

Numerical simulation of the *Cheerios Effect*

Erdi ÇAN

erdi.can@etu.sorbonne-universite.fr

Baptiste BRAUN-DELVOYE

baptiste.delvoye-braun@etu.sorbonne-universite.fr

Sorbonne Université, L2 CMI Mécanique

14 décembre 2022

Table des matières

1	Le problème et notre modélisation	2
1.1	Effet Cheerios	2
1.2	Force des bords	4
2	Méthodes numériques et algorithmes	4
2.1	Integration de Verlet	4
2.2	Collisions	5
2.2.1	Collisions Objet-Objet . . .	5
2.2.2	Collision des bords	5
3	Conception de notre algorithme	5
3.1	Notre algorithme	5
3.2	Améliorations possibles	6
4	Nos Résultats	7

Bibliographie

Table des figures

1	Géométrie d'une sphère reposant sur une interface liquide-gaz. La partie rayée représente le poids de liquide équivalent à la force de flottabilité due à la pression hydrostatique appuyant sur la sphère. . . .	2
2	Graphique représentant les fonctions de Bessel modifiées à l'ordre 0 et 1	3
3	Schéma des forces des bords. . . .	4
4	Schéma d'un rebond d'un objet sur un bord	5
5	Diagramme de notre algorithme pour un objet.	6
6	A 5 secondes	7
7	A 40 secondes	7

8	A 45 secondes, la collision en vraie vie se passe en 55 secondes	7
---	--	---

Introduction

Dans le cadre de l'UE Projet en Calcul Scientifique Numérique (LU2ME232), nous devons travailler sur un projet, afin de nous apprendre plus en détail, la programmation et le calcul numérique avec un langage compilé, le C. Notre sujet était sur l'"Effet Cheerios", ou l'interaction d'objets à la surface d'un liquide par l'effet de la gravité et la déformation interfaciale. Cet effet se caractérise par la tension d'une surface liquide sous le poids d'un objet, par exemple une punaise sur l'eau. Lorsque nous ajoutons plusieurs objets sur la même surface, à distance plus ou moins grande, les objets vont potentiellement s'attirer puis créer des tas mobiles. Ce phénomène est notamment visible avec des céréales dans du lait, d'où le nom de Cheerios, célèbre marque de céréales américaine. Pour réaliser à bien ce projet nous avons dû faire de nombreuses recherches sur la mécanique des fluides, les collisions inélastiques et nous avons également dû faire un travail conséquent sur l'optimisation de notre algorithme. Nous allons vous raconter comment fonctionne l'effet Cheerios et comment nous avons calculé les forces d'interaction. Puis nous vous expliquerons les méthodes numériques principales utilisées dans notre algorithme. Enfin nous vous expliquerons notre algorithme en pointant ses défauts, et nous finirons par une analyse critique des résultats obtenus.

Tout notre code source pour ce projet peut être trouvé sur : <https://github.com/erdi-cn/Cheerios>

1 Le problème et notre modélisation

Nous avons tous mangé des céréales ou vu des objets flottant s'attirer ou se repousser entre eux, mais quel est la raison de cette force ? Nous avons essayé de décrire ces interactions dans ce projet.

Quelque notations : dans ce rapport les vecteurs sont annotés en gras \mathbf{v} : vecteur v

Nom	Abréviation	Dimension
Rayon de courbure	R	$[L]$
Surface de tension	γ	$[MT^{-2}]$
Densité du solide	ρ_s	$[ML^{-3}]$
Densité du liquide	ρ_l	$[ML^{-3}]$
Densité de l'air	ρ_a	$[ML^{-3}]$
Nombre de Bond	B	1

TABLE 1 – Table des variables

1.1 Effet Cheerios

Cette partie est plutôt faite pour l'intégrité du rapport. Le lecteur est fortement encouragé à lire "Cheerios Effect"[1] pour avoir une compréhension plus complète du sujet. Les équations viennent principalement de cet article.

Lorsque nous posons un objet sur la surface de l'eau (une aiguille, une punaise ou un cheerio), il est possible que l'objet reste à la surface de l'eau. L'eau va donc se courber, enveloppant une partie de l'objet, sous la masse de celui-ci. Cela se nomme la déformation interfaciale. Elle se retrouve dans la nature avec certains insectes pouvant marcher sur l'eau grâce à cette loi physique. Si nous mettons plusieurs objets de la sorte et qu'ils sont plus ou moins proche, la courbure de l'eau sous ces objets va créer une tension de surface qui attirera les objets jusqu'à qu'ils se touchent. De plus, si nous mettons ces objets dans un récipient, au fil du temps ils vont s'approcher des bords. Nous pouvons également expliquer cela par la tension de surface qui est créée entre le récipient et l'eau qui créera un ménisque.

Nous voulons déterminer comment ces objets réagissent entre eux et les bords d'un récipient et représenter nos résultats de façon numérique et animée. Nous devons, pour cela, calculer tout d'abord les forces intervenant dans ce phénomène.

