

Table des matières

Table des figures

Liste des tableaux

Introduction

Dans le cadre de l'UE Projet en Calcul Scientifique Numérique, 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 du faire de nombreuses recherches sur la mécanique des fluides, les collisions inélastiques et nous avons également du faire un travail conséquent sur l'optimisation de notre algorithme.

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"[vella_cheerios_2005](#) 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.

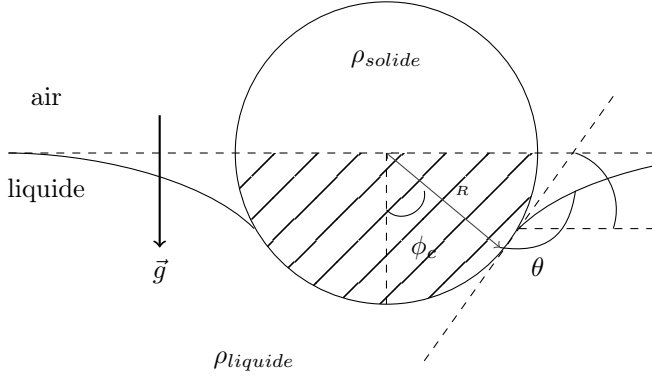


Figure 1. Géométrie d'une sphère reposant sur une interface liquide-gaz. La partie rayée représente le poids de liquide équivalent à la force de flottabilité due à la pression hydrostatique appuyant sur la sphère.

Une des raisons pour laquelle les objets flottent est due à la poussée d'Archimède, comme nous pouvons le voir dans la figure ???. Pour que notre 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 $\phi_c = \pi - \theta + \arctan z'_c/R$ 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 (??) contient deux paramètres sans dimensions ; le nombre de Bond B et Σ , qui sont très importants pour notre modélisation.

Le nombre de Bond vaut :

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

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.

JE TE LAISSE CETTE PARTIE MAIS FAIT AU PLUS SIMPLE, SURTOUT NIVEAU BESSEL, SULTAN LUI MEME A DIT QUE C'ETAIT POUR MASTER. MDR To calculate the interaction energy using the Nicolson approximation, we must also calculate the interfacial displacement caused by an isolated floating sphere, which is determined by the hydrostatic balance $\gamma \nabla^2 h = \rho g h$ — the co-ordinate invariant statement of equation (1). With the assumption of cylindrical symmetry, this becomes :

$$\gamma \frac{d^2 h}{dr^2} = \rho_l g h \quad (6)$$

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

$$\Rightarrow \frac{1}{r} \frac{d}{dr} \left(r \frac{dh}{dr} \right) = \frac{h}{L_c^2} \quad (7)$$

Nous pouvons déduire une solution de cette équation avec la fonction de Bessel modifiée à l'ordre 0 (K_0) **intrototoBessel**.

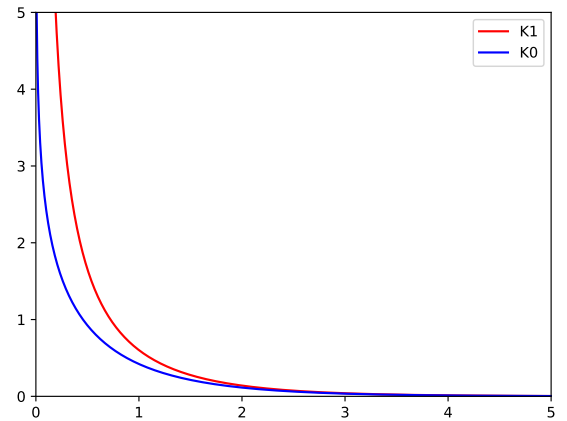


Figure 2. Graphique représentant les fonctions de Bessel modifiée à l'ordre 0 et 1

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 $2\pi RB\Sigma$. 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 :

$$E(l) = -2\pi\gamma R^2 b^2 \Sigma^2 K_0 \left(\frac{l}{L_c} \right) \quad (8)$$

Avec L_c la longueur capillaire.

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

$$F(l) = -2\pi\gamma RB^{5/2}\Sigma^2 K_1 \frac{l}{l_c} \quad (9)$$

1.2 Force 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 (??). 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???. 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.

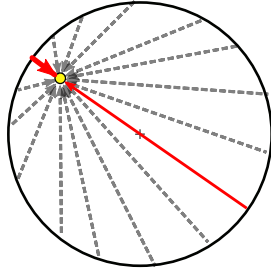


Figure 3. Schéma des forces des bords.

2. Méthodes numériques et algorithmes

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 [agarwal_introduction_2011](#) :

$$DL_n f(x) = \sum_{i=0}^n \frac{f^{(i)}(x_0)}{i!} (x - x_0)^i + o((x - x_0)^n) \quad (10)$$

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 :

$$\Rightarrow DL_3 \mathbf{x}(t) = \mathbf{x}(t_0) + \mathbf{x}'(t_0)(t - t_0) + \frac{\mathbf{x}''(t_0)}{2!} (t - t_0)^2 + o((t - t_0)^3) \quad (11)$$

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

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

$$\Rightarrow DL_3 \mathbf{x}(t+dt) = \mathbf{x}(t) + \mathbf{v}(t)(dt) + \frac{\mathbf{a}(t)}{2!} (dt)^2 + o(dt^3) \quad (13)$$

L'erreur sur le temps t_n est de l'ordre de $o(\exp(Lt_n)dt^2)$

Notre accélération ne dépendant pas du changement de vitesse mais de l'équation (??), 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.

$$\sum \mathbf{F} = m\mathbf{a} \Rightarrow \mathbf{a} = \frac{\sum \mathbf{F}}{m} \quad (14)$$

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.

$$DL_2 \mathbf{v}(t) = \mathbf{v}(t_0) + \mathbf{v}'(t_0)(t - t_0) + o((t - t_0)^2) \quad (15)$$

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

$$DL_2 \mathbf{v}(t+dt) = \mathbf{v}(t) + \mathbf{a}(t)(dt) + o(dt^2) \quad (16)$$

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 utilisant le théorème des accroissements finis. TODO ICI JE SUIS PAS DU TOUT SUR ???

