TP7 : Surface produit tensoriel - Surface B-splines

L'objectif de ce TP est d'implémenter une méthode permettant la construction de surfaces B-splines. Le TP est à faire en binôme. Le code et le rapport contenant les images et réponses aux questions sera rendu sous forme d'archive *nom1_nom2.zip*. Les algorithmes seront implémentés en C++.

Surface poduit tensoriel de B-splines

Formulation

De même que pour les surfaces de Bézier, on réutilise maximum ce qui est connu pour les courbes B-splines.

Courbes : $d_0, \ldots, d_n$, k degré et $t_0, \ldots, t_{m+k+1}$.

Surfaces :

(d_{ij}) avec $i = 0, \ldots, m$ et $j = 0, \ldots, n$

k le degré en u et l le degré en v

$u_0, \ldots, u_{m+k+1}$ les noeuds en u

$v_0, \ldots, v_{m+k+1}$ les noeuds en v

Les surfaces B-splines sont évaluées en $(u, v) \in [u_{i_0}, u_{i_0+1}] \times [v_{j_0}, v_{j_0+1}]$ par le produit tensoriel ci-dessous.

$$S(u,v) = \sum_{i=i_0-k}^{i_0} \sum_{j=j_0-l}^{j_0} d_{ij} N_i^k(u) N_j^l(v) \tag{1}$$

Evaluation de surface de B-splines

On a deux algorithmes géométrique pour évaluer cette surface de Bézier.

Evaluation 1

On fixe d'abord j et on fait varier i.

$$S(u,v) = \sum_{j=j_0-l}^{j_0} N_j^l(v) \underbrace{\left[\sum_{i=i_0-k}^{i_0} d_{ij} N_i^k(u) \right]}_{=d_j(u)} \tag{2}$$

$d_j(u)$ défini une courbe B-spline en u que l'on peut évaluer par DeBoor-Cox.

$$S(u,v) = \sum_{j=j_0-k}^{j_0} d_j(u) N_j^l(v) \tag{3}$$

L'équation (3) définit une courbe B-spline en v dont les points de contrôle $d_j(u)$ dépendent de u. Au final il y aura :

$$\begin{cases} l+1 \text{ évaluation de DeBoor-Cox de degré } k \\ 1 \text{ évaluation de DeBoor-Cox de degré } l \end{cases}$$

Evaluation 2

Le second algorithme consiste tout simplement à d'abord fixer i et faire varier j.

Travail demandé

1. En utilisant le travail effectué sur les courbes B-splines, implémenter une fonction calculant une surface B-spline en (u, v) pour un ensemble de points de contrôle $d_{i,j}$ et de noeuds fournis.

2. Visualiser la surface à l'aide de gnuplot.

3. Construiser un tore et une sphère à l'aide de votre programme. Comment pouvez-vous assurer la fermeture de votre surface ?

Note :

1. Pour vous permettre de vous concentrer sur l'algorithme en lui-même une structure permettant de gérer les points de contrôle et les noeuds vous est fournie ainsi que quelques fonctions facilitant l'affichage des données. Vous êtes libre de l'utiliser ou pas.

2. N'oubliez pas d'utiliser des `Vector3d` dans vos anciennes implémentations de DeBoor-Cox.