

Projet de Simulateur de Fluide

Stephane LEJEUNE, Jacques PHAM BA NIEN

9 mai 2025

Table des matières

1	Introduction	3
2	Théorie de simulateur de fluide	3
2.1	Equation de Navier-Stokes	3
2.2	Simulateurs	5
2.2.1	Simulateur Newtonien	5
2.2.2	Simulateur Lagrangien	5
2.2.3	Simulateur Mixtes	5
2.2.4	Comparaison des approches	5
2.2.5	Simulateur SPH	6
3	Le projet	6
3.1	Pre-unity : le choix de rust	6
3.2	Post-unity : l'abandon de rust	6
4	Conclusion	6

1 Introduction

Pour le projet, nous avons choisis de construire un simulateur de fluide, projet non proposé de base, mais c'est un projet que Stephane et Jacques voulaient déjà faire depuis la L2, nous avons donc proposer le projet et il a été accepté.

Étant en double licence, le style de compétences utiles qu'on peut apprendre avec un projet informatique sont différentes, le projet de simulateur de fluide nous a sembler être un bon compromis entre demandant des compétences informatiques et mathématiques.

Ce projet, nous permet d'explorer comment les techniques qu'on as appris durant différent cours analyse numériques peuvent être utiliser pour quelque chose de plus "réel" (et visuel) que résoudre un système linéaire ou des équations différentiels abstraites.

Du point de vu informatique, ce projet nous oblige à interagir avec une interface graphique, choisir nos structures pour représenter les particules (ou ne pas les représenter)

Nous sommes donc reconnaissant que notre proposition de projet ait été accepté.

L'objectif du projet est double :

1. Implémenter une simulation de fluide stable, visuellement cohérente, et interactive, capable de gérer plusieurs centaines, voire milliers de particules.
2. Intégrer cette simulation dans un moteur de jeu 2D, en l'occurrence, *Unity*, pour permettre une visualisation en temps réel avec un contrôle sur les paramètres physiques.

Durant le développement, de nombreux défis ont été rencontrés : gestion des densités divergentes, explosions numériques liées à une mauvaise évaluation de la pression, instabilités aux frontières, ou encore des performances non optimales à partir d'un certain nombre de particules. Plusieurs améliorations progressives ont été apportés, incluant l'ajout d'une grille spatiale, la prédiction de position, un modèle de pression stable (Tait), et une modélisation simplifiée de la viscosité.

Ce rapport détaille les fondements physiques du modèles SPH, la manière dont il a été implémenté, optimisé et visualisé dans *Unity*, ainsi que les résultats obtenus, leurs limites et des pistes d'améliorations futures.

2 Théorie de simulateur de fluide

2.1 Equation de Navier-Stokes

Pour comprendre comment un simulateur de fluide marchent, il nous as d'abord fallu comprendre comment un fluide est sensé se comporter.

On as certes un modèle en nous de comment cela fonctionne, on a déjà vu des fluides, mais transmettre cette intuition en instructions est loin d'être facile, cela n'est pas notre travail, c'est celui des physiciens. Nous avons donc regarder à comment les physiciens décrivent les lois que les fluides doivent respecter.

Déjà, un “fluide” en physique ne décrit pas seulement le comportement d’un liquide, cela décrit aussi le comportement des gas, leurs comportement sont étudiier dans la “mécanique des fluides”.

Ses lois sont appeler “équations de Navier-Stokes” :

1. Équation de continuité :

$$\nabla \cdot \vec{V} = 0$$

2. Équation de moments :

$$\rho \left(\frac{\partial \vec{V}}{\partial t} + \vec{V} \cdot \nabla \vec{V} \right) = -\nabla p + \mu \nabla^2 \vec{V} + \vec{F}$$

L’équation de continuité nous dit que l’énergie mécanique du fluide ne change pas avec le temps (sa dérivé est nulle).