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

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) :

$$\mathbf{c} = \frac{\mathbf{AB}}{\|\mathbf{AB}\|} \rightarrow \|\mathbf{c}\| = 1 \quad (18)$$

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 :

$$v_{col} = \mathbf{v}_{rel} \cdot \mathbf{c} \quad (19)$$

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'entrer dedans. 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 ($v_{col} < 0$). Il est possible

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

Impulse (I)

$$I = \frac{2v_{col}}{m_A + m_B} \quad (20)$$

$$\mathbf{v}'_A = \mathbf{v}_A - Im_B \mathbf{c} \quad (21)$$

$$\mathbf{v}'_B = \mathbf{v}_B + Im_A \mathbf{c} \quad (22)$$

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

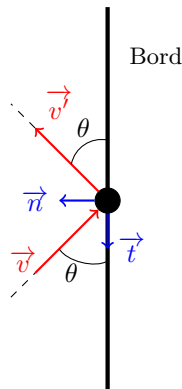


Figure 4. Schéma d'un rebond d'un objet sur un bord

l'énergie à chaque collision.

$$\mathbf{v} = (\mathbf{v} \cdot \mathbf{n})\mathbf{n} + (\mathbf{v} \cdot \mathbf{t})\mathbf{t} \quad (23)$$

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

$$\Rightarrow \mathbf{v}' = -(\mathbf{v} \cdot \mathbf{n})\mathbf{n} + (\mathbf{v} \cdot \mathbf{t})\mathbf{t} \quad (24)$$

3. Comment on a concue notre probleme

Voici un diagramme du fonctionnement de notre algorithme (??).

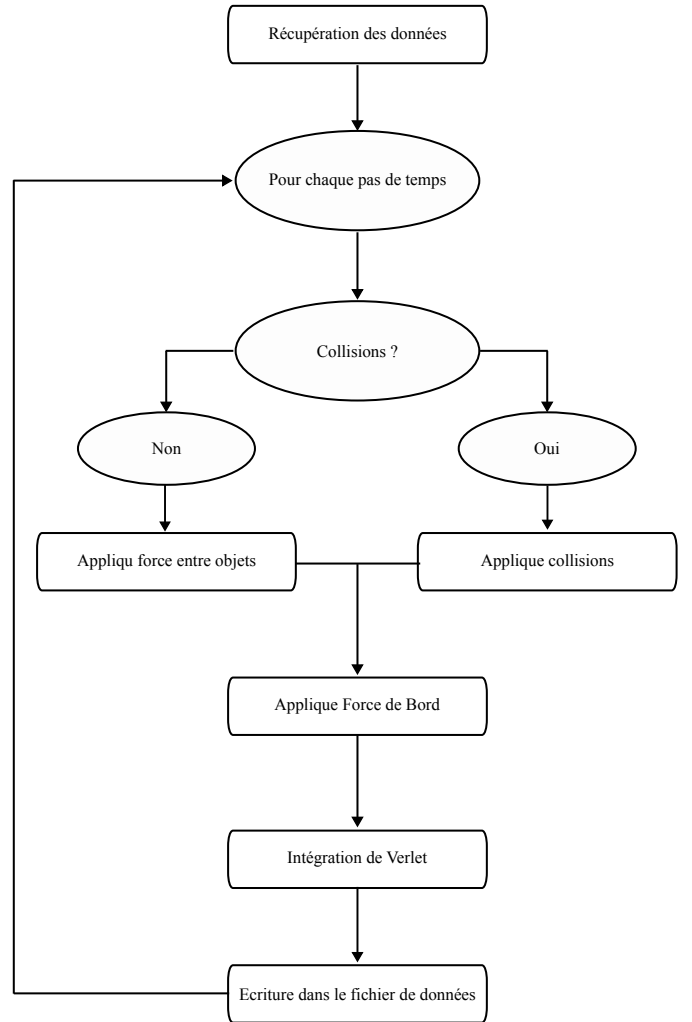


Figure 5. Diagramme de notre algorithme

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.

4. Les choses a ameliorer

- Code en $O(NTn^2)$ et peut etre ameliorer en $O(NTn \log n)$ en faisant le calcul de collisions plus intelligement a la place de une recherche exhaustive et en calculant une seule fois le *milieu* des forces de chaque particule pour avoir un centre de attraction et comme ca on calcule le centre de attraction regarde si on a des collisions ou pas et a la fin ajoute les forces du bords a chaque particule
- pour l'instant on utilise les equations *aproximatives* on peut les essayer de les resoudre sans approximations en utilisant l'approximation de Nicholson (fine difference method)
- le code marche seulement pour les objets ronds faux ajouter une facon plus complexe pour plus de objets

Conclusion