

Implantation fonctionnelle d'algorithmes géométriques en 3D

Titouan Laurent

1 Introduction

— $vertex \in \mathbf{N}$, l'indice du sommet incident dont elle est issue.

2 Structures de données implémentées

2.1 Données d'entrées

Un *vec3* permet de modéliser aussi bien un point qu'un vecteur dans l'espace 3D. Les composantes sont données dans l'ordre usuel x, y et z ($x, y, z \in \mathbf{R}$).

Dans notre implémentation, on considérera une simple liste ordonnée des n points (sous le nom de la variable globale *GLOBAL_V_LIST*) représentant les données d'entrée du problème à résoudre.

2.2.2 Maillage triangulaire — *dcel*

Une *dcel* permet la modélisation d'un maillage topologique triangulaire. Elle dispose des attributs suivants :

- *he_list*, une liste de *he*.
- *available_he_index* $\in \mathbf{N}$, le prochain indice non attribué à une *he* de la liste.

2.2 Stockage et manipulation de l'enveloppe convexe en mémoire

La structure en liste de demi-arêtes orientées (doubly connected edge list, abrégé DCEL) est parmi les plus communes qu'il soit dès lors que l'on souhaite modéliser un maillage topologique.

Ici on plante une version plus limitée de cette dernière, puisqu'elle permet de créer maillages triangulaires uniquement — ce qui est sans effet sur les capacités du programme final.

2.2.3 Exemple simple en 2D

Considérons le maillage de la Figure ?? . Une structure *dcel* associée valide serait alors telle que celle présentée Figure ??.

Note : si les numéros d'indices choisis pour les demi-arêtes sont plus ou moins arbitraires, la structure *dcel* impose tout de même des contraintes comme nous allons le voir.

2.2.1 Demi-arête — *he*

Une *he* (abréviation de halfedge, ou demi-arête) est une structure de données disposant d'informations sur elle-même et son voisinage topologique. Celles-ci sont les suivantes :

- *index* $\in \mathbf{Z}$, un identifiant unique pour la demi-arête, avec :

$$\begin{cases} \textit{index} = -1 & \text{est une demi-arête nulle} \\ \textit{index} \geq 0 & \text{sinon} \end{cases}$$

- *opposite* $\in \mathbf{Z}$, l'identifiant de la demi-arête opposée, avec :

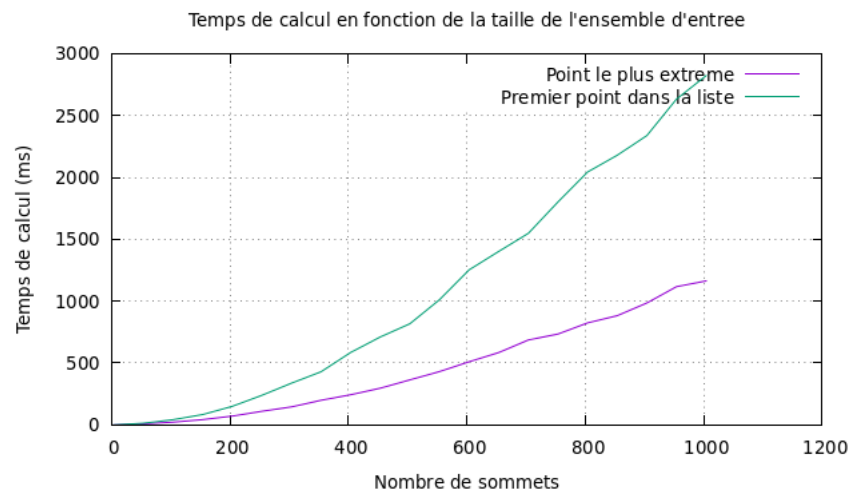
$$\begin{cases} \textit{opposite} = -1 & \text{est une demi-arête du bord} \\ \textit{opposite} \geq 0 & \text{sinon} \end{cases}$$

2.2.4 Ajout et retrait d'une face dans une *dcel*

Les opérations de modification d'une *dcel* sont au nombre de deux (si l'on ne prend pas en compte le constructeur) : l'ajout et le retrait de face.

L'ajout d'une nouvelle face entraîne implicitement la création de trois demi-arêtes (*he*) dans la structure. Leur indice est lui calculé aux lignes 3, 4 et 5. De même, l'indice de leur demi-arête opposée respective (si elle existe) est recherché parmi toutes les demi-arêtes de *dcel* aux lignes 7, 8 et 9. Notez que l'attribut *dcel.available_he_index* est mis à jour dans la structure retournée ligne 29.

3 Évaluation des performances



2