TP3 : Subdidivision

1. Algorithme de de Casteljau - subdivision

L'algorithme de de Castejau que vous avez dejà programmé pour évaluer une courbe de Bézier, peut aussi être utilisé pour subdiviser cette courbe, et ainsi la visualiser plus efficacement. Programmer d'abord une fonction effectuant un pas de subdivision pour une courbe de Bézier, puis une fonction effectuant i pas de subdivision récursivement.

2. Petit complément de cours

Nous défini en CTD les splines, par le degré n, le vecteur de noeud, et le choix des points de controle correspondants à la valeur de la floraison évaluée en n valeurs consécutives du vecteur de noeuds. L'algorithme de de Boor permet de trouver des points sur la courbe (mais vous ne m'implémentez pas, faute de temps). Une autre façon de déterminer une courbe spline correpondant aux points de controle est d'utiliser une algorithme de subdivision. Un tel algorithme peut etre défini de façon naturelle et pratique quand le vecteur de noeuds est uniforme, i.e. régulier. On prend donc comme vecteur de noeud Z (l'ensemble des entiers relatifs). L'algorithme de subdivision consiste à passer des points de controle $(P_i)_{i \in Z}$ correspondants au vecteur de noeud Z aux points de controle $(\tilde{P}_i)_{i \in Z/2}$ correspondants au vecteur de noeuds Z/2. Il y en a environ deux fois plus. Les coefficients de dépendent pas de i car le vecteur est uniforme.

Cet algorithme de subdivision pour le degré n peut se décomposer en 2 étapes. La première étape est une étape de génération de points, et consiste à doubler le nombre de points simplement en répétant chaque point :

$$(\dots, P_{-1}, P_0, P_1 \dots) \text{ donne } (\dots, P_{-1}, P_{-1}, P_0, P_0, P_1, P_1 \dots)$$

La seconde étape, qui est répétée n fois consiste à remplacer l'ensemble des points par les milieux de 2 points consécutifs. On récupère donc environ le même nombre de points.

$$\begin{array}{ccccccccc}
\dots & P_0 & & & P_0 & & & P_1 & \dots \\
\frac{1}{2} & & \frac{1}{2} & \frac{1}{2} & & \frac{1}{2} & \frac{1}{2} & & \frac{1}{2} \\
\swarrow & & \searrow & \swarrow & & \searrow & \swarrow & & \searrow \\
\frac{1}{2} & & \frac{1}{2} & \frac{1}{2} & & \frac{1}{2} & \frac{1}{2} & & \frac{1}{2} \\
\searrow & & \swarrow & \searrow & & \swarrow & \searrow & & \swarrow \\
& \tilde{P}_0 & & & \tilde{P}_{\frac{1}{2}} & & & \tilde{P}_1 &
\end{array}$$

3. Courbes splines de degré arbitraire

3.1. Courbes fermées

Implémenter une fonction qui pour un polygone de contrôle fermé (points de contrôle cliqués à la souris) et un degré choisi, dessine la courbe spline uniforme définie par ces points. Pour cela, il faut implémenter une fonction subdivise qui prend en paramètres le degré et le nombre d'itérations de la subdivision.

3.2. Courbes ouvertes

Implémenter le même algorithme pour les courbes ouvertes. Vous observez que la courbe 'diminue' aux bords. Proposer une solution 'ad hoc' pour pouvoir maintenant définir jusqu'au bord des courbes splines ouvertes.

3.3. Pour vous avancer pour le projet : Surfaces splines fermées en produit tensoriel

Pour le projet, vous programmerez la subdivision pour les surfaces B-spline uniformes fermées en produit tensoriel et des exemples illustratifs. Notez que la topologie d'une surface en produit tensoriel est celle du tore.