

WaterSim

Simulation réaliste de fluides

Pocheville Aurélien
15 mai 2002

Introduction

Le but de ce projet était d'implémenter une simulation réaliste de fluides en mouvement. Cette simulation repose sur un modèle mathématique qui s'appuie sur les équations de Navier-Stokes. Le but n'est pas d'obtenir un système qui se comporte le plus possible comme la réalité, mais bien de faire quelque chose d'approchant la réalité. Cette démarche implique que l'on peut sacrifier quelque exactitude pour gagner en temps de calcul. De plus, comme le système simulé est bien connu (il est aisé d'observer des fluides dans la nature), il est possible de paramétrer la simulation de manière adéquate afin d'obtenir un résultat satisfaisant. Les équations sont issues et dérivées de l'article par James F. O'Brien et Jessica K. Hodgins qui peut être trouvée à l'adresse : <http://www.cc.gatech.edu/gvu/animation/Animation.html>. Elles ont été légèrement modifiées pour obtenir un résultat concluant.

Dans la partie 1, nous indiquerons les équations et les approximations utilisées. Dans la partie 2, nous discuterons de l'implémentation et des difficultés rencontrées.

1 Modèle mathématique

Pour simuler un fluide de manière exacte, il suffit de résoudre les équations 3D de Navier-Stokes. Toutefois, pour des systèmes complexes, ces équations peuvent être soit insolubles, soit prendre un temps considérable à résoudre. Cette approche, privilégiée pour les simulations scientifiques requérant un haut degré de réalisme, ne nous intéresse pas ici. D'autres techniques impliquent l'usage d'un système de particule avec interaction, ou bien des fonctions décrivant la forme des vagues. Nous utilisons ici la méthode proposée par James F. O'Brien et Jessica K. Hodgins, c'est à dire considérer le fluide comme un regroupement de cellules de

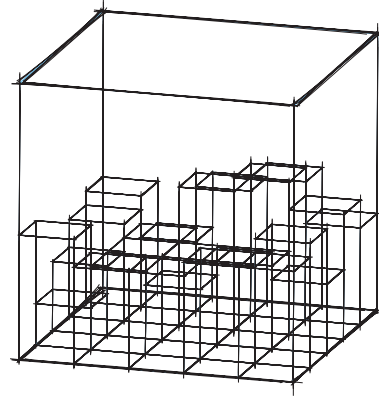


FIG. 1 – Le fluide est représenté par des colonnes de différentes hauteurs. La surface est ensuite interpolée suivant la hauteur des colonnes. Pour la représentation graphique, une interpolation plus fine que celle de la simulation a été mise en place (pour la simulation, il n'y a des points de contrôles qu'aux extrémités des colonnes)

différentes hauteurs, qui peuvent interagir entre elles. Cela permet de ramener une complexité en n^3 (ou n est la dimension du système) à une complexité en n^2 (donc un gain de temps non négligeable). De plus, une fois les interactions entre les différentes cellules spécifiées, la forme des vagues ainsi que le retour à un état de repos sont inhérents au système. Cette méthode permet aussi de gérer très simplement et rapidement les interactions avec l'extérieur (objets, sources, puits...). La méthode de James F. O'Brien et Jessica K. Hodgins étend celle de Kass et Miller où le fluide est considéré comme un champ de hauteur.

1.1 Le modèle utilisé

Le fluide est principalement un regroupement de cellules (Fig.1) (cela revient au même qu'un champ de hauteur). Ces cellules représentent le volume du fluide. Ce sont elles qui seront responsables du changement de forme de la sur-

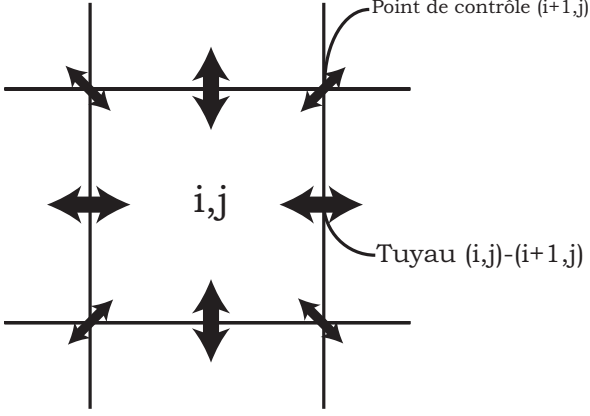


FIG. 2 – La grille des colonnes. Chaque colonne communique avec des tuyaux avec ses voisins. Il y a 8 tuyaux qui relient une colonne à ses voisins. Les points de contrôle de la surface sont aux intersections.

face. Toutefois, pour atteindre un plus haut degré de réalisme, deux autres systèmes sont ajoutés : La surface représente la couche externe du fluide, et permet les interactions avec les objets extérieurs. Enfin, un système de particule est utilisé pour simuler les gouttelles¹ qui peuvent s'échapper du fluide. Ces trois système interagissent pour créer la simulation.

