

Rapport TP1 Algorithmique : génération de maillages

Aire et transformation affine du triangle

L'aire du triangle unité vaut $\frac{1}{2}$. Si l'on considère un triangle quelconque de sommets q_0 , q_1 , et q_2 , alors l'aire de ce triangle est donnée par :

$$A = \frac{1}{2} |\det(q_1 - q_0, q_2 - q_0)|.$$

Ainsi, pour obtenir un maillage d'un triangle quelconque à partir du triangle unité, il suffit d'appliquer la transformation affine :

$$q = q_0 + (q_1 - q_0)x + (q_2 - q_0)y,$$

où (x, y) sont les coordonnées d'un point du triangle unité. Les tableaux de connectivité **me** restent identiques, seule la position des points change suivant la transformation.

Carré unité

Pour les **q**, on peut remarquer que les points d'une ligne à l'autre ont les mêmes coordonnées en X, c'est-à-dire qu'un point a la même abscisse que le point au-dessus. On stocke alors ceci dans un tableau **A** de taille N_x tel que :

$$T(i+1) = \frac{i}{N_x - 1} \quad \text{pour } i = 0 \text{ à } N_x - 1.$$

C'est ainsi que l'on remplit la première ligne de **q**, on aura N_y répétitions des mêmes abscisses. Pour ce qui est de la deuxième ligne de **q**, c'est assez simple : les points d'une ligne ont tous la même ordonnée, en commençant de 0 et en incrémentant de $\frac{1}{N_y - 1}$ à chaque N_x .

Pour les **me**, on peut remarquer que la connectivité se construit cellule par cellule : chaque cellule carrée du maillage, définie par deux points consécutifs sur une ligne et les deux points correspondants sur la ligne du dessous, se découpe toujours en deux triangles. On parcourt donc les cellules ligne par ligne et colonne par colonne, et pour chaque cellule on stocke les deux triangles correspondants.

Plus précisément : si on numérote les nœuds ligne par ligne, alors le premier point situé à la ligne j a pour indice :

$$ip = (j-1) \times N_x + 1.$$

Les quatre coins de la cellule sont donc :

$$ip, ip + 1, ip + N_x, ip + N_x + 1.$$

On découpe la cellule selon la même diagonale. Les deux triangles créés sont :

$$\text{triangle 1 : } [ip, ip + 1, ip + N_x], \quad \text{triangle 2 : } [ip + 1, ip + N_x + 1, ip + N_x].$$

Avec les deux boucles $j = 1 : N_y - 1$ et $i = 1 : N_x - 1$, **ip** prend la valeur de chacun des points de la ligne et les deux lignes du dessus permettent de relier les points pour former les triangles inférieurs et supérieurs respectivement.

Rectangle unité

Pour le **unitrectangle**, c'est exactement la même idée que pour le carré : on construit d'abord le maillage comme pour le **unitsquare**, puis on applique une transformation affine sur les coordonnées pour passer de $[0, 1] \times [0, 1]$ à $[a, b] \times [c, d]$. Les abscisses restent répétées ligne par ligne, les ordonnées sont incrémentées par pas $\frac{1}{N_y - 1}$, et **me** ne change pas.

Triangle unité

Pour remplir le tableau **q** pour le **unittriangle**, on utilise une logique similaire au carré. On crée un tableau de tableaux **T = cell(1,N)** qui fait que **T{i}** va contenir toutes les abscisses des points de la i ème ligne. La longueur des **T{i}** varie donc avec la ligne, correspondant au nombre de points de celle-ci.

On effectue ensuite une simple boucle pour remplir les deux lignes de **q** : pour la première, du premier point de la ligne i jusqu'au dernier (soit $\text{length}(\mathbf{T}\{i\})$), on assigne directement $\mathbf{T}\{i\}$. Pour la seconde, chaque point d'une ligne possède la même ordonnée ; on utilise donc un tableau initialisé avec les valeurs des ordonnées, tel que :

$$B(i) = \text{ordonnée des points de la } i\text{ème ligne.}$$

On affecte donc à la deuxième ligne de **q** les valeurs de $B(k)$ où k est incrémenté à chaque itération de la boucle sur les lignes.

En ce qui concerne les **me**, de légères modifications sont nécessaires par rapport au code du **unitsquare**. La formule pour **ip** devient :

$$ip = \frac{(j-1)(2N-j)}{2} + j,$$

qui donne le premier point de la j ème ligne. On ajuste ensuite les relations entre les points des deux triangles d'une cellule :

$$\text{triangle 1 : } [ip, ip+1, ip+N], \quad \text{triangle 2 : } [ip+1, ip+N+1, ip+N].$$

Cependant, notre ligne n'est pas constituée de N_x points : à chaque montée d'une ligne, le nombre de points diminue de 1. On remplace donc $ip+N$ par $ip+N-m$, où m est initialisé à 0 et incrémenté à la fin de chaque boucle.

Puisque le nombre de points diminue, on adapte aussi les boucles :

pour les triangles inférieurs : $i = 1 : N-1-m$, pour les supérieurs : $i = 1 : N-1-m-1$.

Triangle quelconque

Enfin, pour un triangle quelconque, on procède de la même manière que pour le rectangle : une simple application affine permet de transformer les coordonnées du triangle unité, tandis que les tableaux **me** restent inchangés.