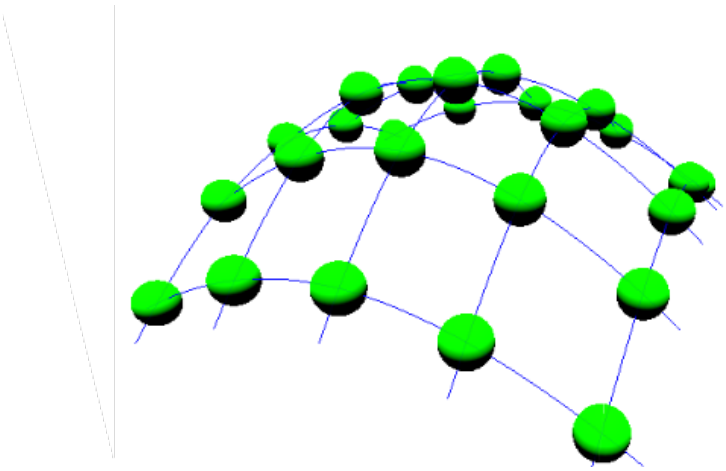
TP1 - Modélisation Géométrique

Interpolation Polynomiale

Julien Desvergnes, Géraldine Morin



Ce TP illustre la partie de cours sur l'interpolation polynomiale.

Table des matières

1 Installation et présentation rapide de Unity

1.1 Installation

Pour installer Unity3D, rendez-vous sur le site https://store.unity.com et téléchargez le UnityHub. Une fois ceci fait, il faut télécharger un éditeur Unity (en gros une version). La version à choisir est 2109.3.5f1 si elle est disponible. Le cas échéant, télécharger la version la plus récente.

Pour vérifier que tout s'est bien passé, essayez de créer un nouveau projet en faisant :

$$Project >> NewProject >> 3D >> CreateProject$$

1.2 Prise en main de l'interface

Je vous encourage à essayer de jouer un peu avec Unity, c'est à dire créer des objets, les déplacer, utiliser la caméra, faire des petites bidouilles. Cela vous permettra de ne pas être perdus lors des premiers TP.

Vous trouverez une petite vidéo qui résume le fonctionnement de l'interface ci-dessous.

`https://unity3d-dev.com/cours/les-bases-dunity-3d-et-du-langage-c/lessons/linterface-de-unity/`

2 Préparation pour le TP1

Les sources de ce TP sont à récupérer sur Google Drive :
`https://drive.google.com/open?id=11irLeQa7cpAKW0dRgvURCUbrbSjJZ-Ug`

Ensuite, pour ouvrir le projet dans UnityHub, aller dans la section *Projects* et cliquer sur *Open* puis choisir le dossier *TP1* comme racine.

Suite à cela le projet devrait être importé dans Unity, vous pourrez alors le lancer.

3 Descriptif du projet

3.1 Scène

Dans la scène, il y a trois sections :

UI : La partie UI (*User Interface*) regroupe des éléments liés à l'affichage textuel. **Vous n'aurez pas à modifier cette partie**

Setup : Des données de mise en place, caméra, lumière, etc. **Vous n'aurez pas à modifier cette partie**

TP1 : C'est ici qu'il y a les éléments spécifiques au TP1. Une structure plus détaillées de cette section est présentée dans les paragraphes suivants.

3.1.1 Interpolateur

Cet objet servira à contrôler les paramètres d'interpolation depuis l'éditeur Unity. Il suffit pour cela de régler les paramètres directement dans la fenêtre d'inspection des objets (*Inspector*)

- **Pas** : Pas de discrétisation des temps d'échantillonnage,
- **Interpolation Type** : Type d'interpolation choisie (Lagrange ou Neville),
- **Subdivision Type** : Type de temps $(T_i)_{i=0..n}$ choisi,
- **Donnees *(Ne pas modifier !)*** : Récupère les points cliqués par l'utilisateur,
- **Text *(Ne pas modifier !)*** : Texte qui peut s'afficher dans le UI.