Une des raisons pour laquelle les objets flottent est due à la poussée d'Archimède, comme nous pouvons le voir dans la figure 1. Pour que notre

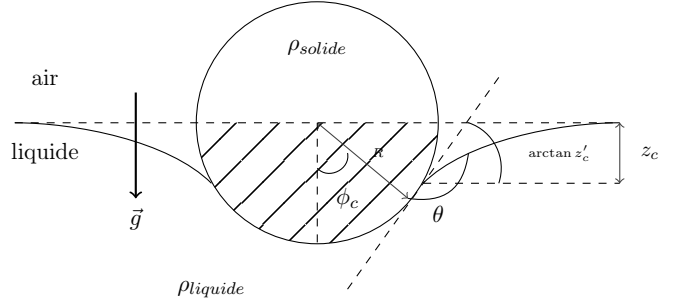


FIGURE 1 – Géométrie d'une sphère reposant sur une interface liquide-gaz. La partie rayée représente le poids de liquide équivalent à la force de flottabilité due à la pression hydrostatique appuyant sur la sphère.

sphère reste sur l'interface liquide-gaz elle a besoin que la norme de son poids $\|\mathbf{P}\| = \frac{4}{3}\pi\rho_s g R^3$; doit être équilibrée par la composante de tension superficielle agissant le long de la ligne de contact (circulaire) et par la force de flottabilité due au déplacement du fluide en vrac. La première composante a pour équation :

$$2\pi\gamma R \sin \phi_c \frac{z'_c}{\sqrt{1+z'^2_c}} \quad (1)$$

Et nous avons également la force de flottabilité par l'équation :

$$\pi\rho_l g R^3 \left(\frac{z_c}{R} \sin^2 \phi_c + \frac{2}{3} - \cos \phi_c + \frac{1}{3} \cos^3 \phi_c \right) \quad (2)$$

Nous avons donc l'équilibre des forces donné par :

$$\frac{4}{3}\pi\rho_s g R^3 = 2\pi\gamma R \sin \phi_c \frac{z'_c}{\sqrt{1+z'^2_c}} + \pi\rho_l g R^3 \left(\frac{z_c}{R} \sin^2 \phi_c + \frac{2}{3} - \cos \phi_c + \frac{1}{3} \cos^3 \phi_c \right) \quad (3)$$

Si nous substituons $\phi_c = \pi - \theta + \arctan z'_c$ et gardons uniquement les termes linéaires en z'_c , nous retrouvons l'expression pour $z'_c \sin \phi_c$ qui est précis par rapport à l'ordre linéaire du nombre de Bond, $B \equiv R^2/L_c^2$

Nous avons donc :

$$z'_c \sin \phi_c = B \left(\frac{2D-1}{3} - \frac{1}{2} \cos \theta + \frac{1}{6} \cos^3 \theta \right) \equiv B\Sigma \quad (4)$$

Avec $D \equiv \frac{\rho_s}{\rho}$.

L'équation (4) contient deux paramètres sans dimensions ; le nombre de Bond B et Σ , qui sont très importants pour notre modélisation.

Le nombre de Bond vaut :

$$B = \frac{(\rho_l - \rho_a)gR^2}{\gamma} \simeq \frac{R^2}{L_c^2} \quad (5)$$

Il nous donne la mesure relative de l'importance des effets de gravité et de la tension de surface ; si B est très grand, cela correspond à des particules grandes ou à une tension de surface petite.

Pour déterminer Σ , nous avons besoin de l'angle de contact θ . Pour le trouver nous avons suivi l'article *Lattice Boltzmann Simulation of Capillary Interactions among Colloidal Particles*[2] dans lequel ils utilisent un angle de contact (θ) satisfaisant la loi de Young-Dupré supposant que les particules sont assez petites pour que nous puissions négliger les effets de leurs poids sur l'angle de contact.

$$\cos \theta = \frac{\gamma_{SV} - \gamma_{SL}}{\gamma_{LV}} \quad (6)$$

Où $\gamma_{SV,SL,LV}$ est la tension superficielle des interfaces Solide/Vapeur, Solide/Liquide et Liquide/Vapeur.

De là, nous avons besoin des tensions superficielles expérimentales de chaque milieu ($\gamma_{SV,SL,LV}$). Mais comme nous n'avons pas ces valeurs pour chaque objet par rapport aux milieux nous pouvons seulement déterminer ceci avec des expériences. C'est pour cela que dans nos simulations nous avons essayé de prendre des valeurs d'angle de contact déjà calculées expérimentalement pour chaque objet.

Ensuite nous avons le déplacement interfacial[3] qui est :

$$\gamma \frac{d^2 h}{dx^2} = \rho_l g h \quad (7)$$

Si nous prenons en compte que l'objet a une symétrie sphérique nous obtenons :

$$\Rightarrow \frac{1}{r} \frac{d}{dr} \left(r \frac{dh}{dr} \right) = \frac{h}{L_c^2} \quad (8)$$

Nous pouvons en déduire une solution de cette équation avec la fonction de Bessel modifiée à l'ordre 0 [4] ($K_0 \left(\frac{l}{L_c} \right)$).

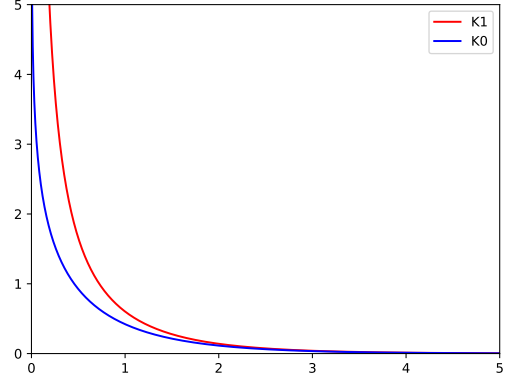


FIGURE 2 – Graphique représentant les fonctions de Bessel modifiée à l'ordre 0 et 1

Pour déterminer, maintenant la force d'attraction entre deux objets nous partons du poids effectif d'une sphère sur une interface déformée, que nous donnons comme $2\pi R B \Sigma$. Nous avons également calculé la déformation interfaciale causée par la présence d'une seule sphère. Nous sommes donc capables de calculer l'énergie d'interaction entre deux sphères. Cette énergie est le produit du poids résultant d'une sphère et de son déplacement vertical causé par la présence d'une autre sphère dont le centre est éloigné de l'horizontale d'une distance horizontale l . Nous pouvons donc écrire l'énergie, $E(l)$, comme suit :

$$E(l) = -2\pi\gamma R^2 b^2 \Sigma^2 K_0 \left(\frac{l}{L_c} \right) \quad (9)$$

Avec L_c la longueur capillaire.

