

Animation de fluide

IFT3355 : Infographie - TP #3

Jérémie DUMAS

7 avril 2011

Université de Montréal

Table des matières

1	Animation des particules	2
1.1	Vitesse et position	2
1.2	Correction au niveau de la paroi	2
2	Surface implicite	3
2.1	Définition de l'interface	3
2.2	Approximation du gradient	4
3	<i>Marching tetrahedra</i>	4
3.1	Partage sur une grille	4
3.2	Interpolation des valeurs	4
	Conclusion	5

1 Animation des particules

1.1 Vitesse et position

Une première partie du travail effectué concerne l'animation d'un fluide. Concrètement, un fluide est représenté ici par un ensemble de particules ponctuelles, sur lesquelles s'exercent un certain nombre de forces : gravité, pression, viscosité et tension élastique dans notre cas.

Pour une particule donnée à une date t de l'animation, les équations de la mécanique classiques nous permettent de lier l'accélération de cette particule à la somme des forces qui s'y appliquent par la relation :

$$m \cdot a(t) = \sum_k F_k(t) \quad (1)$$

Où m représente la masse de la particule, $a(t)$ son accélération à la date t , et les $F_k(t)$ les forces subie par cette particule à la date t .

En utilisant le fait que $a(t) = \lim_{\Delta t \rightarrow 0} \frac{v(t+\Delta t) - v(t)}{\Delta t}$, il est facile d'obtenir une approximation de la vitesse d'une particule à une date $t + \Delta t$ à partir des valeurs à la date t , si le pas de temps Δt est suffisamment petit. De même on réutilise la valeur de la vitesse $v(t + \Delta t)$ pour approximer la position $p(t + \Delta t)$ de la particule en faisant une intégration semi-explicite. Concrètement cela nous donne donc comme équations :

$$v(t + \Delta t) \approx v(t) + \Delta t \cdot a(t) \quad (2)$$

$$p(t + \Delta t) \approx p(t) + \Delta t \cdot v(t + \Delta t) \quad (3)$$

1.2 Correction au niveau de la paroi

L'équation 3 ci-dessus est valable tant que la particule ne rencontre pas d'obstacle sur son chemin. En cas de contact avec la paroi d'un contenant (il n'y a pas de collision entre particules), il est donc nécessaire d'adapter le modèle afin de corriger la position finale de chaque particule après intégration. Notons que l'on fait là encore des approximations, plutôt que de calculer explicitement le rebond de chaque particule sur la paroi par exemple.

Changeons les notations de l'équation ci-dessus et appelons $p_i = p(t)$ la position de notre particule à l'instant t , et $p_f \approx p_i + \Delta t \cdot v(t + \Delta t)$ la position finale estimée. En cas de collision sur la paroi, on va projeter la vitesse et la position finale estimée répéter le processus jusqu'à ce que la particule se trouve à une position acceptable. Concrètement on définit les entités suivantes :

- $p_c^{(k)} = [p_i^{(k)}, p_f^{(k)}] \cap \mathcal{C}$, point de contact (s'il existe) à l'intersection du contenant \mathcal{C} et des positions courantes et finales de la particule, où $\vec{n}^{(k)}$ désigne la normale à \mathcal{C} .
- $p_i^{(k)} = p_c^{(k)} + \varepsilon \vec{n}^{(k)}$, le point de contact corrigé (avec ε petit et $p_i^{(0)} = p_i$).
- $p_f^{(k)} = p_f^{(k-1)} - \left((p_f^{(k-1)} - p_i^{(k)}) \cdot \vec{n}^{(k)} \right) \vec{n}^{(k)}$ la position finale estimée (avec $p_f^{(0)} = p_f$).

– $v^{(k)} = v^{(k-1)} - (v^{(k-1)} \cdot \vec{n}^{(k)})\vec{n}^{(k)}$ la vitesse de la particule (avec $v^{(k)} = v(t + \Delta t)$).

Finalement on prendra donc comme valeur finale pour la position $p(t + \Delta t) = \lim_{k \rightarrow +\infty} p_f^{(k)}$, et la vitesse corrigée $\tilde{v}(t + \Delta t) = \lim_{k \rightarrow +\infty} v^{(k)}$. Voir la figure 1 pour une illustration de la situation.

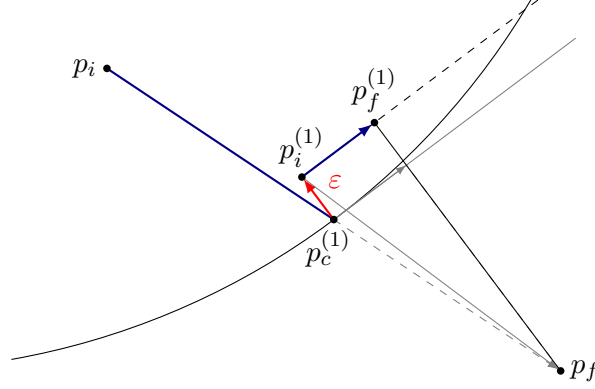


FIGURE 1: Correction de la trajectoire d'une particule.

Notons le terme de correction $\varepsilon \vec{n}$ qui sert à deux choses : premièrement éviter les instabilités numériques et l'ambiguïté qui se produit si l'on place la particule aux positions $p_i^{(k)} = p_c^{(k)}$ (possibilité de se retrouver "à l'extérieur" du contenant sans le vouloir) ; deuxièmement dans le cas du cercle, la projection exacte de $p_f^{(k)}$ sur chaque tangente donnerait toujours un point à l'extérieur du disque (ou convergerait vers un point de la paroi, mais au bout d'un temps infini), d'où un risque de boucle infinie pour le programme.

