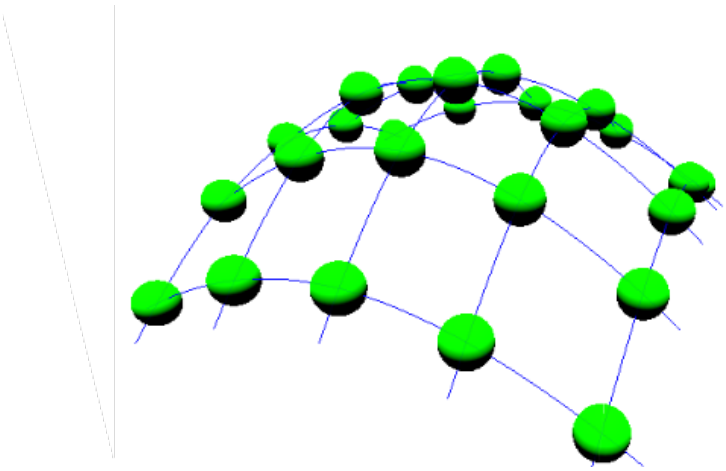
TP2 - Modélisation Géométrique

Analyse de la Courbure et Calcul des k Plus Proches Voisins pour les Maillages 3D

Marie Pelissier, Géraldine Morin



Ce TP illustre la partie de cours sur les modèles discrets.

Table des matières

1 Préparation pour le TP2

Les sources de ce TP sont à récupérer sur la page moodle de la matière :

Département Sciences du Numérique >> 2ème année FISE >> Ensemble des UEs de S8 >>

N8EN06 - UE Modélisation Géométrique et EDP >> N8EN06A - Modélisation Géométrique

Le langage choisi pour ce TP est le Pyhton. Ce TP se présente sous la forme de notebooks et de fichiers pythons à compléter.

Prise en main du code : Nous vous conseillons utiliser le logiciel Visual Studio Code pour visualiser et compléter les codes fournis. Voici les étapes à suivre pour avoir un bon environnement de travail :

- Ouvrir le dossier associée au TP2 dans Visuel Studio Code, ce dernier contient 5 fichiers.
- Installer Python (si nécéssaire) : choisir le logo Extensions présent sur la barre latéral de l'interface, puis rechercher Python. Une fois la page du langage python ouvert, cliquer sur *Install*.
- Ouvrir le premier notebook intitulé : *TP2_curvature.ipynb*
- Vous devez initialiser un environnement de travail, autrement dit, choisir un *Kernel*. Pour faire cette sélection, cliquez en haut à droite sur *Select Kernel*, puis *Python Environnements* et choisir la version */bin/python*3.8 de python (ou ultérieure).
- Dans la **première cellule** du notebook, deux ligne de commande sont commentées. Décommentez-les et exécutez la cellule, cela va lancer l'installation des librairies pythoon : tqdm et trimesh.

Si vous rencontrez des problèmes avec le notebook, une première solution a tester est de *uUinstall/Install* pyhton et jupyter dans visuel studio.

Fichiers disponibles :

- *tp2_curvature.ipynb* à remplir,
- *utils_tp2_curvature.ipynb* contient les fonctions pour effectuer une ACP : *make_pca* ainsi qu'une résolution au moindre carré : *lstsq_quadrics_fitting*,
- *TP2_curvature.ipynb* le notebook avec le code principal. Il faut **adapter la variable *absolute_path*** qui doit correspondre au chemin absolue du dossier du TP2,
- un dossier *models3D* contenant les fichiers obj de maillages 3D,
- un dossier *verite_terrain* contenant des fichiers pkl pour vérifier l'implémentation des fonctions sur les voisinages,
- un dossier *sorties* qui contiendra les fichiers obj des vos maillages colorés en fonction de leurs courbures.
- un pdf *Meynet_2019.pdf* correspondant aux travaux dont nous allons nous inspirer la formule du calculs de la courbure pour un modèle 3D discret.

2 Approximation de la courbure moyenne et de son résidus

Objectif : L'objectif de ce TP est d'implémenter des fonctions pour analyser la courbure des surfaces représentées par des maillages 3D à l'aide des k plus proches voisins de chaque sommet dans ces maillages. L'approximation de la courbure qui a été choisie est celle explicitée dans le papier de Meynet 2019. L'erreur résiduelle obtenue lors de cette approximation en chaque sommet sera également calculer.