Nous pouvons donc trouver la force d'interaction $F(l) = -\frac{dE}{dl}$, ce qui donne :

$$F(l) = -2\pi\gamma R B^{5/2} \Sigma^2 K_1 \left(\frac{l}{L_c} \right) \quad (10)$$

Nous voyons bien grâce à la Figure 2 que dès que $l/L_c > 5$, notre force sera très faible et inversement quand $l/L_c \ll 1$ la force d'attraction sera très élevée. La force entre objets flottants dépend donc de la distance entre eux, et cela de façon exponentielle.

Nous avons décidé de ne pas prendre en compte les forces de frottement entre les objets et le liquide car elles étaient négligeables et elles auraient grandement complexifiées les calculs de notre algorithme.

1.2 Force des bords

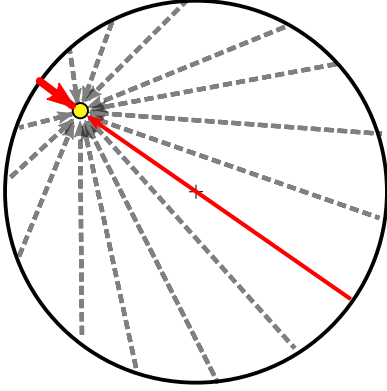


FIGURE 3 – Schéma des forces des bords.

Maintenant que nous avons vu l'application des forces entre objets, nous allons expliquer les forces entre les bords et les objets. La force se calcule de la même manière qu'entre deux objets (équation 10) car la force dépend du rayon de courbure(R), du nombre de Bond(B), de Sigma(Σ), de la longueur capillaire(L_c) et de la distance entre deux points(l) de force et tous ces paramètres peuvent être déterminés pour le bord. Pour la force appliquée par les bords sur les objets nous avons, à la place de calculer les forces à chaque point du cercle, opté d'utiliser la symétrie d'un cercle. Nous avons remarqué que la plupart des forces s'annulent entre elles (en gris) et il nous reste seulement deux forces (en rouge) qui interagissent comme le montre la figure 3. Les seuls paramètres à changer sont le nombre de Bond et l'angle de contact, que nous prenons à 45 degré, angle du ménisque formé par l'eau dans un récipient en verre.

2 Méthodes numériques et algorithme

2.1 Intégration de Verlet

Pour déterminer nos coordonnées, vitesses et accélérations en fonction du temps nous avons opté pour l'intégration de Verlet. L'intégration de Verlet est un algorithme simple à mettre en place et qui permet de conserver l'énergie dans le système. L'algorithme utilise le développement limité de Taylor de notre vecteur position à l'ordre 3.

Démonstration du développement limité de Taylor Young de $f(x)$ au point x_0 [5] :

$$DL_n f(x) = \sum_{i=0}^n \frac{f^{(i)}(x_0)}{i!} (x - x_0)^i + o((x - x_0)^n) \quad (11)$$

Si on applique le développement limité d'ordre 3 à la position($\mathbf{x}(t+dt)$) au point $t+dt$ on a l'équation suivante avec t_0 comme le pas de temps précédent :

$$DL_3 \mathbf{x}(t) = \mathbf{x}(t_0) + \mathbf{x}'(t_0)(t - t_0) + \frac{\mathbf{x}''(t_0)}{2!} (t - t_0)^2 + o((t - t_0)^3) \quad (12)$$

Si t_0 est le pas de temps précédent, $\mathbf{x}'(t)$ la vitesse et $\mathbf{x}''(t)$ l'accélération, nous avons :

$$DL_3 \mathbf{x}(t + dt) = \mathbf{x}(t) + \mathbf{x}'(t)(t + dt - t) + \frac{\mathbf{x}''(t)}{2!} (t + dt - t)^2 + o((t + dt - t)^3) \quad (13)$$

$$DL_3 \mathbf{x}(t + dt) = \mathbf{x}(t) + \mathbf{v}(t)(dt) + \frac{\mathbf{a}(t)}{2!} (dt)^2 + o(dt^3) \quad (14)$$

L'erreur sur le temps t_n est de l'ordre de $o(e^{Lt_n} dt^2)$ avec un ordre de 2[6]. Avec L la constante de Lipschitz qui est la mesure du changement maximum d'une fonction dans une région dans laquelle il est défini.

