

Détails de la fonctionnalité des codes créés

`_main2.c`

Fichier principal gérant les étapes de la méthode réparti dans plusieurs fichiers de bibliothèques .c
Pour observer la méthode sur une autre géométrie il suffit de changer les fichiers d'ouvertures à la ligne 23, 104 et 110.

`_maillage.c`

Bibliothèque de lecture de fichier .mesh, contient 7 fonctions.

- `nbP` : Retourne le nombre de points d'un fichier .mesh
- `rechercherPoints` : Fonction aidant à la lecture du fichier .mesh pour retrouver le mot clé "Vertices"
- `nbT` : Retourne le nombre de triangles d'un fichier .mesh
- `rechercherTriangles` : Fonction aidant à la lecture du fichier .mesh pour retrouver le mot clé "Triangles"
- `nbE` : Retourne le nombre d'arêtes d'un fichier .mesh
- `rechercherEdges` : Fonction aidant à la lecture du fichier .mesh pour retrouver le mot clé "Edges"
- `remplissage` : Remplissage des tableaux associés aux données du fichier .mesh, tableau de points, triangles et arêtes

`_boite.c`

Construction d'une boîte englobante de la géométrie d'origine. Le fichier ne contient qu'une fonction "construct-Boite". La fonction calcule l'abscisse et l'ordonnée minimale de la position des points de la géométrie et construit une boîte autour de la géométrie en fonction de ses deux données . Il est possible de modifier la taille de la boîte.

`_cercle.c`

Bibliothèque gérant la méthode d'ajout des points de la géométrie suivant le critère du noyau de Delaunay. Elle possède 11 fonctions.

- `test_inscritm` : Vérifie si un point appartient au cercle circonscrit d'un triangle en utilisant une méthode géométrique de résolution. Elle se base sur le calcul du barycentre du système associé aux longueurs des arêtes du triangle ainsi qu'à ses angles.
- `init` : Fonction d'initialisation du tableau des voisins par des valeurs -1 avant son remplissage par la fonction `rtaVoisin`
- `rtaVoisin` : Remplissage du tableau des voisins en recherchant le ou les triangles d'appartenance de chaque arête associée à un triangle. Si une arête n'appartient qu'à un seul triangle on ne fait pas de modification car la fonction "init" à déjà prérempli le tableau avec des -1.
- `verifDansLeTriangle` : Fonction renvoyant si oui ou non (à l'aide d'un booléen) un point appartient à un triangle. Pour cela on calcule les coordonnées barycentriques du point dans ce triangle. Si toutes les coordonnées barycentriques associées sont positives ou nulles alors le point est bien dans le triangle. Dans le cas contraire la fonction remplit un tableau de 3 cases avec des 1 ou -1 selon le signe des coordonnées barycentriques. Cette information est ensuite utilisée dans "QTriangles" pour se déplacer vers un autre triangle suivant la direction indiquées par les coordonnées barycentriques.
- `QTriangle` : Fonction renvoyant le numéro du triangle dans lequel se trouve le point à ajouter. On ne parcourt pas tous les triangles, on suit la direction indiquée par les signes des coordonnées barycentriques renvoyé par "verifDansLeTriangle" et par les informations de `tabVoisin` permettant de savoir où se trouve les bords du domaine.
- `LCirconscrit` : Remplit le tableau des triangles faisant partie de la cavité à supprimer pour l'ajout du nouveau point. La fonction utilise les fonctions "test_inscritm", "rtaVoisin" et "QTriangles".
- `testOrdo` : Permet de définir les points de la cavité connaissant les triangles la composant (information issu de "LCirconscrit").
- `Maj` : Mets à jour la table des triangles en utilisant "testOrdo" pour connaître les triangles à supprimer. On remplace par des -1 les valeurs associées aux triangles à supprimer. Cela permet de redimensionner le tableau en supprimant seulement les éléments du tableau avec des valeurs négatives. Le tableau de triangles étant un tableau 1D comprenant la numérotation des trois points associés à ses sommets.
- `verif` : Vérifie si une arête est déjà présente dans la table des arêtes.
- `newT` : Permet de mettre à jour la table des triangles, des points et des arêtes associée à l'ajout d'un nouveau point. Utilise les différentes fonctions précédentes pour cela.
- `ajout_pt` : Fonction utilisée en dernière étape de la méthode pour qu'une fois que la suppression des triangles de la boîte englobante a été faite il est possible de raffiner le maillage en ajoutant des points

sur les segments du maillage.

