

PROJET ÉCOULEMENT MULTIPHASIQUE/INTERACTION FLUIDE-STRUCTURE

CODE DE CALCUL : BBAMR

novembre 2020

CROGUENNEC Guillaume

DUPONT Ronan

Enseignant : M. Golay

Table des matières

1	Introduction	2
2	Problème physique	3
3	Résolution numérique	3
4	Code de calcul	4
4.1	Les différents paramètres du fichier <code>bbamr.inp</code>	4
4.1.1	Paramètres physiques	4
4.1.2	Domaine de la simulation et discrétisation du maillage	5
4.1.3	Condition initiales	5
4.1.4	Conditions limites	5
4.1.5	Paramètres de maillage	5
4.1.6	Paramètres des solides	5
4.1.7	Paramètres numériques	6
4.2	Choisir les bons paramètres	6
4.2.1	Le pas de temps	6
4.2.2	Le sharpening	6
4.2.3	Influence VCDE/VCRA	7
5	Projet	8
5.1	Condition initiales	8
5.2	Paramètres de maillage	8
5.3	Paramètres numériques	9
5.3.1	TF	9
5.3.2	Le sharpening	9
5.3.3	VCDE/VCRA	9
6	Conclusion	9

1 Introduction

Dans un monde où les nouveaux défis sont de révolutionner nos industries pour préserver l'environnement, les énergies marines sont plus que jamais au centre de l'attention. Mais cela nécessite d'être capable de maîtriser les interactions entre les fluides et les structures.

Interaction qui néanmoins attirent depuis longtemps l'intérêt des scientifiques tant elles sont importantes dans de nombreux domaines, comme les barages, l'aéronautique, le navale, l'offshore et tant d'autres.

Pour observer ces interactions et comprendre ce qu'elles peuvent impliquer, le meilleur moyen est de les simuler à l'aide de modèles numériques.

Dans un premier temps, il faut donc commencer par trouver des hypothèses physiques, qui sont générales à toute interaction entre fluides et structures. Pour ensuite les traduire en équations, afin les résoudre numériquement.

Le résultat est donc un code, développé par M. Golay et M. Altazin, qui résout numériquement le modèle choisi avec les conditions initiales et limites imposées. Dans les conditions initiales, sont pris en compte la position initiale de chaque fluide et solide étudiés, ainsi que le maillage initial, qui s'adaptait automatiquement.

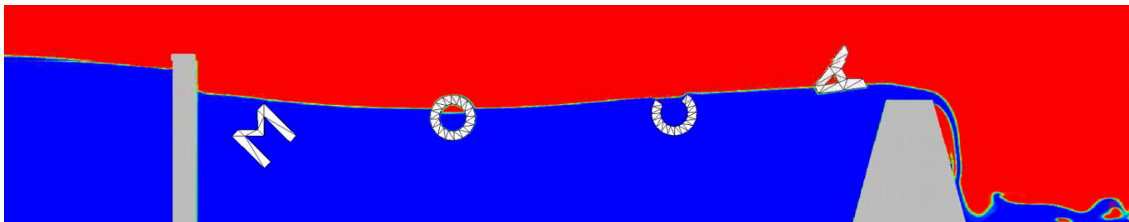


FIGURE 1 – Image obtenu par le code de calcul BBAMR - source : site de M. Golay : moca-big

2 Problème physique

Tout d'abord, il est nécessaire d'avoir des hypothèses physique qui sont toujours respectées et qui peuvent servir à décrire le phénomène. La première hypothèse est la conservation de la masse, c'est-à-dire le faire qu'il n'y a ni masse qui apparaît, ni masse qui disparaît. La deuxième hypothèse est celle de la conservation de quantité de mouvement, avec la même explication que pour la masse. La troisième hypothèse est celle de la conservation de l'énergie. Et la dernière est la loi d'état du système.

Ces quatre hypothèses forment donc un système d'équations, qui ne vont pas être les mêmes suivant le problème. En effet, le code peut utiliser 7 systèmes d'équations différents, chacun adapté à un type de problème physique. Les équations d'Euler et de Navier-Stokes sont utilisées pour les problèmes à un seul fluide, les équations k-epsilon pour les problèmes de turbulence, les équations granulaires pour les problèmes avec des granules dont le comportement peut se rapprocher d'un fluide, et les équations diphasiques isothermes, une énergie et deux énergies pour des problèmes à deux fluides.

Les cas étudiés mettant en scène deux fluides, le système des équations diphasiques isothermes est choisi. Il s'agit donc du système qui correspond au cas physique où il y a deux fluides, mais considérés comme incompressibles ou quasiment incompressibles, car dans le cas contraire une équation d'état serait nécessaire pour chaque fluide, et la différente vitesse de propagation du son dans les deux milieux entraînerait des contraintes sévères sur le pas de temps, cf Prigent. Cette incompressibilité induit donc une isothermie. D'où des équations isothermes. Et le problème de modélisation d'un solide en utilisant des équations diphasiques peut être contourné en considérant qu'un solide n'est qu'un fluide rigide.