Notre accélération ne dépend pas du changement de vitesse mais de l'équation (10), nous pouvons calculer l'accélération à partir du principe fondamental de la dynamique avec une masse constante. Il est important de faire cela après le calcul de position mais avant celui de la vitesse car la position prend l'accélération précédente mais la vitesse prend celle précédente et au même moment.

$$\sum \mathbf{F} = m\mathbf{a} \implies \mathbf{a} = \frac{\sum \mathbf{F}}{m} \quad (15)$$

Maintenant nous avons la nouvelle position et l'accélération, nous pouvons calculer la nouvelle vitesse de la même façon en utilisant le développement limité de Taylor Young à l'ordre 2.

$$DL_2 \mathbf{v}(t) = \mathbf{v}(t_0) + \mathbf{v}'(t_0)(t - t_0) + o((t - t_0)^2) \quad (16)$$

De la même façon que nous avons déterminé la position ($\mathbf{x}(t + dt)$)(13), nous l'appliquons à la vitesse et nous avons :

$$DL_2 \mathbf{v}(t + dt) = \mathbf{v}(t) + \mathbf{a}(t)(dt) + o(dt^2) \quad (17)$$

Comme nous connaissons l'accélération au pas de temps suivant et précédent en même temps, nous pouvons avoir une meilleure approximation de notre vitesse en prenant le point au milieu au lieu de prendre uniquement l'accélération précédente. On utilise la *velocity-Verlet* à la place de la *Störmer-Verlet* car avec cette dernière nous avons besoin des 2 positions précédentes et nous gardions

seulement un pas de temps à chaque fois. De plus, les deux méthodes ont une précision similaire.

$$\mathbf{v}(t + dt) = \mathbf{v}(t) + \frac{\mathbf{a}(t) + \mathbf{a}(t + dt)}{2} dt \quad (18)$$

Même si la méthode d'Euler et de Verlet paraissent similaire, nous n'avons pas opté pour la méthode d'Euler car avec celle-ci notre énergie n'était pas conservée. De plus, les erreurs sont plus hautes avec la méthode d'Euler qu'avec l'intégration de Verlet. Nous avons également décidé de ne pas opter pour la méthode de Runge-Kutta car, pouvant être dissipative, elle aurait demandé plus de calcul, malgré sa plus grande précision. Enfin, l'intégration de Verlet était la plus accessible et la seule à être réversible dans le temps[6].

2.2 Collisions

2.2.1 Collisions Objet-Objet

Pour les collisions, nous sommes partis sur un modèle assez simple qui itère chaque objet et regarde si la distance entre leurs centres est plus petite que leurs rayons additionnés. Si c'est le cas, nous disons qu'il y a collision entre eux et nous appliquons la collision avec la conservation du momentum. Nous avons mis en place les collisions entre deux objets mais également entre un objet et les bords. Le fonctionnement des collisions entre ces deux cas est très différent. Pour les collisions entre objets, nous prenons dans un premier temps le vecteur normé de collision, dans le sens de objet 1(A) vers objet 2(B) :

$$\mathbf{c} = \frac{\mathbf{AB}}{\|\mathbf{AB}\|} \rightarrow \|\mathbf{c}\| = 1 \quad (19)$$

Puis nous calculons la vitesse relative $\mathbf{v}_{rel} = \mathbf{v}_A - \mathbf{v}_B$ pour comprendre comment les 2 objets vont s'affecter. Après cela, nous trouvons la vitesse des objets lors de la collision afin de nous être utile pour déterminer l'impulsion qui suivra la collision :

$$v_{col} = \mathbf{v}_{rel} \cdot \mathbf{c} \quad (20)$$

Nous ajoutons à cette vitesse un coefficient compris entre 0.2 et 0.7 car nous n'avons pas de collisions élastiques parfaites. Il faut cependant faire attention à cette constante ; Si elle est trop basse, les objets n'auront pas le rebond nécessaire et vont commencer à s'entrecroiser. Si elle est trop haute, les objets vont, à l'inverse, beaucoup rebondir. Toutefois, plus notre pas de temps est petit, plus ces effets vont disparaître.

Le signe de la vitesse de la collision nous précise si les objets s'attirent ($v_{col} > 0$) ou s'ils s'éloignent ($v_{col} < 0$).

Si la vitesse de collision est plus grande que 0 ; nous appliquons la conservation de momentum.

$$I = \frac{2v_{col}}{m_A + m_B} \quad (21)$$

$$\mathbf{v}'_A = \mathbf{v}_A - Im_B \mathbf{c} \quad (22)$$

$$\mathbf{v}'_B = \mathbf{v}_B + Im_A \mathbf{c} \quad (23)$$

Avec les dérivées de nos nouvelles vitesses. On observe que lorsque les objets se touchent, ils ne rebondissent pas entre eux, pour cela après chaque rebond on multiplie la nouvelle vitesse avec un coefficient de collision (C_c , $0 \leq C_c \leq 1$). Quand C_c est égal à 1 nous avons des collisions élastiques parfaites et s'il vaut 0 nous avons des collisions inélastiques parfaites. Pour déterminer ce coefficient nous avons fait des essais et pris celui qui fonctionnait le mieux. Et nous avons remarqué que plus notre pas de temps (dt) est petit moins le coefficient de collision est important. Généralement, le C_c peut être pris entre 0.3 et 0.7.

2.2.2 Collision des bords

Pour les collisions de bord on a fait tel que si notre objet dépassait le bord de très peu nous inversons le vecteur vitesse par rapport à la normale. Et après nous multiplions ce vecteur par un coefficient de collision, $0 \leq C_c \leq 1$, qui simule l'absorption de l'énergie à chaque collision.

