

TDP4 - Surfaces implicites et remplissage de trous dans les maillages

Alexandra Bac

Modélisation géométrique
Polytech Marseille - IRM 4A

Dans ce TP, on souhaite utiliser des surfaces implicites (de type RBF) dans CGAL pour réparer des maillages.

1 Principe

Etant donné un maillage (troué), on identifie tout d'abord le bord du trou. Rien de plus simple : ce sont les arêtes qui ne touchent qu'une face (`is_border` fournit cette information). Puis on étend cet ensemble de sommets en prenant son 2-voisinage : c'est l'ensemble de points sur lequel on va calculer le “rebouchage” du trou.

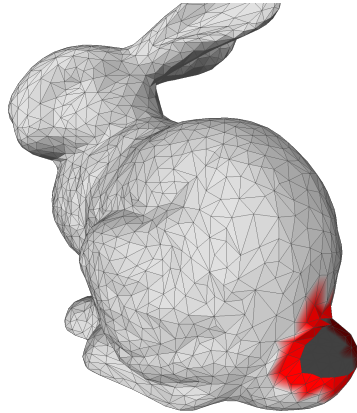


FIGURE 1 – Bord d'un trou (les sommets rouge constituent le 2-voisinage des sommets de bord)

On note $\{X_i\}_{i=1\dots N}$ l'ensemble de ces sommets (on connaît la normale à ces sommets : $\vec{n}_i$).
On souhaite interpoler ces points par une fonction dans la base de fonctions RBF, donc :

$$f(X) = \sum_{i=1}^N \alpha_i \cdot \phi(\|X - X_i\|)$$

où ϕ est la RBF de plaque mince $\phi(r) = r^2 \cdot \log r$. La figure suivante illustre une telle fonction.

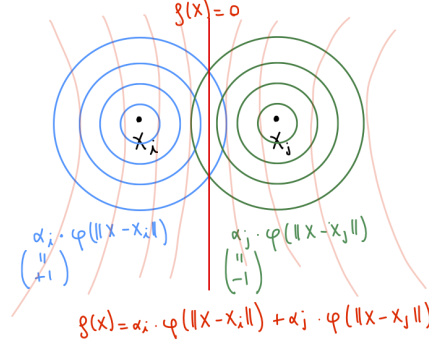


FIGURE 2 – Exemple de fonction à base RBF.

On souhaite construire une fonction (donc une combinaison linéaire des fonctions de base) passant par les points de données. Donc telle que :

$$\forall k \quad f(X_k) = \sum_{i=1}^N \alpha_i \cdot \phi(\|X_k - X_i\|) = 0$$

mais si $\alpha_1, \dots, \alpha_N$ sont solution, on constate que n'importe quels multiples $\lambda\alpha_1, \dots, \lambda\alpha_N$ le sont aussi. Donc ce système est singulier (on cherche une fonction s'annulant en certains points mais en ne précisant rien de ses valeurs non nulles ...).

Pour résoudre le problème, il faut imposer des contraintes "artificielles" qui contraindront les lignes de niveau de la fonction implicite recherchée.

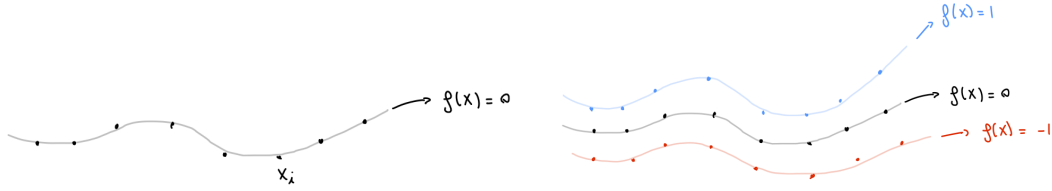


FIGURE 3 – Exemple de points et fonction implicite interpolante : (gauche) problème sur les points initiaux, (droite) contraintes ajoutées pour désingulariser le problème.

Pour ajouter ces points artificiels (contrôlant les lignes de niveau), une solution simple est d'utiliser les normales aux points (puisque elles sont connues grâce au maillage sous-jacent). Pour chaque point X_i on pose donc la contrainte $f(X_i) = 0$ ainsi que les contraintes $f(X_i + \delta\vec{n}_i) = 1$ et $f(X_i - \delta\vec{n}_i) = -1$ (on appelle v_i cette valeur attendue pour la fonction f). Etant donnés nos N points de données, on a donc $3N$ centres (notés c_i) constitués par :

- (i) Les points X_i où $v_i = 0$
- (ii) Les points $X_i + \delta\vec{n}_i$ où $v_i = 1$
- (iii) Les points $X_i - \delta\vec{n}_i$ où $v_i = -1$

Et on souhaite résoudre le problème d'interpolation :

$$f(c_i) = v_i \quad \forall i \in \{1 \dots 3N\}$$

Pour garantir l'existence d'une solution pour $\phi(r) = r^2 \log(r)$ (thin plate spline), il est nécessaire d'ajouter un polynôme de degré 2 à la combinaison linéaire des fonctions de base. Comme $\dim(\mathbb{R}_2[x, y, z]) = 10$, le