3.1.2 Donnees

Cet objet contient les points cliqués par l'utilisateur. **Il n'est pas à modifier !**

3.1.3 Point

Cet objet décrit ce qu'est un point qui sera instancié sur un clic. **Il n'est pas à modifier !**

3.1.4 Interpolateur de surface

Cet objet permet de réaliser l'interpolation d'une surface définie par un ensemble de points.

Il contient un exemple de surface : plusieurs points dont les coordonnées sont contenues dans des listes X, Y et Z qui définissent la surface dans l'espace.

3.2 Assets » Scripts » Utils

C'est dans cette section que vous trouverez les fichiers à modifier.

Les deux seuls fichiers de travail sont :

- `Interpolateur.cs`
- `InterpolateurDeSurface.cs`

4 Travail à faire

4.1 Cas fonctionnel : Interpolation de points

Étant données $n+1$ points $P_i = ((x_i, y_i))_{i=0}^{n}$ du plan, dessiner la fonction polynomiale f de degré n telle que

$$f(x_i) = y_i, i = 1...n$$

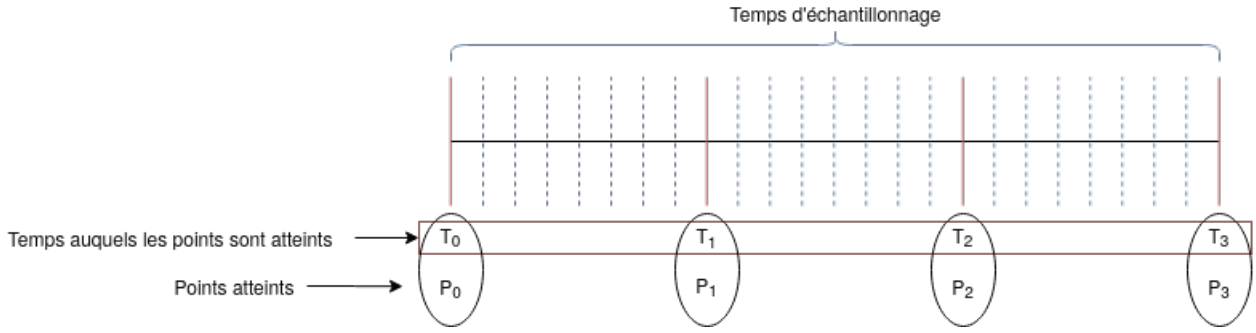
En pratique : Compléter le code de la fonction *lagrange* (fichier *Interpolateur.cs*) afin de calculer l'ordonnée y du point d'abscisse x appartenant à la courbe qui interpole les points de coordonnées (X,Y).

Tester ses modifications : Dans l'éditeur, s'assurer que le bouton *Gizmos* est bien coché (dans la barre de scène), sélectionner l'objet **interpolateur** dans la hiérarchie, choisir les options **Lagrange** et **None** pour le type d'interpolation et le type de paramétrisation. Une fois ces opérations effectuées, cliquer sur **Play**. Pour placer des points, il suffit de cliquer avec la souris ; pour lancer l'interpolation, appuyer sur **Entrée**.

4.2 Cas paramétrique : Interpolation de points

Étant donnés $n+1$ points $P_i = ((x_i, y_i))_{i=0}^n$ du plan, et des paramètres (t_i) pour $i \in [\![0\ ;\ n]\!]$, dessiner la courbe paramétrique F polynomiale de degré minimum telle que $F(t_i) = (x_i, y_i)$ pour tout $i \in [\![0\ ;\ n]\!]$.

En pratique : Compléter le code de la fonction ***buildParametrisationReguliere*** qui doit construire à la fois le vecteur des temps T d'évaluation et le vecteur d'échantillonnage de ces mêmes temps. Compléter également le code de la fonction ***applyLagrangeParametrisation***.



Dans cette figure à quoi correspondent tToEVal et T ?