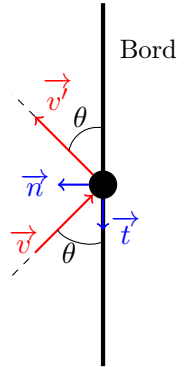


FIGURE 4 – Schéma d'un rebond d'un objet sur un bord

$$\mathbf{v} = (\mathbf{v} \cdot \mathbf{n})\mathbf{n} + (\mathbf{v} \cdot \mathbf{t})\mathbf{t} \quad (24)$$

Nous inversons le coefficient de la normale pour le faire 'rebondir'

$$\Rightarrow \mathbf{v}' = -(\mathbf{v} \cdot \mathbf{n})\mathbf{n} + (\mathbf{v} \cdot \mathbf{t})\mathbf{t} \quad (25)$$

3 Conception de notre algorithme

3.1 Notre algorithme

Voici un diagramme du fonctionnement de notre algorithme pour un seul objet (s'il y a x objets

alors on répète x-fois celui-ci). (5).

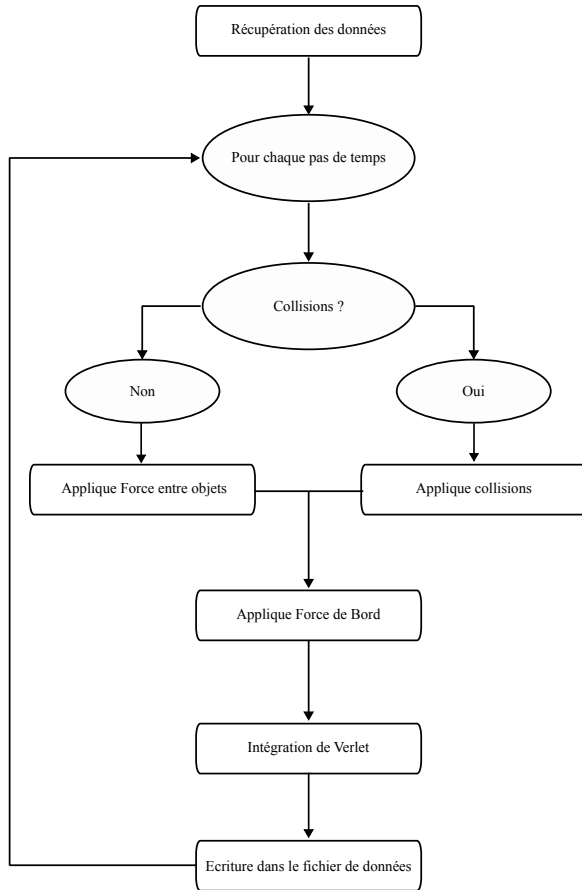


FIGURE 5 – Diagramme de notre algorithme pour un objet.

Nous avons essayé d’être le plus efficace dans notre programme et de limiter le nombre d’itération. Les données initiales sont à mettre dans un fichier texte, les données finales sont également mises dans un fichier texte. Ce dernier est lu par un script Python afin de créer l’animation avec matplotlib et sa classe *animate*. Pour représenter nos objets nous utilisons la classe *circle* de matplotlib également. Cela nous permet de définir des cercles avec des rayons précis.

3.2 Améliorations possibles

Notre code, actuellement n’est pas optimisé pour déterminer s’il y a des collisions entre les objets. En effet, nous regardons pour chaque objet s’il est en collision avec tous les autres objets présents. Cela nous donne un algorithme qui s’effectue en $O(NTn^2)$ (NT : nombre de pas de temps total dans la simulation, n : nombre d’objet), alors que nous pourrions l’effectuer en $O(NTn \log n)$. En utilisant un algorithme plus efficace. Seule l’amélioration dans la detection des collisions ne suffirait pas, car pour calculer les forces nous aurons

quand meme une complexité de $O(n^2)$ et notre temps d’exécution de notre programme augmenterait puisque nous faisons les calculs des forces et le test des collisions en même temps. Pour atteindre une complexité plus basse, nous aurons aussi besoin d’optimiser les calculs des forces. Pour cela, nous pouvons prendre un point dans lequel nous pouvons projeter toutes les forces et pour chaque particule prendre la force entre le point où les forces sont projetées.

Nous utilisons actuellement certaines équations qui sont approximatives, nous pourrions utiliser des équations plus précises comme l’approximation de Nicholson pour trouver un angle de contact plus précis et plus fiable.

Enfin, notre code fonctionne uniquement avec des objets sphériques de même taille. Nous pourrions ajouter les calculs pour différentes formes géométriques.

4 Nos Résultats

Pour tester notre programme nous avons fait l'expérience en vrai. Nous avons mis différents objets sphériques dans un bol rempli d'eau et nous avons filmer leurs mouvements. Nous avons ensuite analysé leurs coordonnées initiales avec Tracker et nous les avons implémentées dans notre code. Nous avons pu, après cela, analyser les différences entre la réalité et notre simulation. Voici sur les Figures suivantes nos comparaisons pour l'une d'elles.