L’équation de moments est plus complexe, elle décrit les forces locales qui sont exercé sur notre fluide, décortiquons ce que veut dire chaque termes :

- ρ est la densité (locale, mais cela n’importe peu pour la suite) de notre fluide, donc c’est la masse divisé par le volume autour d’un point.
- \vec{F} est la force externe exercé sur notre fluide, ici c’est juste la gravité. La force exercé par la gravité localement est $\vec{g} \frac{m}{V}$ où \vec{g} est la force gravitationnel universelle, m est la masse et V est le volume. Autrement dit, $\vec{F} = \rho \vec{g}$.
- \vec{V} est le champ vectoriel de la vitesse du fluide. Champ vectoriel car en tout point, le vecteur direction peut être différent.
- p est la pression du fluide.
- μ est une constante propre au fluide étudié, cela modèle la viscosité, plus elle est grande, plus le fluide est visqueux, ce terme limite l’accélération.

Cette équation nous dit donc que le changement de vitesse locale est du aux forces extérieurs, et à la différences de vitesse environnantes, en évitant de trop se compresser, et n’accélération pas trop vite proportionnellement à la constante de viscosité.

Tout du long, nous allons simplifier notre fluide étudiier, en général, les fluides dont on s’intéresse (surtout l’eau) ne sont pas visqueux (nous rajouterons la viscosité plus tard pour des raisons de précision numériques, mais ignorons ce paramètre pour l’instant), et aussi (quasiment) incompressible, cela simplifie nos équations, ce qui rends la simulation plus simple et plus rapide, nos nouvelles équations :

- Conservation :

$$\nabla \cdot \vec{V} = 0$$

- Incompressible :

$$\nabla p = 0$$

- Moments :

$$\frac{\partial \vec{V}}{\partial t} = \vec{g} - \vec{V} \cdot \nabla \vec{V}$$

2.2 Simulateurs

Certes maintenant on a “comment les fluides doivent agir”, on n’a pas “comment les faire agir comme tel”.

Pour observer la vie réel, cela n’a pas d’importance, pour un mathématicien non plus, mais un ordinateur ne peut pas gérer des équations différentiels de manière exactes sur un volume continu, il y aurait juste beaucoup trop de données à stocker et de calcul à faire (même, une infinité non dénombrable, d’où cette incapacité).

Ainsi, on est obligé de simplifier notre tâche, au lieu d’obtenir un système agissant exactement comme un fluide idéal, il faut faire un système qui agit approximativement comme un fluide idéal, les physiciens font déjà cela pour simplifier les calculs (nous ne faisons pas en général de la théorie des probabilités et des équations depuis la quantique pour des fluides, car il y aurait trop de termes agissant entre eux, obscurcissant les calculs, et même pourrait rendre des équations insolubles de manière numériques), en informatique, nous allons passer du continu au discret, perdant ainsi de la précision pour en échange, avoir de la possibilité et de la vitesse de calcul.

Il y a plusieurs manières de discrétiser notre problème :

2.2.1 Simulateur Newtonien

On peut discrétiser notre problème en considérant d’avoir une grille de vecteurs au lieu d’un champ, nous appelons chaque élément de la grille une “boîte” pour des raisons de facilité de visualisation.

Ainsi, il faut modéliser la quantité de fluide passant d’une boîte à ses boîtes voisines.

Nous avons seulement besoin de modéliser le passage de fluide d’une boîte à ses voisines directes dans les directions de gauche, droite, haut et bas, pas besoin des diagonales.

Ainsi, une optimisation est de sauvegarder les magnitudes de ces flux au lieu de la magnitude et direction du flux partant du centre de la boîte.

TODO

2.2.2 Simulateur Lagrangien

Une autre approche est de discrétiser notre problème en une manière plus intuitive, en simulant nos particules d’eaux, néanmoins, dans un fluide usuel, il y a beaucoup trop de particules, donc à la place, on va simuler des très grosses particules, ce qui est absurde physiquement mais marche relativement bien.

TODO

2.2.3 Simulateur Mixtes

TODO

2.2.4 Comparaison des approches

TODO

2.2.5 Simulateur SPH

C'est le simulateur que nous avons décider d'implémenté, nous avons d'abord pencher sur une approche Newtonienne, mais TODO TODO

3 Le projet

3.1 Pre-unity : le choix de rust

TODO

3.2 Post-unity : l'abandon de rust

TODO

4 Conclusion

TODO