

Simulation d'écoulements à surface libre avec des particules

Objectif du projet (6h TD, 4h TDAO, 14h TP)

L'objectif du projet est de mettre en œuvre les notions et outils de la programmation orientée objet (classes, algorithmes, tableaux, listes, fichiers, interfaces graphiques) sur un problème concret et original donnant lieu à la réalisation d'un petit logiciel.

Les parties du projet abordées en TD constituent le socle *minimum* du projet. Il vous appartient ensuite de poursuivre votre projet par du travail personnel dans la direction de votre choix. Quelques pistes d'extensions sont données à titre d'exemple à la fin du sujet.

I Cadrage du projet

1 Contraintes et attendus

Les contraintes suivantes sont imposées :

- 1 le projet doit être réalisé en binôme (ou seul avec l'accord de votre enseignant de TP);
- 2 le projet doit être réalisé en Java avec Netbeans ;
- 3 le projet doit utiliser les notions et outils abordés dans le module ;
- 4 le projet doit comporter une interface permettant de tester rapidement les principales fonctionnalités du logiciel réalisé.

Le projet doit être rendu sous forme d'une archive compressée **zip** contenant :

- tous les fichiers du projet NetBeans (répertoire complet du projet NetBeans);
- un rapport en pdf (voir plus loin).

Le fichier **zip** doit s'intituler **NOM1-NOM2.zip** et doit être déposé sur **moodle**. La date butoir pour rendre le projet est indiquée sur **moodle**.

2

Programme de travail

TD (2h) Analyse des besoins et conception architecturale.

TP (6h) Première partie du projet.

Travail personnel

TD (4h) Approfondissements.

TP (4h) Deuxième partie de projet.

Travail personnel

TDAO (4h) Découverte des interfaces graphiques.

TP (4h) Réalisation d'une interface graphique permettant l'utilisation des algorithmes réalisés.

Travail personnel

3

Évaluation

L'évaluation du projet tient compte des éléments suivants :

- 1 résultats obtenus : étendue des fonctionnalités programmées, justesse des résultats, absence de bugs, etc. ;
- 2 utilisation adaptée des notions abordées dans le module : classes, structures de contrôle, tableaux, listes, arbres, récursivité, fichiers, interfaces graphiques, etc. ;
- 3 qualité de la programmation : lisibilité du programme (utilisation de noms pertinents pour les classes, les attributs, les méthodes et les variables), architecture du programme (découpage en différentes classes, répartitions des traitements dans les différentes classes, utilisation et ré-utilisation des méthodes), présence de tests ;
- 4 qualité du rapport écrit ;
- 5 qualité de la présentation orale ;
- 6 comportement en séance de TP (préparation, implication, sérieux, etc.).

4

Contenu du rapport

Un rapport d'une dizaine de pages doit accompagner le projet NetBeans et doit permettre à un informaticien ne connaissant pas votre sujet de comprendre le travail effectué. Par exemple, un autre groupe d'étudiant devrait être capable de poursuivre le projet.

La rapport doit présenter la démarche suivie pendant la réalisation de votre projet. Il pourra notamment lister les différentes fonctionnalités réalisées, décrire les classes Java sous la forme d'un diagramme de classes UML, présenter un algorithme participant à la réalisation d'une fonctionnalité importante du logiciel, donner un exemple d'utilisation du logiciel avec des captures d'écran.

Le rapport devra également inclure une conclusion présentant un bilan et une analyse critique du travail effectué, les limites du logiciel, ainsi que les perspectives d'amélioration ou d'extension. Dans le bilan, il est possible d'inclure un aspect de « répartition des tâches » si le projet a été réalisé en binôme.

II Objectif

Dans ce projet, vous allez simuler des écoulements à surface libre à l'aide de méthodes particulaires. Ces méthodes sont très utilisées en physique et dans les jeux vidéos pour simuler des liquides comme l'illustre la figure ci-dessous.