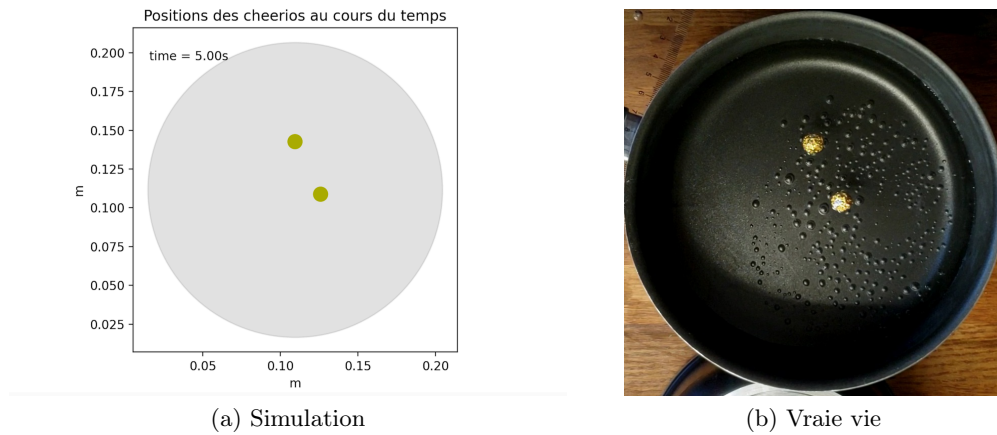


FIGURE 6 – A 5 secondes

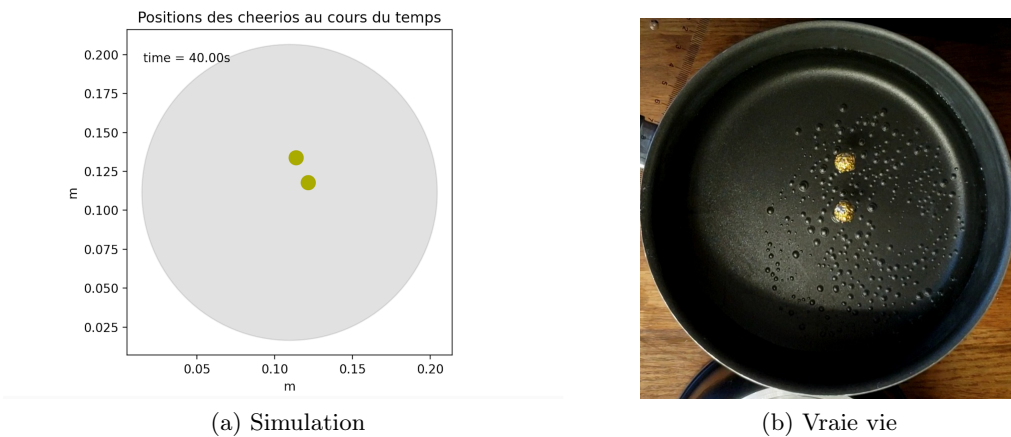


FIGURE 7 – A 40 secondes

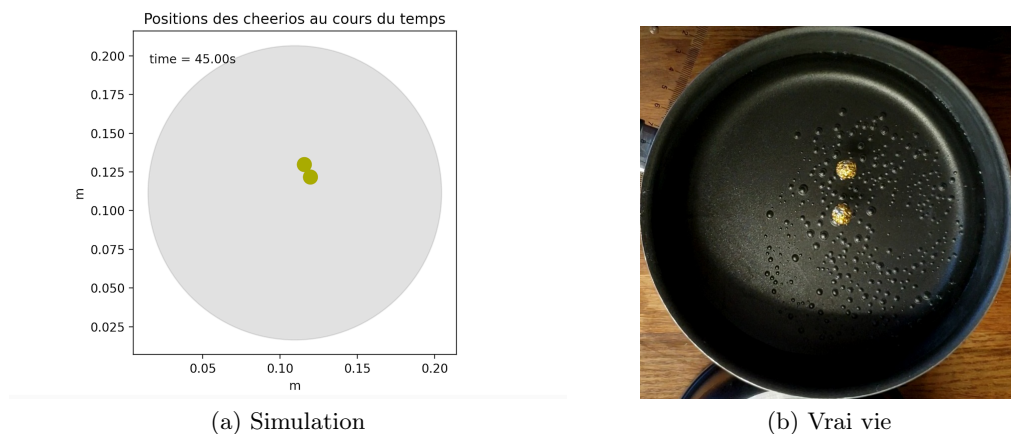


FIGURE 8 – A 45 secondes, la collision en vrai vie se passe en 55 secondes

Dans ces Figures, nous pouvons visualiser que la simulation n'est pas parfaite car il y a quelques différences avec les vidéos. L'une des raisons est que nous ne prenons pas en compte les légers écoulements d'air qu'il peut y avoir, qui ont des effets importants sur des objets flottants, de par leur faible stabilité. Néanmoins, nous pouvons être satisfait du résultat de nos simulations.

Conclusion

Pour conclure, nous avons réussi à analyser un problème de mécanique afin de créer une simulation de cet effet. Nous avons analysé le problème physique et l'avons résolu. Notre algorithme est fonctionnel et donne des résultats concluants. Cependant, notre code est fonctionnel jusqu'à une certaine durée. Après un certain temps la simulation devient instable. Néanmoins, nous avons su surpasser beaucoup de problème durant ce projet et nous avons pu développer nos techniques en programmation et notre savoir en mécanique des fluides, notamment avec notre recherche bibliographique qui eut une place importante dans ce projet.