Tester ses modifications : Dans l'éditeur, s'assurer que le bouton Gizmos est bien coché (dans la barre de scène), sélectionner l'objet **interpolateur** dans la hiérarchie, choisir les options **Lagrange** et **Reguliere** pour le type d'interpolation et le type de paramétrisation. Une fois ces opérations effectuées, cliquer sur **Play**. Pour placer des points, il suffit de cliquer avec la souris ; pour lancer l'interpolation, appuyer sur **Entrée**.

4.3 Influence des paramètres T_i

Pour mesurer l'influence du choix des T_i sur les courbes, on peut essayer de prendre des valeurs de T_i dépendantes de la distance entre les points, de la racine carrée de la distance entre les points, ou encore des valeurs de Tchebychev qui ont été vues en cours.

En pratique : Ici il faut coder les méthodes :
- ***buildParametrisationDistance***
- ***buildParametrisationRacineDistance***
- ***buildParametrisationTchebycheff***

Remarque : **les coordonnées des points cliqués sont accessibles via les variables globales X et Y**. Vous pouvez donc récupérer ces points en faisant $X[i]$ ou $Y[i]$

Tester ses modifications : Dans l'éditeur, s'assurer que le bouton Gizmos est bien coché (dans la barre de scène), sélectionner l'objet **interpolateur** dans la hiérarchie, choisir les options **Lagrange** et la paramétrisation (choix des T_i) voulue. Une fois ces opérations effectuées, cliquer sur **Play**. Pour placer des points, il suffit de cliquer avec la souris ; pour lancer l'interpolation, appuyer sur **Entrée**.

4.4 Interpolation paramétrique via l'algorithme de Neville

Dans cette partie, il faut programmer le calcul de l'interpolation par l'algorithme de Neville.

En pratique : Ici, il faut coder les deux méthodes restantes du fichier *Interpolateur.cs* : ***neville*** et ***applyNevilleParametrisation***.

Tester ses modifications : Dans l'éditeur, s'assurer que le bouton Gizmos est bien coché (dans la barre de scène), sélectionner l'objet **interpolateur** dans la hiérarchie, choisir les options **Neville** et la paramétrisation voulue. Une fois ces opérations effectuées, cliquer sur **Play**. Pour placer des points, il suffit de cliquer avec la souris ; pour lancer l'interpolation, appuyer sur **Entrée**.

4.5 Bonus : surface interpolantes en produit tensoriel

On rappelle qu'une surface en produit tensoriel est définie par :

$$S(s,t) = \sum_{i=0}^{n}\sum_{j=0}^{m} P_{ij}L_i(s)L_j(t) = \sum_{i=0}^{n} L_i(s)P_i(t)$$

avec $n+1$ points $P_i(t)$ pris sur $n+1$ courbes (à t fixé, pour $i \in [\![0\ ;\ n]\!]$) :

$$P_i(t) = \sum_{j=0}^{m} P_{ij}L_j(t)$$

En vous basant sur le script *Interpolateur.cs* que vous avez écrit et sur vos connaissances sur l'interpolation, vous devez générez des surfaces interpolantes en produit tensoriel sur une grille de points de contrôle régulière (c'est à dire, les points de contrôles dans une matrice).

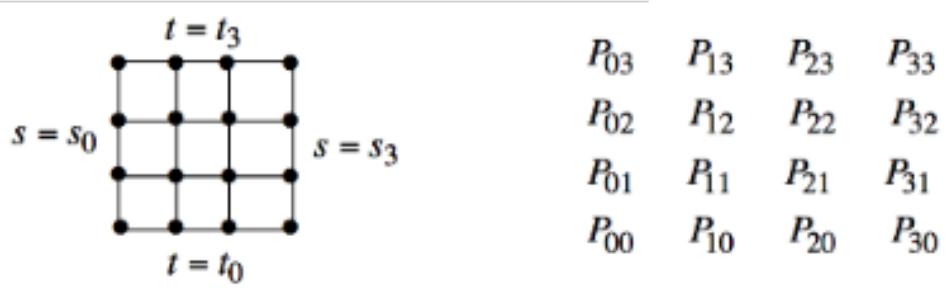


Figure 1 – Domaine de paramétrisation et grille 4*4 de points de contrôle 3D pour une surface de bi-degré(n, m) = (3,3)