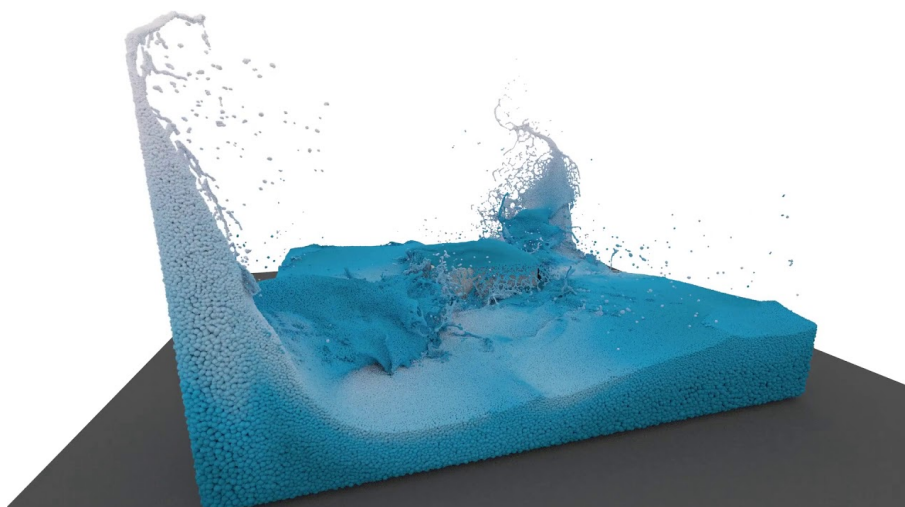


FIGURE 1.1 – Simulation d'une vague avec OpenMaelstrom.

Les méthodes particulaires permettent de représenter un écoulement par une multitude de particules en mouvement. Pour simplifier, vous traiterez le cas 2D où les particules se déplacent dans un plan (celui de l'écran). Les équations de base sont fournies pour que votre projet fonctionne bien mais vous pourrez naturellement l'enrichir avec les quelques pistes données à la fin du sujet.

Enfin, pour vous aider dans la réalisation du projet un certain nombre de classes sont fournies dans le projet NetBeans disponible sur Moodle, notamment la classe **Simulation** qui contient toutes les constantes et paramètres physiques pour que votre simulateur fonctionne bien.

III Une première version pleine de rebondissement

La première version de votre simulateur consistera à faire évoluer une particule unique soumise à la gravité.

Une particule est définie par sa masse m , son rayon h , ses coordonnées $\mathbf{r} = (r_x; r_y)$, sa vitesse $\mathbf{v} = (v_x; v_y)$ et son accélération $\mathbf{a} = (a_x; a_y)$. L'unité de longueur est le pixel.

Pour calculer l'accélération à chaque instant t , on utilise le principe fondamental de la dynamique :

$$\mathbf{a}(t) = \frac{1}{m} \sum \mathbf{f}(t)$$

Dans cette partie, nous considérons uniquement le poids et le frottement de l'air.

Pour intégrer les équations du mouvement entre deux instants t et $t + \tau$, on peut utiliser la méthode d'Euler qui considère l'accélération constante pendant le pas d'intégration τ pour calculer la nouvelle vitesse :

$$\mathbf{v}(t + \tau) = \mathbf{v}(t) + \tau \mathbf{a}(t)$$

puis considère la vitesse constante pour calculer la nouvelle position :

$$\mathbf{r}(t + \tau) = \mathbf{r}(t) + \tau \mathbf{v}(t)$$

Classiquement, on choisit le pas d'intégration τ en fonction de la célérité c du son dans le milieu (condition de Courant-Friedrichs-Lewy). La distance parcourue par une onde sonore pendant un pas d'intégration est égale à τc . Le pas d'intégration doit être suffisamment petit pour qu'une onde sonore évolue au maximum d'un rayon de particule à chaque pas d'intégration et ainsi :

$$\tau = \frac{h}{c}$$

Dans les fluides incompressibles, la célérité peut être très importante. En pratique, on choisit $c = 10v_{max}$ où v_{max} est la vitesse maximum d'une particule dans la simulation. Dans le cadre de votre projet, on peut considérer que la vitesse maximum est définie par la chute libre de la particule du haut en bas de la fenêtre, et donc :

$$v_{max} = \sqrt{2gH}$$

avec H la hauteur de la fenêtre.

Pour que la particule reste dans la fenêtre, on ajoute des collisions inélastiques avec les frontières du plan 2D. A chaque rebond, la particule est déviée et sa vitesse décroît avec un coefficient d'amortissement.

On ajoute également le frottement de l'air sur la particule, sous forme d'un simple frottement visqueux

$$f_{air} = -\beta.v$$

On peut enfin calculer l'énergie cinétique et potentielle de la particule pour s'assurer que le système est dissipatif.

IV Une deuxième version dans les nuages

Votre objectif est maintenant de simuler un nuage de particules avec un modèle de collision simple.