Bibliographie

- [1] D. VELLA et L. MAHADEVAN, « The "Cheerios effect" », *American Journal of Physics*, t. 73, n° 9, p. 817-825, sept. 2005, ISSN : 0002-9505, 1943-2909. DOI : 10.1119/1.1898523. adresse : <http://aapt.scitation.org/doi/10.1119/1.1898523> (visité le 29/11/2022).
- [2] J. ONISHI, A. KAWASAKI, Y. CHEN et H. OHASHI, « Lattice Boltzmann simulation of capillary interactions among colloidal particles », *Computers & Mathematics with Applications*, t. 55, n° 7, p. 1541-1553, avr. 2008, ISSN : 08981221. DOI : 10.1016/j.camwa.2007.08.027. adresse : <https://linkinghub.elsevier.com/retrieve/pii/S0898122107006463> (visité le 29/11/2022).
- [3] G. K. BATCHELOR, *An introduction to fluid dynamics / by G.K. Batchelor,...* (Cambridge mathematical library), eng. Cambridge New York : Cambridge University Press, 2000, ISBN : 0-521-66396-2.
- [4] F. BOWMAN, *Introduction to Bessel functions / by Frank Bowman*, eng. New York : Dover Publications, 1958, ISBN : 0-486-60462-4.
- [5] R. P. AGARWAL, K. PERERA et S. PINELAS, *An Introduction to Complex Analysis*. New York : Springer, 2011, 331 p., ISBN : 978-1-4614-0194-0.
- [6] F. CRIVELLI, « The Störmer-Verlet method », p. 14, 8 mai 2008.
- [7] R. BENZI, S. SUCCI et M. VERGASSOLA, « Introduction to the Lattice Boltzmann Equation for Fluid Dynamics », in *Relaxation in Complex Systems and Related Topics*, sér. NATO ASI Series, I. A. CAMPBELL et C. GIOVANNELLA, éd., Boston, MA : Springer US, 1990, p. 329-334, ISBN : 978-1-4899-2136-9. DOI : 10.1007/978-1-4899-2136-9_45. adresse : https://doi.org/10.1007/978-1-4899-2136-9_45 (visité le 03/10/2022).
- [8] D. CHAN, J. HENRY et L. WHITE, « The interaction of colloidal particles collected at fluid interfaces », *Journal of Colloid and Interface Science*, t. 79, n° 2, p. 410-418, fév. 1981, ISSN : 00219797. DOI : 10.1016/0021-9797(81)90092-8. adresse : <https://linkinghub.elsevier.com/retrieve/pii/0021979781900928> (visité le 29/11/2022).
- [9] N. CHARLTON. « Drawing and Animating Shapes with Matplotlib ». (s. d.), adresse : <https://nickcharlton.net/posts/drawing-animating-shapes-matplotlib.html> (visité le 16/11/2022).
- [10] K. D. DANOV, R. DIMOVA et B. POULIGNY, « Viscous drag of a solid sphere straddling a spherical or flat surface », *Physics of Fluids*, t. 12, n° 11, p. 2711, 2000, ISSN : 10706631. DOI : 10.1063/1.1289692. adresse : <http://scitation.aip.org/content/aip/journal/pof2/12/11/10.1063/1.1289692> (visité le 29/11/2022).

- [11] H. N. DIXIT et G. M. HOMSY, « Capillary effects on floating cylindrical particles », *Physics of Fluids*, t. 24, n° 12, p. 122-102, déc. 2012, ISSN : 1070-6631, 1089-7666. DOI : 10.1063/1.4769758. adresse : <http://aip.scitation.org/doi/10.1063/1.4769758> (visité le 29/11/2022).
- [12] D.-x. FENG et A. V. NGUYEN, « Contact angle variation on single floating spheres and its impact on the stability analysis of floating particles », *Colloids and Surfaces A : Physicochemical and Engineering Aspects*, t. 520, p. 442-447, mai 2017, ISSN : 09277757. DOI : 10.1016/j.colsurfa.2017.01.057. adresse : <https://linkinghub.elsevier.com/retrieve/pii/S0927775717300948> (visité le 29/11/2022).
- [13] M. A. FORTES, « Attraction and repulsion of floating particles », *Canadian Journal of Chemistry*, t. 60, n° 23, p. 2889-2895, 1^{er} déc. 1982, ISSN : 0008-4042, 1480-3291. DOI : 10.1139/v82-414. adresse : <http://www.nrcresearchpress.com/doi/10.1139/v82-414> (visité le 29/11/2022).
- [14] W. GIFFORD et L. SCRIVEN, « On the attraction of floating particles », *Chemical Engineering Science*, t. 26, n° 3, p. 287-297, mars 1971, ISSN : 00092509. DOI : 10.1016/0009-2509(71)83003-8. adresse : <https://linkinghub.elsevier.com/retrieve/pii/0009250971830038> (visité le 29/11/2022).
- [15] J. B. KELLER, « Surface tension force on a partly submerged body », *Physics of Fluids*, t. 10, n° 11, p. 3009-3010, nov. 1998, ISSN : 1070-6631, 1089-7666. DOI : 10.1063/1.869820. adresse : <http://aip.scitation.org/doi/10.1063/1.869820> (visité le 29/11/2022).
- [16] P. A. KRALCHEVSKY et K. NAGAYAMA, « Capillary interactions between particles bound to interfaces, liquid films and biomembranes », *Advances in Colloid and Interface Science*, t. 85, n° 2-3, p. 145-192, mars 2000, ISSN : 00018686. DOI : 10.1016/S0001-8686(99)00016-0. adresse : <https://linkinghub.elsevier.com/retrieve/pii/S0001868699000160> (visité le 29/11/2022).
- [17] J. -. LOUDET, M. QIU, J. HEMAUER et J. J. FENG, « Drag force on a particle straddling a fluid interface : Influence of interfacial deformations », *The European Physical Journal E*, t. 43, n° 2, p. 13, fév. 2020, ISSN : 1292-8941, 1292-895X. DOI : 10.1140/epje/i2020-11936-1. adresse : <http://link.springer.com/10.1140/epje/i2020-11936-1> (visité le 29/11/2022).
- [18] E. H. MANSFIELD, H. R. SEPANGI et E. A. EASTWOOD, « Equilibrium and mutual attraction or repulsion of objects supported by surface tension », *Philosophical Transactions of the Royal Society of London. Series A : Mathematical, Physical and Engineering Sciences*, t. 355, n° 1726, p. 869-919, 15 mai 1997, ISSN : 1364-503X, 1471-2962. DOI : 10.1098/rsta.1997.0049. adresse : <https://royalsocietypublishing.org/doi/10.1098/rsta.1997.0049> (visité le 29/11/2022).
- [19] A. MARCHAND, J. H. WEIJS, J. H. SNOEIJER et B. ANDREOTTI, « Why is surface tension a force parallel to the interface ? », *American Journal of Physics*, t. 79, n° 10, p. 999-1008, oct. 2011, ISSN : 0002-9505, 1943-2909. DOI : 10.1119/1.3619866. adresse : <http://aapt.scitation.org/doi/10.1119/1.3619866> (visité le 29/11/2022).
- [20] J. C. MAXWELL, *The Scientific Papers of James Clerk Maxwell*, 1^{re} éd., W. D. NIVEN, éd. Cambridge University Press, 20 jan. 2011, ISBN : 978-1-108-01538-7. DOI : 10.1017/CB09780511710377. adresse : <https://www.cambridge.org/core/product/identifier/9780511710377/type/book> (visité le 29/11/2022).
- [21] M. M. NICOLSON, « The interaction between floating particles », *Mathematical Proceedings of the Cambridge Philosophical Society*, t. 45, n° 2, p. 288-295, avr. 1949, ISSN : 0305-0041, 1469-8064. DOI : 10.1017/S0305004100024841. adresse : https://www.cambridge.org/core/product/identifier/S0305004100024841/type/journal_article (visité le 29/11/2022).
- [22] « Calculateur de Tension Superficielle d'une Aiguille « Flottante » • Hydraulique – Fluides • Convertisseurs d'unités En Ligne ». (s. d.), adresse : <https://www.translatorscafe.com/unit-converter/fr-FR/calculator/surface-tension/> (visité le 07/11/2022).
- [23] « Compressible Lattice Boltzmann Method and Applications | SpringerLink ». (s. d.), adresse : https://link.springer.com/chapter/10.1007/978-3-642-11842-5_3 (visité le 01/10/2022).