_ basculement_arete.c

Fichier ne comportant qu'une fonction "basculement". Permet de basculer deux triangles lorsque l'arête manquante de la géométrie résulte de leur basculement. Fonction ne prenant en compte que le cas où un basculement est nécessaire pour obtenir l'arête manquante. Elle recherche tous les triangles possédant un des points de l'arête manquante et fait de même pour l'autre point. Elle bascule ensuite les deux triangles ayant une arête commune dans ses deux listes.

_ basculement_niveau2.c

Autre méthode de basculement, fonctionne aussi dans le cas d'un seul basculement à opérer. La méthode calcule tous les triangles possédant une arête intersectant l'arête manquante, à chaque fois qu'elle en croise un, elle bascule les deux triangles ayant l'arête commune croisant le segment manquant. Pour cela on utilise une propriété issue des produits vectoriels.

Pour l'instant la méthode n'est pas optimale et possède des défauts lorsqu'il y a plusieurs basculements à faire, cela semble dû au fait qu'il faut opérer un certain ordre sur les basculements (voir "main3.c" pour le test de cette méthode).

_ suppression.c

Bibliothèque gérant la suppression des triangles de la boîte englobante. Elle possède 9 fonctions, 3 fonctions "afficherFile", "defiler" et "enfile" sont utilisées pour gérer le principe de file du langage C et ne sont pas intrinsèques à la méthode. L'utilisation des autres fonctions sont quant à elles détaillée. La méthode se base sur l'identification des différentes zones connexes du maillage après basculement.

- `arete_commune` : Renvoie si oui ou non deux triangles voisins ont leur arête commune sur le bord d'une zone.
- `recherche_zone2` : Recherche la zone de la boîte englobante en sachant que les 4 coins de la boîte englobante sont forcément dans cette zone. On utilise plus spécialement que le premier coin de la boîte identifie cette zone.
- `ref_non_rempli` : Pour gérer les zones connexes on modifie le tableau des triangles en lui rajoutant de la place pour prendre en compte une référence correspondant à l'indice de la zone connexe dans lequel il se trouve. Cette fonction renvoie si tous les triangles sont associés à une référence (les références sont initialisées à zéro) si un triangle n'a pas encore de référence c'est qu'il existe encore une zone connexe et on recherche la zone définie par ce triangle, en l'ajoutant au début d'une nouvelle file des triangles définissant la nouvelle zone.
- `recherche_zone_conserver` : Identifie et retourne la zone à conserver connaissant la zone connexe représentant la boîte englobante. On utilise l'ordonnement des points dans la table des points donnant que le 5^e point est forcément un point du bord de la géométrie et qu'il est en contact avec la zone connexe de la boîte englobante.
- `suppression_triangles` : Fonction principale. Dans un premier temps on identifie les différentes zones connexes, en connaissant la zone de la boîte englobante ("recherche_zone2") on est capable de connaître la zone à garder ("recherche_zone_conserver") et on supprime les zones qui ne sont pas la zone à garder.
- `modification_arete` : Après suppression des triangles des zones connexes à ne pas conserver on met à jour la table des arêtes.

_ ajout_point.c

Fichier ne contenant qu'une fonction "tab_point_ajout" définissant un tableau de point à ajouter en prenant le point milieu des segments plus grand que le segment minimale du maillage. On cherche à réduire la taille des éléments du maillage les plus grossiers. On utilise cette information dans "main2.c" en ajoutant les points par l'utilisation de la fonction "ajout_pt" de la bibliothèque "cercle.c".

_ qualite.c

Fichier contenant trois fonctions "taille", "aire" et "TabQuali" permettant de remplir un tableau avec la qualité associée aux éléments du maillage (utilisé ensuite pour afficher un histogramme).