Un nuage est défini par une collection de particules. Dans cette partie, vous utiliserez une `ArrayList` pour stocker les particules (nous verrons plus tard que ce choix n'est pas le meilleur).

Au sein du nuage, les particules évoluent de façon autonomes en utilisant les équations précédentes mais des interactions se produisent quand elles entrent en collision. Quand une particule i se trouve à une distance inférieure à deux rayons d'une particule j , la particule j exerce une force répulsive sur la particule i égale à :

$$f_{collision} = \begin{cases} -k.(2h - d_{ij}).u_{ij} & \text{si } d_{ij} < 2h \\ 0 & \text{sinon} \end{cases}$$

où k est un coefficient de raideur, $d_{ij} = \|r_j - r_i\|$ la distance entre les particules et u_{ij} le vecteur unitaire du centre de la particule i vers le centre de la particule j .

$$u_{ij} = \frac{r_j - r_i}{\|r_j - r_i\|}$$

V Une troisième version qui voit les choses en grand

Vous avez sans doute remarqué que si vous créez quelques centaines de particules, la simulation devient très lente. Le problème vient du fait que le calcul de la densité et des forces d'interactions entre les particules nécessite $O(n^2)$ opérations à chaque pas d'intégration. Or, il est évident que la plupart de ces calculs sont inutiles car les particules ne sont en contact qu'avec au plus six autres particules.

Pour ne tester que les particules proches, il faut remplacer la liste des particules par une structure de donnée mieux adaptée qui vous sera présentée en TD.

VI Une quatrième version plus réaliste

Le modèle de collision précédent n'est pas très réaliste et pose des problèmes de stabilité numérique. Pour améliorer votre simulation, vous allez maintenant implanter la méthode SPH (smoothed-particle hydrodynamics ou hydrodynamique des particules lissées) qui est utilisée dans la plupart des logiciels de simulation d'écoulements à surface libre.

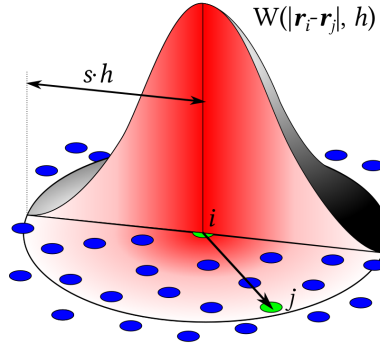


FIGURE 1.2 – Exemple de noyau de lissage à support compact (figure extraite de Wikipedia).

Dans la méthode SPH, la densité et la pression du fluide sont estimées localement à l'aide d'un noyau de lissage W (voir la figure 1.2). Dans le cadre de ce projet, vous utiliserez le noyau défini par une spline cubique par morceau :

$$W(d, h) = \frac{1}{S} \cdot \begin{cases} \frac{1}{6} \left(2 - \frac{d}{h}\right)^3 - \frac{4}{6} \left(1 - \frac{d}{h}\right)^3 & \text{pour } 0 \leq d \leq h \\ \frac{1}{6} \left(2 - \frac{d}{h}\right)^3 & \text{pour } h \leq d \leq 2h \\ 0 & \text{pour } 2h \leq d \end{cases}$$

avec S une constante dépendante de la dimension du problème. En 2D, on a $S = \frac{14}{30}\pi h^2$. Cette fonction ainsi que sa dérivée sont fournies dans la classe **Noyau**.

La densité du fluide à la position d'une particule i est ainsi estimée par la convolution :

$$\rho_i = \rho(\mathbf{r}_i) = \sum_j m_j W_{ij}$$

où $W_{ij} = W(\|\mathbf{r}_i - \mathbf{r}_j\|, h)$.

La pression relative est liée à la densité par l'équation de Cole :

$$P_i = \frac{\rho_0 c^2}{\gamma} \left(\left(\frac{\rho_i}{\rho_0} \right)^\gamma - 1 \right)$$

où ρ_0 est la densité de référence du fluide, c la célérité du son dans le fluide et γ son coefficient polytropique (pour l'eau $\gamma = 7$). La densité de référence correspond à la densité du fluide quand les particules sont empilées. En 2D, pour des particules de rayon h et avec le noyau défini ci-dessus, elle est égale à :

$$\rho_0 = \frac{m}{\pi h^2}$$

Une fois que l'on a calculé les densités et pressions de chaque particule, la force de la pression du fluide sur la particule i est donnée par :

$$\mathbf{f}_{\text{pression}} = m_i \mathbf{g} - m_i \sum_j m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) W'_{ij} \mathbf{u}_{ij}$$

où W'_{ij} est la dérivée de la fonction de lissage.

