

# Génération de maillages

MAP-SIM2

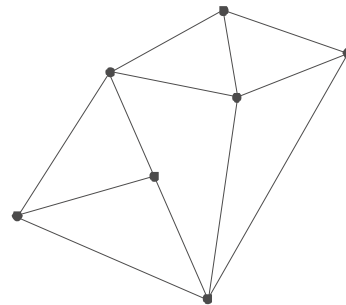
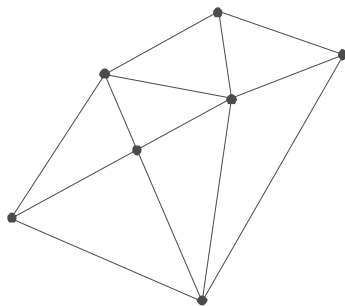
2017

On cherche à réaliser dans ce projet des triangulations de domaines bidimensionnels. Les algorithmes de maillage d'un domaine quelconque sont assez complexes, c'est pourquoi on s'intéressera à des géométries particulières dans une première étape puis on abordera le cas de géométrie un peu plus complexe en utilisant une méthode dite frontale. On devra disposer d'un outil de représentation des maillages.

## 1 Triangulation

Rappelons qu'une triangulation  $\mathcal{T}$  d'un domaine polygonal est un ensemble de triangles ( $T_l$ ) vérifiant les propriétés suivantes :

1.  $\forall l, \overset{\circ}{T}_l$  (intérieur de  $T_l$ ) est non vide
2.  $\forall l, l' \overset{\circ}{T}_l \cap \overset{\circ}{T}_{l'} = \emptyset$
3. toute arête d'un triangle est soit l'arête d'un autre triangle soit une arête du bord



ne satisfait pas (3)

Pour décrire une triangulation, il suffit de disposer

- d'une liste des sommets de la triangulation (coordonnées)
- d'une liste des triangles (triplet des numéros de sommet du triangle)

Il peut être utile dans le contexte d'un calcul par éléments finis par exemple de disposer d'autres informations

- liste de triangle définissant un sous-domaine du domaine
- liste d'arête de triangles définissant une partie du bord ou une frontière "intérieure" du domaine

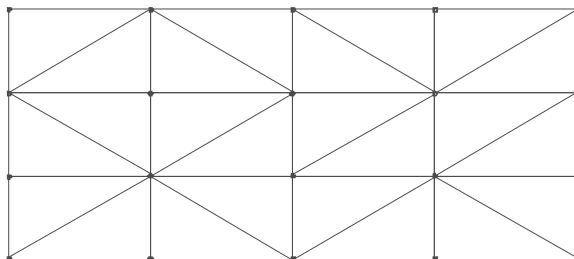
Afin de manipuler le maillage, on créera les classes **sommet** (coordonnées), **triangle** (triplet de sommets), **arête** (triangle (pointeur) et n° d'arête), **domaine** (dimension, nom et liste de triangles (pointeur) ou d'arêtes (pointeur)) et une classe **triangulation** qui sera constituée de la liste de tous les sommets, de la liste de tous les triangles, de la liste des arêtes, de la liste des domaines (pointeur). On adjoindra à ces classes toutes les fonctionnalités dont on aura besoin, en particulier les algorithmes de maillage seront des fonctionnalités de la classe

## 2 Triangulations particulières

On s'intéressera dans un premier temps au maillage de domaines simples : rectangle et secteur angulaire ainsi qu'à des transformations géométriques de tels domaines.

### 2.1 Maillage s'un rectangle

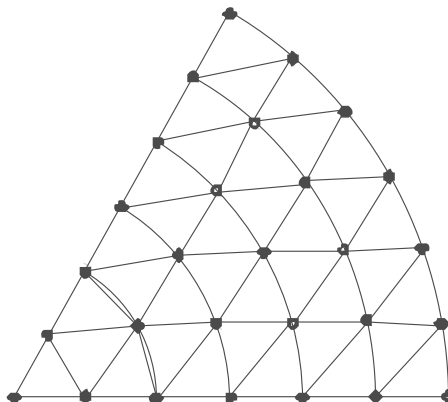
Le maillage d'un rectangle repose sur le découpage en rectangle élémentaire du rectangle, chaque rectangle élémentaire étant ensuite coupé aléatoirement en 2 triangles.