- [24] *Computational Physics*. s. d. adresse : <https://link-springer-com.accesdistant.sorbonne-universite.fr/book/10.1007/978-3-319-00401-3> (visité le 01/12/2022).
- [25] T. ONDARÇUHU, P. FABRE, E. RAPHAËL et M. VEYSSIÉ, « Specific properties of amphiphilic particles at fluid interfaces », *Journal de Physique*, t. 51, n° 14, p. 1527-1536, 1990, ISSN : 0302-0738. DOI : 10.1051/jphys:0199000510140152700. adresse : <http://www.edpsciences.org/10.1051/jphys:0199000510140152700> (visité le 29/11/2022).
- [26] REDUCIBLE, director, *Building Collision Simulations : An Introduction to Computer Graphics*, 19 jan. 2021. adresse : https://www.youtube.com/watch?app=desktop&v=eED4bSkYCB8&ab_channel=Reducible (visité le 03/10/2022).
- [27] F. BOWMAN, *Introduction to Bessel functions / by Frank Bowman*, eng. New York : Dover Publications, 1958, ISBN : 0-486-60462-4.
- [28] N. B. VARGAFTIK, B. N. VOLKOV et L. D. VOLJAK, « International Tables of the Surface Tension of Water », *Journal of Physical and Chemical Reference Data*, t. 12, n° 3, p. 817-820, juill. 1983, ISSN : 0047-2689, 1529-7845. DOI : 10.1063/1.555688. adresse : <http://aip.scitation.org/doi/10.1063/1.555688> (visité le 29/11/2022).
- [29] N. D. VASSILEVA, D. van den ENDE, F. MUGELE et J. MELLEMA, « Capillary Forces between Spherical Particles Floating at a Liquid-Liquid Interface », *Langmuir*, t. 21, n° 24, p. 11 190-11 200, 1^{er} nov. 2005, ISSN : 0743-7463, 1520-5827. DOI : 10.1021/la051186o. adresse : <https://pubs.acs.org/doi/10.1021/la051186o> (visité le 29/11/2022).
- [30] D. VELLA et L. MAHADEVAN, « The 'Cheerios effect' », *American Journal of Physics*, t. 73, n° 9, p. 817-825, sept. 2005, ISSN : 0002-9505, 1943-2909. DOI : 10.1119/1.1898523. arXiv : cond-mat/0411688. adresse : <http://arxiv.org/abs/cond-mat/0411688> (visité le 29/11/2022).
- [31] D. VELLA, P. D. METCALFE et R. J. WHITTAKER, « Equilibrium conditions for the floating of multiple interfacial objects », *Journal of Fluid Mechanics*, t. 549, p. 215, -1 8 fév. 2006, ISSN : 0022-1120, 1469-7645. DOI : 10.1017/S0022112005008013. adresse : http://www.journals.cambridge.org/abstract_S0022112005008013 (visité le 29/11/2022).
- [32] D. J. R. VELLA, « The Fluid Mechanics of Floating and Sinking », p. 143, s. d.