Quand votre simulation fonctionne bien avec la méthode SPH, vous pouvez ajouter la force liée à la viscosité du fluide exercée sur la particule i par une particule j quand $(\mathbf{v}_i - \mathbf{v}_j) \cdot \mathbf{u}_{ij} < 0$ (la force est nulle sinon) :

$$f_{viscosite} = m_i \sum_j m_j \alpha h c \left(\frac{\rho_i + \rho_j}{2} \right) ((\mathbf{v}_i - \mathbf{v}_j) \cdot \mathbf{u}_{ij}) W'_{ij} \mathbf{u}_{ij}$$

avec α un coefficient de viscosité artificielle.

VII Une cinquième partie à votre goût

Maintenant que votre simulateur fonctionne, vous pouvez faire plein de choses avec !

La liste suivante propose un certain nombre de pistes de travail dans les domaines de la modélisation et du rendu graphique. Vous êtes libres de poursuivre votre projet dans les directions de votre choix.

- Utiliser une ColorMap pour colorer les particules en fonction de leur pression ou de leur vitesse (voir la classe **Colormap** fournie dans le projet)
- Créer une interface pour contrôler les paramètres de la simulation (voir TD sur les interfaces graphiques)
- Ajouter des interactions avec la souris (voir TD sur les interfaces graphiques)
- Simuler le ballotement d'un liquide dans un réservoir (sloshing)
- Simuler un écoulement continu (avec une entrée de nouvelles particules et la sortie d'autres)
- Remplacer la méthode d'intégration d'Euler par une méthode plus précise (Runge-Kutta, Verlet, etc.)
- Ajouter d'autres modèles de viscosité ou de capillarité pour simuler d'autres écoulements (du miel par exemple)

VIII Annexe 1 : quelques liens pour vous aider

En plus de l'exposé de la section suivante, vous trouverez de nombreuses vidéos et sites expliquant la méthode SPH, comme par exemple :

- Vidéo sur l'intégration et la collision avec les bords (première moitié de la vidéo) : <https://www.youtube.com/watch?v=eED4bSkYCB8>
- Vidéo sur la méthode SPH : <https://www.youtube.com/watch?v=rSKMYc1CQHE>
- Vidéo sur la réduction de la complexité en utilisant une table : <https://www.youtube.com/watch?v=YhcC3eXiW88>
- Vidéo donnant quelques exemples de simulations possibles : <https://www.youtube.com/watch?v=Pj6okvh0wYw>
- Article de synthèse sur la méthode SPH : https://sph-tutorial.physics-simulation.org/pdf/SPH_Tutorial1.pdf

IX Annexe 2 : présentation plus détaillée de la méthode SPH

La méthode SPH est une méthode lagrangienne sans maillage qui permet de représenter un écoulement par une multitude de particules en mouvement.

1 Particules

La méthode SPH consiste à représenter un fluide par un ensemble d'éléments discrets, appelés particules. Chaque particule i est définie par sa position \mathbf{r}_i et sa vitesse \mathbf{v}_i . En coordonnées cartésiennes 2D, on a naturellement :

$$\mathbf{r}_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

et :

$$\mathbf{v}_i = \frac{d\mathbf{r}_i}{dt} = \begin{pmatrix} \dot{x}_i \\ \dot{y}_i \end{pmatrix}$$

2 Estimation de la densité avec un noyau de lissage

Pour calculer un champ continu (comme la densité et la pression) sur un ensemble de particules discrètes, la méthode SPH utilise une "moyenne glissante" définie par une convolution. De manière générale, la convolution d'un noyau de lissage W avec un champ continu A donne une moyenne locale de ce champ :

$$\bar{A}(\mathbf{r}) = \int_{\Omega} A(\mathbf{r}') W(\|\mathbf{r} - \mathbf{r}'\|, h) dV(\mathbf{r}')$$

h est le rayon du noyau de lissage et \mathbf{r} une position. Dans le cas de particules discrètes, cette intégrale peut être approchée par une somme de Riemann :

$$A(\mathbf{r}) = \sum_j A_j W(\|\mathbf{r} - \mathbf{r}_j\|, h) V_j$$

où V_j est le volume de la particule j , A_j est la valeur de la quantité A pour la particule j . Dans la méthode SPH, on utilise cette somme pour estimer la densité en un point \mathbf{r} :

$$\rho(\mathbf{r}) = \sum_j \rho_j W(\|\mathbf{r} - \mathbf{r}_j\|, h) V_j = \sum_j m_j W(\|\mathbf{r} - \mathbf{r}_j\|, h)$$

où $m_j = \rho_j V_j$ est la masse de la particule j et ρ_j sa densité. La densité à la position d'une particule i est ainsi définie par :

$$\rho_i = \rho(\mathbf{r}_i) = \sum_j m_j W_{ij}$$

où $W_{ij} = W(\|\mathbf{r}_i - \mathbf{r}_j\|, h)$.