1.2 Le système volumique

Pour modéliser le volume du fluide, on subdivise celui-ci suivant une grille. Cela forme une série de colonnes qui ont la hauteur moyenne du fluide à ce point là. On suppose que le fluide est isotropique verticalement (pas de gradient de température par exemple), et que donc le fluide contenu dans une colonne a les mêmes propriétés tout au long de cette colonne. Pour communiquer, les colonnes sont reliées par des

tuyaux (Fig.2) qui permettent au fluide de s'écouler. Chaque colonne a donc 8 voisins (les diagonales sont prises en compte) avec qui elle communique. Le flux dans chaque tuyau est calculé grâce aux équations de l'hydrostatique. Soit H_{ij} la pression de la colonne située en (i, j) . On a alors :

$$H_{ij} = h_{i,j}\rho g + p_0 \quad (1)$$

où ρ est la densité du fluide, g la gravitation (9.81 m.s^{-2}), et p_0 la pression atmosphérique standard. h_{ij} est la hauteur de la colonne, qui est reliée au volume par :

$$h_{i,j} = \frac{V_{i,j}}{c_{i,j}} \quad (2)$$

où $c_{i,j}$ est la section d'une colonne (dans la simulation présente, c'est une constante déterminée par la finesse de la grille et les dimensions physiques de la simulation). L'équation 1 est une approximation, car on suppose que le fluide ne se déplace pas vite. On veut aussi que le fluide soit en interaction avec les objets, on utilise donc une force externe qui entraîne une pression externe E_{ij} . Cette force est obtenue grâce à la force sur la surface (Cf.1.3), pour le moment il suffit de l'inclure dans l'équation :

$$P_{ij} = h_{ij}\rho g + p_0 + E_{ij} \quad (3)$$

où P_{ij} est la pression totale s'exerçant sur la colonne. La force qui agit sur le fluide est proportionnelle à la différence de pression entre les deux extrémités de la façon suivante :

$$F_{i,j \rightarrow k,l} = c\Delta P_{i,j \rightarrow k,l} \quad (4)$$

où (k, l) représente la colonne voisine². On en déduit l'accélération³ :

$$a_{i,j \rightarrow k,l} = \frac{c\Delta P_{i,j \rightarrow k,l}}{m} \quad (5)$$

¹La différence entre une particule et un objet est très mince, nous verrons dans la partie 1.4 que le système de particule gère aussi les objets

² $((k, l) \in (i \pm 1, j \pm 1))$

³grâce à $F = ma$.

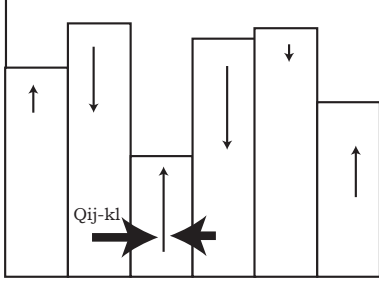


FIG. 3 – Modélisation des échanges entre les colonnes. Plus la différence de hauteur est grande, plus le flux est grand.

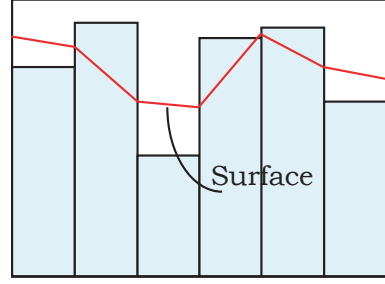


FIG. 4 – La surface réalise une interpolation entre les colonnes. Chaque point de contrôle de la surface appartient à quatre colonnes.

où m est la masse du fluide dans le tuyau ⁴. On utilise une intégration discrète pour obtenir le flux :

$$Q_{i,j \rightarrow k,l}^{t+\Delta t} = Q_{i,j \rightarrow k,l}^t + \Delta t c a_{i,j \rightarrow k,l} \quad (6)$$