On envisagera le cas régulier (pas constants suivant  $Ox$  et  $Oy$ ) et le cas non régulier (pas variables suivant les deux directions). Dans ce dernier cas il faudra se donner une liste d'abscisses et d'ordonnées.

### 2.2 Maillage d'un secteur angulaire

Le maillage d'un secteur angulaire est un peu plus compliqué.



A partir du nombre de points  $N$  sur l'arc du secteur angulaire, on construit les  $N$  arcs intérieurs répartis régulièrement. Sur l'arc le plus proche de l'arc extérieur on pose  $N - 1$  points équirépartis, sur l'arc suivant  $N - 2$  points et ainsi de suite. On crée ensuite la triangulation naturelle associée à ces points. Lorsque le secteur angulaire à un angle supérieur à  $\pi/2$ , il conviendra de découper le secteur en plusieurs secteurs angulaires (par exemple pour le maillage d'un demi-disque on pourra mailler 2 quarts de disque puis fusionner les deux maillages).

### 2.3 Outils généraux

On écrira un outil de fusion de maillage qui fusionnera les sommets qui ont les mêmes coordonnées (à  $\varepsilon$  près) et mettra à jour la liste fusionnée des sommets et la liste fusionnée des triangles.

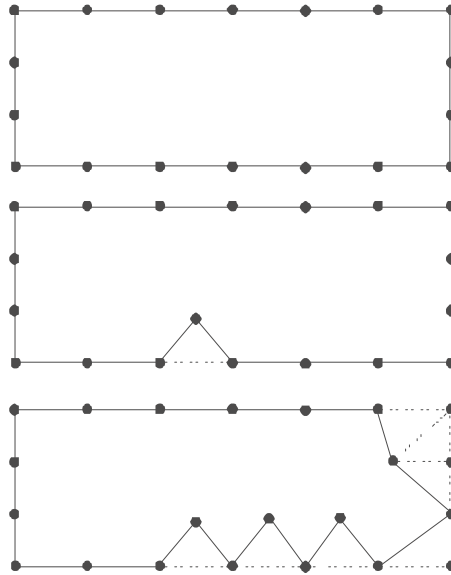
En appliquant une transformation géométrique  $F : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  aux coordonnées du maillage, on pourra générer des maillages d'autres objets (ellipse, rectangle déformé,...). On prévoira donc d'inclure cette possibilité d'appliquer une transformation géométrique (translation, rotation, homotétie, ...). En particulier, en utilisant des transformations de type symétrie et la fusion de maillage on pourra générer des maillages par technique de symétrisation.

## 2.4 Représentation graphique

Pour faire les représentations graphiques des maillages on pourra utiliser Matlab qui permet de visualiser un maillage à l'aide de la fonction **trimesh**. On ajoutera des fonctionnalités supplémentaires dans le code Matlab pour afficher les n° des sommets et les n° des triangles. Il sera nécessaire de disposer d'une fonctionnalité (attachée à la triangulation) générant un fichier contenant le maillage, fichier qui sera relu dans le code Matlab.

## 3 Maillage de domaines quelconques

Il existe plusieurs méthodes pour générer une triangulation d'un domaine quelconque. Nous proposons d'utiliser la **méthode frontale**. On supposera par la suite que le bord du domaine est constitué de segments, eux mêmes découpés en petits segments (données du maillage). Ce découpage sert à définir la finesse du maillage. La méthode frontale consiste à partir du bord du domaine, le front initial, alors constitué d'une liste de segments, à générer des triangles de proche en proche en introduisant des nouveaux points ou en utilisant des points existants. Chaque création de triangle induit une modification du front



On n'introduit pas nécessairement de nouveaux points car il peut y en avoir des points du front qui conviennent. Ce procédé soulève plusieurs questions :

1. comment choisir l'arête du front qui va être traitée à chaque itération ?
2. comment trouver un point qui convient ?
3. comment en créer un nouveau ?
4. comment se termine l'algorithme ?