Les noyaux de lissage communément utilisés dans la méthode SPH sont la fonction gaussienne, les splines et le noyau de Wendland. Contrairement à la gaussienne qui est définie sur $[0; +\infty[$, les deux derniers ont un support compact $[0; 2h]$. Cela présente l'avantage d'économiser des calculs en n'incluant pas les contributions relativement mineures des particules éloignées.

La fonction $W(d, h)$ doit être définie sur \mathbb{R}^+ , positive, continue, monotone décroissante, deux fois continûment différentiable, et vérifier $\int_{\Omega} W(d, h) dV = 1$.

La pression relative est liée à la densité par l'équation de Cole :

$$P = \frac{\rho_0 c^2}{\gamma} \left(\left(\frac{\rho}{\rho_0} \right)^\gamma - 1 \right),$$

où ρ_0 est la densité de référence du fluide, c la vitesse du son dans le fluide et γ son coefficient polytropique (pour l'eau $\gamma = 7$).

La densité de référence correspond à la densité du fluide quand les particules sont empilées. En 2D et pour des particules de rayon h , elle est égale à :

$$\rho_0 = \frac{m}{\pi h^2}$$

La vitesse du son dans des fluides pratiquement incompressibles comme l'eau est très importante de l'ordre de 10^3 m/s. Par conséquent, les ondes de pression se déplacent rapidement par rapport à l'écoulement réel, ce qui nécessite d'utiliser des pas d'intégration temporelle très faibles. En pratique, on adopte une vitesse "numérique" du son telle que des variations de densité inférieures à 1% soient autorisées. C'est ce que l'on appelle l'hypothèse de faible compressibilité.

$$c = 10v_{\max}$$

où v_{\max} est la vitesse maximale des particules dans la simulation.

Le Lagrangien du système est la différence entre la somme des énergies cinétiques des particules et la somme des énergies potentielles liées à la gravité et à la pression :

$$\mathcal{L} = E_c - E_p = \sum_j \frac{1}{2} m_j v_j^2 + m_j \mathbf{g} \cdot \mathbf{r}_j - P_j V_j = \sum_j \frac{1}{2} m_j v_j^2 + m_j \mathbf{g} \cdot \mathbf{r}_j - \frac{m_j P_j}{\rho_j}$$

On peut ensuite écrire l'équation d'Euler-Lagrange pour chaque particule i :

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \mathbf{v}_i} = \frac{\partial \mathcal{L}}{\partial \mathbf{r}_i}$$

Appliqué au Lagrangien du système, on obtient l'équation différentielle suivante :

$$m_i \frac{d\mathbf{v}_i}{dt} = m_i \mathbf{g} + \sum_j m_j \frac{P_j}{\rho_j^2} \frac{\partial \rho_j}{\partial \mathbf{r}_i}$$

Or, pour une particule j différente de i , on a :

$$\frac{\partial \rho_j}{\partial \mathbf{r}_i} = \frac{\partial}{\partial \mathbf{r}_i} \left(\sum_k m_k W_{jk} \right) = m_i \nabla W_{ji}$$

Et pour la particule i :

$$\frac{\partial \rho_i}{\partial \mathbf{r}_i} = \frac{\partial}{\partial \mathbf{r}_i} \left(\sum_j m_j W_{ij} \right) = m_i \nabla W_{ii} - \sum_j m_j \nabla W_{ij}$$

donc :

$$m_i \frac{d\mathbf{v}_i}{dt} = m_i \mathbf{g} + \sum_j m_j \frac{P_j}{\rho_j^2} m_i \nabla W_{ji} - m_i \frac{P_i}{\rho_i^2} \sum_j m_j \nabla W_{ij}$$

Et enfin comme $\nabla W_{ji} = -\nabla W_{ij}$:

$$\frac{d\mathbf{v}_i}{dt} = \mathbf{g} - \sum_j m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla W_{ij},$$