De même pour le changement de volume, en considérant tous les tuyaux :

$$\Delta V_{i,j} = \Delta t \sum_{(k,l) \in (i \pm 1, j \pm 1)} Q_{i,j \rightarrow k,l}^{t+\Delta t} \quad (7)$$

L'équation 2 nous donne :

$$\Delta h_{i,j} = \frac{\Delta t \sum_{(k,l) \in (i \pm 1, j \pm 1)} Q_{i,j \rightarrow k,l}^{t+\Delta t}}{c_{i,j}} \quad (8)$$

Quelques conditions doivent être ajoutées au système :

$$\forall (i,j) \forall (k,l) \in \eta_{i,j} : Q_{i,j \rightarrow k,l} = -Q_{k,l \rightarrow i,j} \quad (9)$$

où $\eta_{i,j} = (i \pm 1, j \pm 1)$. La méthode originale de James F. O'Brien et Jessica K. Hodgins gardait le volume constant. Comme la méthode implémentée ici gère les sources et les puits, cela n'a plus de sens. Toutefois, le principe a été gardé pour la modélisation : Un mur est modélisé par un flux nul sur tout ses voisins, un puit, par un flux négatif, et enfin une source par un flux positif. Le comportement pour les puits est celui souhaité, toutefois, pour les sources, quelques artefacts apparaissent.

⁴Etant donné l la longueur du tuyau, il est évident que $m = \rho c l$.

1.3 Le système surfacique

Le système surfacique permet l'interaction entre le fluide et les éléments externes (les objets). La surface est modélisé par une grille point rectiligne. Chaque point de la surface est le croisement de quatre colonnes (Fig.4). La position du point de la grille est obtenue en faisant la moyenne des quatres colonnes :

$$z_{i,j} = \frac{h_{i+1,j} + h_{i,j} + h_{i,j+1} + h_{i+1,j+1}}{4} \quad (10)$$

De même, les forces externes sont redistribuées de façon uniforme sur les différente colonnes :

$$E_{i,j} = \frac{-f_e}{4 * c_{i,j}} \quad (11)$$

1.4 Le système de particules

Lors d'impacts d'objets dans un fluide, il est fréquent que particules se détachent du volume principal du fluide pour former des gouttellettes. Pour modéliser cet aspect, on utilise un système de particules basique⁵. Pour

⁵Chaque particule est une petite sphère, à chaque itération on calcule les forces qui s'appliquent sur la sphère (en général il n'y a que la gravitation), puis on calcule la vitesse et enfin la position. Si la position implique la destruction de la particule, on la retire du système

déterminer si l'on doit créer une particule, on se base sur la vitesse verticale de la surface. Si la vitesse d'un point de contrôle est supérieure à un certain seuil⁶, une - ou des - particule(s) est (sont) générée(s). la vitesse verticale, vaut, par dérivation de l'équation 10 :

$$z_{i,j} = \frac{\dot{h}_{i+1,j} + \dot{h}_{i,j} + \dot{h}_{i,j+1} + \dot{h}_{i+1,j+1}}{4} \quad (12)$$

La vitesse horizontale peut être calculée selon le même principe, toutefois on s'aperçoit que tirer les vitesses suivant l'axe x et l'axe y au hasard donne un résultat plus convaincant à l'œil. Le volume de la particule générée est déterminé en calculant le volume maximal que pourrait obtenir les colonnes proches, puis en soustrayant ce volume au volume actuel. Une fois séparée du fluide, les particules tombent sous l'influence de la gravité. Lorsqu'elles rencontrent l'eau, leur volume est de nouveau ajouté au système (distribué sur toutes les colonnes que la particule touche).

Les objets

On s'aperçoit que pour modéliser des objets de forme sphérique, le système de particule est idéal. Dès lors, au lieu de séparer les objets dans un autre système, il est préférable de les intégrer au système de particules. La seule différence est lorsque l'impact a lieu : les particules sont immédiatement absorbées (toutefois si leur vitesse est suffisante, elles pourront donner lieu à la naissance d'autres particules si elle génère des vagues importantes), alors que les objets flottent sur - ou s'enfoncent dans - l'eau. Le comportement de l'objet est déterminé comme suit : Tant qu'il n'est pas en contact avec l'eau, il tombe sous l'influence des forces externes (dans la pratique, seule la gravité est considérée). Si il entre en contact