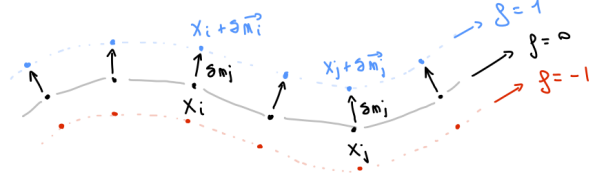


FIGURE 4 – Points et valeurs utilisés pour le problème d'interpolation

modèle avec lequel on souhaite interpoler les données est donc précisément :

$$f(X) = \sum_{i=0}^{3N-1} \alpha_i \cdot \phi(\|X - c_i\|) + \sum_{j=0}^9 \beta_j \cdot P_j(X)$$

où les P_j sont une base de $\mathbb{R}_2[x, y, z]$ ($\{x^2, y^2, z^2, xy, yz, xz, x, y, z, 1\}$ - on pose $d = 10$).

On a donc $3N + d$ inconnues (les $3N$ α_i et les d β_j), pour lesquelles on a $3N + d$ équations :

(i) $f(c_k) = v_k$ pour les $3N$ centres c_k , c'est-à-dire :

$$\sum_{i=0}^{3N-1} \alpha_i \cdot \phi(\|c_k - c_i\|) + \sum_{j=0}^9 \beta_j \cdot P_j(c_k) = v_k$$

(ii) auxquelles on ajoute les d équations :

$$\sum_{i=0}^{3N-1} \alpha_i \cdot P_k(c_i) = 0$$

pour $k = 0 \dots d - 1$

Il s'agit donc d'un système linéaire dont la forme matricielle est :

$$AX = B$$

avec :

$$A = \left(\begin{array}{c|c} \Phi & P \\ \hline P^t & 0 \end{array} \right) \text{ et } B = \begin{pmatrix} \vdots \\ v_i \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

où Φ est une matrice $3N \times 3N$ définie par :

$$\Phi_{ij} = \phi(\|c_i - c_j\|)$$

et P est une matrice $3N \times d$ défini par :

$$P_{ij} = P_j(c_i)$$

En résolvant le système, on obtient les coefficients α_i et β_j du modèle.

Le maillage de la surface implicite peut alors être réalisé par l'algorithme fourni dans CGAL (qui n'est ni du "marching cubes" ni du "dual contouring", mais un algorithme de raffinement de Delaunay contraint : `CGAL::make_surface_mesh` (maillant dans notre cas la surface implicite incluse dans une sphère donnée).

2 A vous

Ce processus a été totalement implémenté et vous commencerez pas télécharger le matériel de TP, le compiler et l'exécuter sur les modèles fournis dans le répertoire `data`. Le `main` se situe dans `hole_fill`.

- (i) On va tout d'abord se concentrer sur la résolution elle-même.
 - (a) Dans la fonction `fill_hole`, quand vous augmentez le troisième paramètre de `compute_approx_matrix` (donc le paramètre `scale` : le δ de $X_i + \delta n_i$ de notre modèle), que se passe-t-il ?
 - (b) L'ajout des centres où la valeur de la fonction est 1 et -1 respectivement permet de contrôler la forme des lignes de niveau autour de la surface recherchée, mais pour limiter la complexité, il n'est peut-être pas nécessaire d'ajouter tous les points $X_i + \delta \vec{n}_i$ et $X_i + \delta \vec{n}_i$. Modifiez la fonction `_f_select` de `Hole_filling` pour qu'elle sélectionne seulement un pourcentage des points. Quel est le résultat sur la surface reconstruite ?
 - (c) Pour estimer la stabilité du système, on souhaite calculer son conditionnement. Rappeler pourquoi celui-ci peut-être obtenu par SVD de la matrice. En Eigen :

```
#include <Eigen/src/SVD/JacobiSVD.h>
JacobiSVD<MatrixXd> svd(A);
svd.singularValues()(0);
svd.singularValues()(...);
```

permettent de calculer puis récupérer les différentes valeurs singulières. Qu'en est-il ici ? Justifier le choix de l'algorithme de résolution de systèmes linéaires de `solve_approx`.

- (ii) On souhaite ensuite finir l'algorithme de réparation du maillage. Si le maillage obtenu à partir de la surface implicite semble bien correspondre aux besoins (et préserver la courbure de la forme), on constate qu'il est nécessaire de le restreindre (au trou à boucher dans le maillage), puis de le reconnecter au maillage.
 - (a) Quelle méthode pouvez-vous proposer pour restreindre le maillage de la surface implicite (de manière à ce qu'il ne "déborde" plus sur le maillage initial). Implémentez.
 - (b) Puis proposez et implémentez une méthode pour "recoudre" ces deux maillages, donc relier le maillage initial au patch issu de la surface implicite.