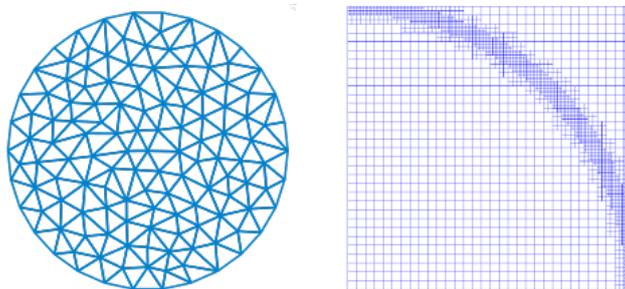


### Informatique S6 - TP3: Templates et classes

**Durée** : trois séances.

**Exercice 1 (Géométrie IV).** Dans cet exercice on s'intéresse aux maillages. Il s'agit de couper une géométrie en un ensemble de petites formes géométriques simples appelées "mailles" (triangles, quadrangles, polygones). Les maillages sont très utiles en animation 3D ou en simulation physique car ils permettent de représenter une géométrie (ensemble continu de point) par un nombre fini de données. Exemple : Un maillage est défini par :



- un tableau de noeuds (type point, taille : nombre de points),
- un tableau de mailles ( triangle ou quadrangle, taille : nombre de mailles),
- un tableau de noeuds de la maille. Il contient pour chaque maille les numéros noeuds qui la composent.
- un tableau de voisin. Il contient pour chaque noeud les numéros des mailles voisines.
- une taille caractéristique  $h$  (un double).

La construction du tableau de voisin ne sera pas traitée dans ce TP.

**Question.** Écrire une classe "maillage" avec "template". Le template sera défini par deux paramètres : le type de maille (triangles ou quadrangles) et un entier définissant un nombre maximum de voisins. Écrire un constructeur par défaut, par copie et le destructeur. On ajoutera un constructeur qui prend le nombre de mailles et le nombre de noeuds et qui initialise l'objet avec.

**Question.** Surcharger l'opérateur "=" ainsi que l'opérateur ">>" qui affichera : le nombre de mailles, de noeuds, le nombre max de voisins et le type de maille.

**Question.** Écrire une méthode "maillage\_regulier" qui crée un maillage régulier (on remplit les tableaux des mailles, des noeuds, des noeuds de chaque maille. Pour les quadrangles :

- on choisira un nombre de mailles  $N^2$ , un nombre de sommet  $(N + 1)^2$ ,  $\Delta = 1/N$ ,
- on construira les points par

$$\begin{cases} x_k = i * \Delta \\ x_k = j * \Delta \end{cases}$$

avec  $k = iN + j$  avec  $i = \{0, N\}$ ,  $j = \{0, N\}$ . On utilisera la boucle sur "i" comme boucle extérieure ( première) et celle sur "j" comme boucle intérieure.

- On construit les mailles (indexées de zéro à  $N^2 - 1$ ). Pour la maille  $k$  on prend les points :

$$k + k/N, k + k/N + 1, k + N + 1 + k/N, k + N + 2 + k/N$$

Pour les triangles on va procéder comme pour les quadrangles puis couper en deux :

- on Choisira un nombre de mailles  $2(N^2)$ , un nombre de sommets  $(N + 1)^2$ ,  $\Delta = 1/N$ ,

- on construira les points par

$$\begin{cases} x_k = i * \Delta \\ x_k = j * \Delta \end{cases}$$

avec  $k = iN + j$  avec  $i = \{0, N\}$ ,  $j = \{0, N\}$ . On utilisera la boucle sur "i" comme boucle extérieure ( première) et celle sur "j" comme boucle intérieure.

- On construit les mailles (indexées de zéro à  $2(N^2) - 1$ ). Pour les mailles  $k$  et  $k+1$  on prend les points :

$$k + k/N, k + k/N + 1, k + N + 1 + k/N$$

et

$$k + k/N + 1, k + N + 1 + k/N, k + N + 2 + k/N$$

**Question.** Écrire une méthode qui calcule la taille caractéristique  $h$  comme le minimum des tailles caractéristiques de toutes les mailles. La tester sur les maillages précédents.

**Question.** Écrire une méthode qui remplit le tableau des voisins.

**Exercice 2 (Matrices).** On considère les matrices de type  $\mathcal{M}_{n,m}(\mathbb{K})$ . Il s'agit donc des matrices à  $n$  lignes et  $m$  colonnes et à coefficients dans  $\mathbb{K}$ . On propose d'écrire un template de classe pour gérer les matrices. On propose ici un stockage de type COO. Il s'agit d'un stockage où on stocke que les éléments non nuls. Cela permet de minimiser l'empreinte mémoire.

**Question.** Ecrire un template de classe avec trois paramètres :  $n$ ,  $m$  et la classe  $\mathbb{K}$  avec comme membres :

- un entier "nnz" correspondant au nombre d'éléments non nuls,
- trois pointeurs, un de type  $\mathbb{K}$ , deux de type "int" qui contiendront les tableaux pour : la valeur, la ligne et la colonne de chaque élément non nul.

Pour ce template de classe on veut les constructeurs/destructeurs classiques et un prenant "nnz" comme paramètre et allouant la mémoire.

**Question.** Construire une fonction qui remplit les tableaux à l'aide de vecteurs de taille "nnz" donnés en paramètre. Construire une fonction ou opérateur permettant d'accéder au coefficient  $a_{ij}$ .

**Question.** Écrire une fonction qui calcule la trace dans un cas spécialisé (exemple  $n=2$ ,  $m=2$  et  $\mathbb{K} = \text{double}$ ).

**Question.** Surcharger l'opération "+" pour les matrices. Il faut faire attention en stockage creux. Il faut d'abord compter le nombre d'éléments non nuls généré par la somme (nombre de coefficient ou l'une des deux matrices à un coefficient non nul).

**Question.** Écrire une classe  $V_m(\mathbb{K})$  les vecteurs de taille  $m$  avec les constructeurs/destructeurs associés. Surcharger le "\*" (la notion d'ami) entre une matrice de  $\mathcal{M}_{n,m}(\mathbb{K})$  et un vecteur de type  $V_m(\mathbb{K})$ .