avec l'eau, sa vitesse est réduite à cause des frottements. De plus, il génère une force sur l'eau proportionnelle à la vitesse de l'objet⁷. Si la vitesse est trop petite, aucune force n'est générée. La force exercée sur l'objet est calculé grâce au volume d'eau déplacé (loi d'Archimède). Cette modélisation des objets en interactions avec l'eau n'est pas celle de James F. O'Brien et Jessica K. Hodgins, toutefois elle donne de très bons résultats. Il est très impressionnant de voir qu'une balle de densité plus importante flotte moins qu'une balle de densité faible, de même les balles de densité importantes remontent plus lentement etc... De plus, cette méthode est très rapide en temps de calcul et permet donc d'avoir de nombreux objets. Lorsqu'un objet rencontre l'eau, la force qu'il exerce sur celle-ci est distribuée entre toutes les colonnes qui sont sous l'objet, cela permet d'avoir des impacts de taille différente, et donc des vagues de fréquence différente. La méthode peut être très simplement étendue pour utiliser des objets de forme cubique⁸.

2 Implémentation des équations

L'implémentation des équations précédentes a été faite en C++ et OpenGL. Il n'y a de difficulté majeure, le principal problème restant la rapidité. En effet, si le temps de calcul est trop grand, il ne devient plus possible

⁷Puisque $F = ma$ et que les forces exercées sur l'eau et sur l'objet sont réciproques, la force exercée sur l'eau devrait en théorie être proportionnelle à l'accélération. Toutefois, dans la pratique, le résultat est beaucoup plus concluant si cette force est proportionnelle à la vitesse (en fait, c'est que l'on s'attend aux frottements, et que ceux-ci sont proportionnels à la vitesse)

⁸ou même parapipédique, toutefois ces objets présentent moins d'intérêt que les sphères. Des objets composés de différentes parties peuvent être aussi gérés simplement

⁶Ce seuil dépend des autres paramètres, de bons résultats pour l'eau sont obtenus entre $2.0ms^{-1}$ et $2.5ms^{-1}$

de faire du temps réel⁹. Pour la représentation, des ombres et des reflets sont ajoutés pour plus de réalisme. De plus, la grille de représentation est indépendante de la surface, donc il est possible d'avoir une grande finesse. OpenGL est chargé de gérer tous les calculs d'éclairage et de texture (toutefois, pour des raisons de rapidité, les normales à la surface sont calculées dans le programme). L'alpha-blending¹⁰ est utilisé pour donner un aspect plus liquide. Le programme possède un mode pas à pas, un mode temps réel, et un mode 'capture', qui réalise une séquence d'image pour transformation en vidéo.

3 Résultats et conclusions

Sur un ordinateur personnel équipé d'un Pentium 700 Mhz et d'une GeForce2, la simulation tourne en temps réel avec une résolution de 50x50 jusqu'à 80x80. Sur un Athlon 1900+, cette même simulation peut aller jusqu'à une grille de 140x140. Les calculs de simulations sont totalement non optimisés, car il prennent en compte n'importe quel type de fluide. Toutefois, il serait possible d'optimiser grandement la simulation en fixant les paramètres. De plus, le haut degré de réalisme OpenGL ralentit aussi la simulation. C'est pour cela que le mode 'capture' a été créé. Il permet d'avoir des vidéos de hautes qualités, toutefois elles ne sont plus en temps réel. Le temps réel est idéal pour interagir avec la simulation. Le résultat visuel de la simulation est assez impressionnant, et démontre la validité du modèle. Un plus grand réalisme pourrait être obtenu en générant du bump-mapping¹¹. On pourrait aussi générer de l'écume, etc... L'objectif principal de ce projet était l'exécution du modèle en temps réel, un

nouvel objectif pourrait être une simulation non différentiable de la réalité.

Références

- James F. O'Brien et Jessica K. Hodgins
Dynamic Simulation of Splashing Fluids,
<http://www.cc.gatech.edu/gvu/animation/Animation.html>, 1995
- Bjarne Stroustrup, The C++ Programming Language, Special Edition, 2000
- Mason Woo, Jackie Neider, Tom Davis, Dave Shreiner, OpenGL Programming guide, Third Edition, 1999
- Flipcode, <http://www.flipcode.com>

⁹Si le temps de calcul est trop grand, le ΔT le sera aussi, et donc le système diverge (approximation ΔT petite fausse)

¹⁰transparence

¹¹Technique qui permet de donner du relief aux textures.