Voici les solutions couramment adoptées :

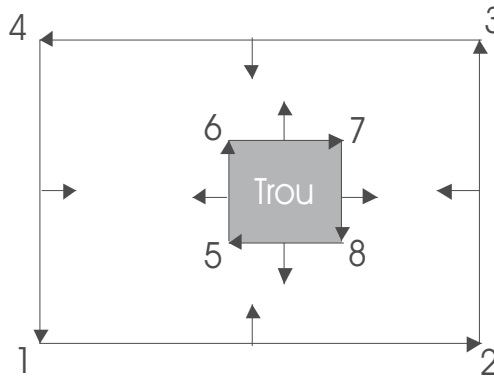
1. on choisit l'arête la plus petite
2. on accepte un point proche s'il permet de construire un triangle de bonne qualité :  
rapport de diamètre du cercle circonscrit et du diamètre du cercle inscrit inférieure à une valeur donnée.  
ou critère de Delaunay : *tout cercle circonscrit à un triangle ne contient aucun sommet en son intérieur.*

3. on crée un nouveau point en faisant un triangle "idéal", typiquement un triangle équilatéral. Attention, il faut placer le point du bon côté du front, c'est-à-dire dans le domaine. Il faut donc avoir une notion d'orientation des segments! Il faut également vérifier que le point ajouté n'est pas déjà dans un triangle déjà construit.
4. On démontre qu'au bout d'un nombre fini d'itération l'algorithme s'arrête (faux en dimension 3). Cela arrive dès que le front est vide.

Il est possible d'obliger le maillage à utiliser certains points intérieurs donnés. Il suffit, lors de la recherche d'un nouveau point, pour construire le triangle d'aller chercher également dans cette liste de points supplémentaires. Dans tous les cas, il faudra veiller à ce qu'aucune arête du nouveau triangle n'intersecte le front et à ce que le nouveau triangle ne contienne pas un point de la liste de points supplémentaires. Attention, le front peut se diviser; c'est le cas lorsque deux parties du front se rencontrent. Dans cette situation, il faut prévoir de gérer des fronts multiples!

Concrètement, on pourra développer une classe **front** qui maintiendra une liste d'**arête** triée selon leur taille (classe **map** de la STL par exemple) pour stocker le front, une liste des **points** qui peuvent être utilisés pour générer un nouveau triangle et de toutes les méthodes utiles à la génération du maillage :

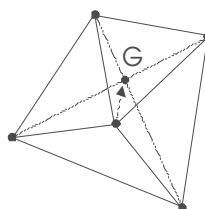
- un outil permettant de générer la liste des arêtes orientés du bord.; le plus simple consiste à partir d'une description du bord déjà orienté de telle sorte que la normale (rotation de  $\pi/2$ ) du vecteur directeur du segment pointe toujours dans le domaine.



- un outil permettant de savoir si un point est dans un triangle (méthode de la classe **triangle**)

Cet algorithme est assez coûteux, il faut donc veiller à optimiser les calculs, en particulier la recherche de points proches et l'admissibilité d'un point.

Les mailleurs utilisant cette méthode frontale incorporent de nombreuses améliorations. En particulier, une fois le maillage construit ils utilisent des algorithmes de régularisation pour améliorer la qualité du maillage, ces algorithmes se contentent de déplacer les sommets du maillage sans ajouter de nouveaux sommets. Un des plus simples consiste à se placer sur un sommet du maillage et à le déplacer au barycentre des sommets voisins (sommets appartenants aux triangles ayant le point à déplacer pour sommet). On applique le procédé à tous les sommets du maillage 2 à 3 fois.



On validera chacune des fonctionnalités élémentaires puis l'algorithme entier sur des petits cas dans des situations très simples (carré par exemple) et enfin on réalisera des essais sur des cas plus gros (millier de triangles).

## 4 Organisation du travail (3p)

Afin de mener un développement efficace, on pourra se partager le travail de la façon suivante

- Dans un premier temps, on définira collectivement les classes générales **point**, **triangle**, **maillage**
- Un élève se chargera du développement de ces classes. Elles devront être opérationnelles très rapidement. Il devra également développer les outils de visualisation et les outils de fusion et symétrie de maillage.
- En parallèle, un autre élève prendra en charge la conception et le développement des outils de maillage de domaines simples, tandis que le dernier s'occupera du développement de la méthode frontale (partie la plus difficile).

Evidemment, il s'agit de conduire le développement dans un mode collaboratif et donc en ayant des discussions sur certains choix.