Courbure selon Meynet 2019 : *To estimate the mean curvature at a point p, we proceed by local least squares fitting of a quadric surface. First we estimate an approximate tangent plane using Principal Component Analysis, which gives us an orthonormal frame* (ux, uy, uz) *such that u z is aligned with an approximate normal to the surface. We take p as the origin of the coordinate system. In this local frame, the neighbour* p_i *of p has coordinates* (xi, yi, zi)*. We thus look for the quadric surface* $Q(x, y) = ax2 + by2 + cxy + dx + ey + f$ *minimizing :*

$$\sum_i ||z_i - Q(x_i, y_i)||^2 \tag{1}$$

The mean curvature can then be directly estimated from the derivatives of Q that are expressed easily by its coefficients :

$$Curv(p) = \frac{(1 + d^2)a + (1 + e^2)b - 4abc}{(1 + e^2 + d^2)^{3/2}} \tag{2}$$

Ainsi, étant donné un sommet p, il est possible de calculer sa courbure mais également son résidus correspondant à l'erreur d'approximation. Autrement dit, ce résidus correspond à la distance entre le sommet et la quatric déterminée, comme l'illustre la figure ci-dessus par le trait magenta. Ce résidus est calculé par la fonction *lstsq_quadrics_fitting*.

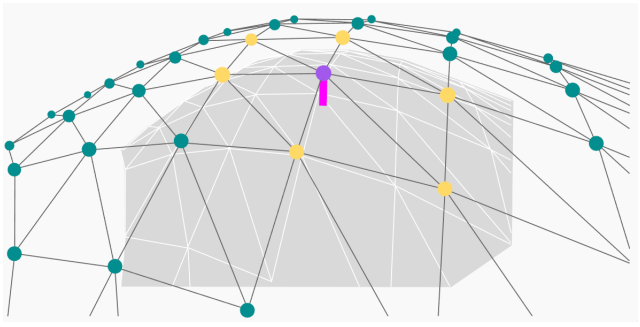


Figure 1 – Illustration du résidus (trait magenta) suite à l'approximation du maillage par une quadric (surface grise)

Pour plus de contexte, voir l'article complet disponible sur moodle.

Instructions :

Voici les fonctions à implémenter dans le fichier *tp2_curvature.ipynb* :

Chargement des Attributs d'un Maillage 3D :

- Utilisez la fonction *load_mesh(path_to_model)* pour charger un maillage 3D à partir d'un fichier spécifié par *path_to_model*.
- Cette fonction retourne un objet Trimesh représentant le maillage ainsi qu'un dictionnaire contenant ses attributs principaux : coordonnées des sommets, nombre de sommet, faces et ses arêtes.

Obtention des Voisins Directs :

- Implémentez la fonction *get_direct_neighbors(nsommets, aretes)* pour obtenir les voisins directs de chaque sommet dans le maillage.
- Cette fonction retourne un dictionnaire où chaque clé est l'indice d'un sommet et la valeur correspondante est un ensemble contenant les indices de ses voisins directs.

Calcul des k Plus Proches Voisins :

- Implémentez la fonction *compute_knn(profondeur_k, nsommet, voisins_directs)* pour calculer les k plus proches voisins de chaque sommet.
- Utilisez la fonction auxiliaire *get_k_neighbors* pour obtenir les k plus proches voisins d'un sommet donné.
- Une implémentation récursive est conseillée.
- Cette fonction retourne un dictionnaire où chaque clé est l'indice d'un sommet et la valeur correspondante est un ensemble contenant les indices de ses k plus proches voisins.

Un peu plus de détails sur l'implémentation de la fonction *get_k_neighbors* :
Cette fonction est une fonction auxiliaire utilisée dans le processus de calcul des k plus proches voisins pour chaque sommet d'un maillage 3D. Son rôle est de trouver **récursivement** les k plus proches voisins d'un sommet donné en explorant les voisins directs et indirects jusqu'à une profondeur spécifiée. Il y a trois cas à prendre en compte relatif à la profondeur :

- Lorsque que la profondeur atteint zéro : retourner l'ensemble des k plus proches voisins actuels,
- Lorsque que la profondeur est égale à 1 : ajouter les voisins directs du sommet (point_index) à l'ensemble n_set,
- Si la profondeur_k est supérieure à 1, la fonction explore récursivement les voisins des voisins jusqu'à la profondeur spécifiée : faire l'appel récursif.

