morceaux de surfaces avec raccord $C^0$ ou raccord $C^1$) et sauvez les points de contrôle respectives dans 3 fichiers *surfacei*. $i = 2, 3, 4$.
Note: Vous pouvez écrire des petites routines Matlab pour générer les points de contrôle (p.ex. en échantillonnant des surfaces paramétriques connues, ou avec une méthode procédurale) et utiliser la commande *save* pour sauver les points dans un fichier ascii.

### 2. Champs des normales

Ecrire une fonction *bezierPatchNormal* qui pour chaque patch $k$ calcule le champs des normales *N(:,:,:,k) = bezierPatchNormal(P(:,:,:,k),u,v)*,
où u,v sont des vecteurs Matlab de $num\_n$ valeurs de paramètre. Elle prend en entrée les 16 points de contôle et l'ensemble des valeurs de parametre en lesquels les normales $N$ (vecteurs unitaires) sont calculées. En sortie N(:,:,:,k) est une matrice de taille $num\_n$ x $num\_n$ x 3.
La normale de la surface $X$ au paramètre $(\bar{u}, \bar{v})$ est défini par $N(\bar{u}, \bar{v}) = \frac{X_u(\bar{u},\bar{v}) \times X_v(\bar{u},\bar{v})}{\|X_u(\bar{u},\bar{v}) \times X_v(\bar{u},\bar{v})\|}$.

### 3. Interrogation de surfaces

Ces méthodes servent à examiner visuellement la qualité d'une surface graçe à des courbes calculées sur la surface ou graçe aux visualisations des courbures (color plots).

**Exercice —TP2a Lignes isophotes**
Les isophotes sont des lignes d'intensité lumineuse égales sur une surface pour une direction de lumière donnée. Soit $L \in \mathbb{R}^3$ la direction des rayons lumineux parallèls. Une isophote est alors définie par l'ensemble des points $X(\bar{u}, \bar{v})$ dont les paramètres $(\bar{u}, \bar{v})$ vérifient

$$< N(\bar{u}, \bar{v}), L >= c \tag{1}$$

où $c$ est une valeur constante.

Remarque: *Comment tracer les isophotes? La façon la plus simple est d'évaluer la fonction* $I(u, v) =< N(\bar{u}, \bar{v}), L >$ *pour un ensemble assez dense de valeurs* $(\bar{u}, \bar{v})$ *et de faire un plot3 des points* $X^k(\bar{u}, \bar{v})$ *pour lesquels* $|I(\bar{u}, \bar{v}) - c| < \varepsilon$.
*Une autre possibilité est de tracer les isophotes sous forme de courbes en utilsant l'algorithm de "marching cubes" en 2D.*

**Travail demandé:** Pour chacune des 4 surfaces, *surface1,...,surface4*, calculez une dizaine d'isophotes en choisissant une direction $L$ adaptée et en faisant varier la valeur $c$. Visualisez toutes les isophotes sur chacune des surfaces et sauvez le résultat dans un fichier image. C'est un plus si vous calculez des images avec des directions $L$ différentes. *Donnez un nom au fichier image qui fait reference au no. de surfaces et autre parametre.*
**En option:** Perturbez légèrement un ou deux points de contrôle des 4 surfaces et sauvez les points de contrôle dans des fichiers, nommés *psurface1,...,psurface4*. Calculez les isophotes avec les mêmes paramètres (view, $L$, $c$, $\varepsilon$) que pour les surfaces originales, puis sauvez les visualisations dans des fichiers image.

**Exercice — TP2b Curvature plot**
Un plot de courbure associe à chaque point de la surface une couleur $[R, G, B] \in [0, 1]^3$ en fonction d'une courbure de la surface. Généralement on utilise la courbure de Gauss $K(u, v) = \kappa_1(u.v) \cdot \kappa_2(u, v)$ ou la courbure absolue $A(u, v) = |\kappa_1(u, v)| + |\kappa_2(u, v)|$, où $\kappa_1, \kappa_2$ sont les courbures principales de la surface $X$.

**Travail demandé:** Pour chacune des 4 surfaces, *surface1,...,surface4*, calculez les courbures principales en une grille **très dense** de points sur la surface (p.ex. 30x30 points). Visualisez le curvature plot en choisissant une fonction de courbure et une echelle de couleur adaptée que vous affichez à coté du plot de la surface colorée.
**En option:** Utilisez les mêmes surfaces perturbées que dans la question précédente.

**Questions:** Quelles sont vos observations quant aux 2 méthodes d'inspection de surfaces? Quelles propriétés d'une surface les deux méthods sont elles capables (et ne pas ca[ables) de visualiser? (p.ex. ordre de continuit, irrégularités, etc.)

**Travail á rendre**
Le travail est à effectuer par chaque élève **individuellement** au plus tard le

**mercredi 16/11/2018**

Chaque élève dépose sur Teide une archive à son nom *nom.zip* contenant
- les sources (matlab ou autre),
- les 4 fichiers ascii *surface1,..,surface4* ainsi que
- les snapshots des résultats, idéalement sous forme d'un seul document *pdf*, sinon sous forme d'images *png/jpeg*.
- vos observations et reponses pour chacune des 2 méthodes d'interrogation de surface (dans un fichier txt ou pdf.

**Doc Matlab**
La doc de Matlab est online. Il suffit d'écrire *help nom_de_commande* pour avoir la doc sur la commande. p.ex. *help load*. En cliquant ensuite sur le lien "reference page for load" vous êtes redirigé sur le site web du support Matlab.

**Conseil:** Pour calculer les dérivée première et secondes de la surface de Bézier, il est certainement plus facile d'utiliser la représentation cardinale d'un polynôme que la représentation Bernstein-Bézier. Cette représentation cardinale est déjà utilisée dans le programme matlab fourni.

**Rappel:** Les **courbures principales** (voir slides du cours) peuvent être calculées comme les valeurs propres de la matrice $H \cdot G^{-1}$, où $G$ est la matrice de la première forme fondamentale, $G^{-1}$ son inverse, et $H$ la matrice de la deuxième forme fondamentale. Les éléments des matrices $G$ et $H$ sont des simples produits scalaires des dérivée premières et secondes, ainsi que la normale (normée).