Enfin nous avons pu remarquer que cette méthode est très sensible dans le sens que si l'on modifie un tout petit peu la façon dont on corrige la position au niveau de la paroi, les particules se mettent très facilement à la traverser, ou le programme à partir en boucle infinie (oscillations des projections, convergence lente, etc.).

2 Surface implicite

2.1 Définition de l'interface

Définir l'interface du liquide à l'aide d'une fonction implicite, cela signifie se donner une fonction f telle que $f(x, y, z) = 0 \Leftrightarrow (x, y, z) \in \mathcal{S}$, où \mathcal{S} est la surface représentant notre liquide. Dans le cas présent, la définition de f repose sur le calcul de la densité de particules dans le liquide. Comme il y a moins de particules dans une région donnée sur les bords du liquide, on va juste définir un seuil à partir duquel on se considère comme étant *en dehors*.

Concrètement pour un point $p = (x, y, z) \in \mathbb{R}^3$, on définit la densité du liquide $\rho(p)$ et notre fonction implicite $f(p)$ comme dans l'énoncé du TP :

$$\rho(p) = \sum_{i \in \mathcal{N}(p)} m_i \cdot W(\|p - p_i\|^2)$$

$$f(p) = \frac{\rho(p)}{\rho_0} - (1 - a)$$

Et où $\mathcal{N}(p) = \{i, \|p - p_i\| \leq h\}$ qui désigne l'ensemble des particules situées à une distance inférieure à h de p (h fixé).

2.2 Approximation du gradient

La direction de la normale en un point de la surface nous est donné habituellement par le gradient ∇f de la fonction implicite f qui la définit. Ici cependant on travaille dans un espace discrétisé, avec un nombre limité de particules, et utiliser directement le gradient ∇f donnerait des résultats trop approximatifs. De fait on utilisera une formule un peu différente, donnée dans l'énoncé, pour approximer le gradient :

$$\vec{n}(p) = \sum_{i \in \mathcal{N}(p)} W(\|p - p_i\|^2) \cdot (p - p_i) \quad (4)$$

Un détail important tout de même : la valeur calculée dans le code correspondant de `SPH.cpp` n'a pas à être normalisée, car il ne s'agit pas véritablement de la normale en un point de la surface \mathcal{S} , mais d'un point de l'espace proche de la surface. En ce sens, on utilise la valeur du gradient en ce point (donné par 4), pour ensuite interpoler la normale sur la surface (et donc on ne normalise qu'après l'interpolation).

3 Marching tetrahedra

3.1 Partage sur une grille

Maintenant que nous avons défini notre surface implicite, il faut trouver les points de l'espace qui vérifient $f(x, y, z)$, en faire un maillage approximatif et l'afficher. On utilise pour cela l'algorithme du *marching tetrahedra*, dont le but est d'échantillonner les valeurs de f sur une grille régulière, et d'interpoler les valeurs trouvées pour approximer la position des zéros de la fonction.

Concrètement on dispose d'une grille régulière qui divise l'espace en cubes de petites tailles. Chaque élément de la grille va ensuite être divisé en 6 tétraèdes comme indiqué sur la figure 2 (le code utilisé reprend la même numérotation pour les sommets). On fait bien attention à découper les faces opposées dans le même sens.

3.2 Interpolation des valeurs

Une fois que l'on dispose d'un découpage en tétraèdes, et des valeurs de f en chaque sommet de nos tétraèdes, on peut en déduire un maillage approximatif de la surface

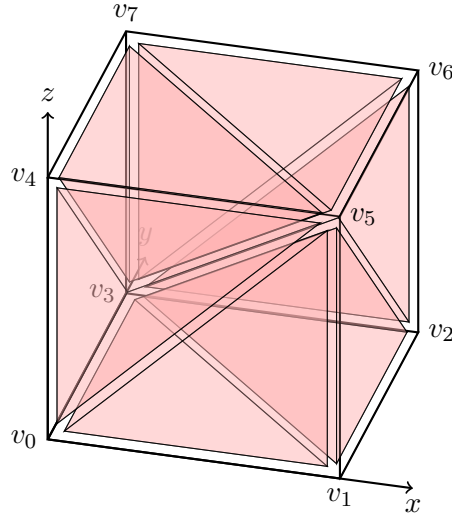


FIGURE 2: Découpage d'un cube en tétraèdres.

du fluide. Considérons un tétraèdre (p_1, p_2, p_3, p_4) , tel que $f(p_i) = v_i$, alors il suffit de comparer les signes des v_i entre eux (fonction **RenderTetrahedron**) :

Si $v_i \cdot v_j < 0$, alors f s'annule sur l'arête $[p_i, p_j]$ par continuité de f . On peut alors approximer la position p_m du point de l'arête qui annule f en interpolant avec la formule :

$$p_m = \frac{1}{v_i - v_j}(v_i p_j - v_j p_i)$$

De même pour calculer la normale à la surface en p_m , on utilise le gradient \vec{n}_i calculé sur les sommets p_i , et l'on peut interpoler de la manière suivante :

$$\begin{aligned} \vec{n}_m &\propto |v_i| \vec{n}_j + |v_j| \vec{n}_i \\ \|\vec{n}_m\| &= 1 \end{aligned}$$

Conclusion

Simuler un liquide en infographie, ce n'est pas évident pour plusieurs raisons : tout d'abord, parce qu'implanter un modèle physique parfaitement réaliste est souvent impossible pour des questions de performances. Il faut donc faire des approximations, à plusieurs niveaux, que ce soit pour accélérer le temps de calcul, éviter les instabilités numériques, etc. Ensuite il faut savoir que le résultat obtenu aura toujours des limites. Que ce soit sur la forme du contenant, le nombre de particules qu'il faudra augmenter pour avoir un liquide plus fluide, etc.

Références Aucune.