Estimation de la Courbure Moyenne d'un sommet :

- Implémentez la fonction *curvature_meynet* (quadrics) pour calculer la courbure moyenne à partir des coefficients de la meilleure quadrique.

— Cette fonction retourne la valeur de la courbure moyenne calculée au point donné en entrée.

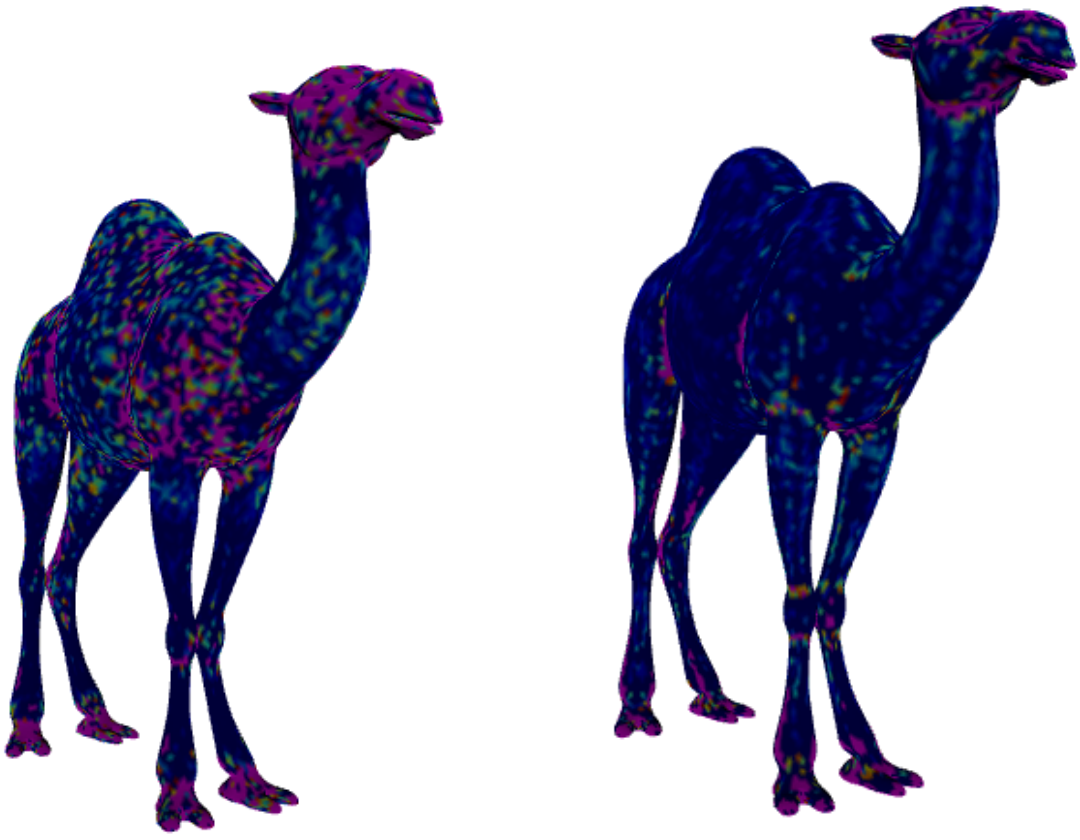
Analyse de la Courbure et du résidus :

— Implémentez la fonction *compute_mean_curvature(sommets, knn)* pour estimer la meilleure quadrique approximant la surface en chaque sommet et approximer la courbure moyenne en chaque point.

— Cette fonction utilise une méthode de régression pour ajuster une quadrique aux points voisins de chaque sommet et calcule la courbure moyenne à partir des coefficients de cette quadrique.

3 Exemple de résultat attendu

Voici les visuels de quelques résultats :



(a) Coloration en fonction de la courbure (b) Coloration en fonction des résidus

Figure 2 – Maillage *camel*