Les équations utilisées sont donc :

$$\begin{cases} \partial_t \rho + \partial_x \rho u = 0 \\ \partial_t(\rho u) + \partial_x(\rho u^2 + p) = 0 \\ \partial_t(\rho E) + \partial_x(\rho E + p) = 0 \\ p = (\sigma - 1)\rho(E - \frac{u^2}{2}) \end{cases}$$

De plus, on introduit également l'entropie dans cette partie, car elle va être utilisé dans la suite afin de mailler correctement la simulation :

$$s = \rho u^2 + c_0^2 \rho \ln(\rho) - c_0^2(\rho_w - \rho_A)\varphi$$

3 Résolution numérique

Pour ce qui est de la résolution numérique, la méthode des volumes finis est utilisée. C'est une méthode qui se construit naturellement en observant que la valeur d'une grandeur en un point vaut la valeur au moment précédent moins le flux de sortie plus le flux d'entrée depuis ce moment. Bien sûr ce n'est valable que quand il y a conservation de la grandeur considérée. Dans ce cas, cette méthode des volumes finis est appliquée à plusieurs grandeurs à la fois.

Pour calculer numériquement ces flux, le code permet d'utiliser quatre modèles : Rusanov, Godunov, Flux centre et VF-Roe.

Le schéma numérique du Flux centre est très peu précis, et celui de Roe n'est pas entropique, c'est-à-dire que lorsqu'il y a plusieurs solutions mathématiques il ne choisit pas forcément la

seule qui est physiquement possible. C'est pour cela que le schéma de Godunov est choisi pour les cas étudiés, car il est le plus précis.

Une fois ces flux calculés, il faut ensuite utiliser un autre schéma numérique pour l'intégration en temps. Avec le code il est possible d'utiliser les schémas numériques de Runge-Kutta et de Adams Bashforth avec ou sans pas de temps local.

Celui de Runge-Kutta est choisi car il demande moins de calculs et est amplement suffisant pour les cas étudiés.

4 Code de calcul

Le code de calcul est un ensemble de fichiers avec des centaines de sous-routines reliés entre elles. Quand on compile le programme, il va chercher un ensemble de paramètres qui sont regroupés dans un fichier `bbamr.inp`.

4.1 Les différents paramètres du fichier `bbamr.inp`

Tous les paramètres sont donc regroupés dans cet unique fichier, qui les séparent en 7 parties, suivant leur nature. Pour expliquer les différentes parties, nous nous baserons sur le fichier `dam4.inp` qui effectue la simulation suivante :

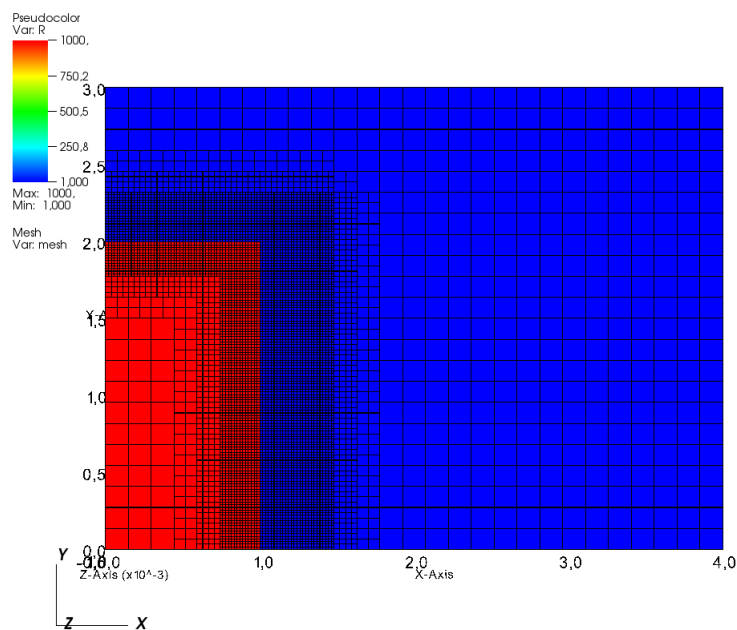


FIGURE 2 – Première image de la simulation de `dam4.inp`

4.1.1 Paramètres physiques

Les premiers paramètres sont des constantes physiques qui décrivent les deux fluides que l'on utilise, comme leur viscosité, leur conductivité thermique ou encore leur densité. Dans les cas étudiés, seuls l'eau et l'air sont utilisés donc il n'y a pas besoin de les modifier. Mais il est possible de le faire pour simuler d'autres fluides que ces deux-là. De plus, la pesanteur est également un des paramètres de cette partie. Donc il serait possible d'effectuer une simulation d'un phénomène sur une autre planète avec une autre pesanteur.

4.1.2 Domaine de la simulation et discrétisation du maillage

Les paramètres suivants servent à définir si la simulation est en une dimension, deux dimensions ou trois dimensions. Ainsi qu'à choisir les mesures du domaines utilisés, et en combien de points chacune des directions va être discrétisée.

Pour ce qui est du nombre de dimensions de la simulation, la technique utilisée est en fait toujours la même. Simplement, dans tous les cas c'est une domaine en trois dimensions qui est utilisé. Mais quand on veut une dimension en moins, le domaine va simuler ça en étant très fin selon cette dimension, et en considérant que cette direction est discrétisée en 0 points.

4.1.3 Condition initiales

Par conditions initiales on entend remplir initialement le domaine par des fluides. Et le code fonctionne de manière à ce que si on met un fluide par dessus un autre, c'est le dernier mis qui prend la place. C'est-à-dire que si on veut de l'eau à un endroit, et de l'air dans le reste du domaine ; on commence par mettre de l'air dans tout le domaine, puis on rajoute de l'eau là où on le veut, et cela va enlever l'air à cet endroit.

Pour définir où on veut mettre un fluide, il faut rentrer les six coordonnées du domaine (le plus simple est de faire des pavés droits), ainsi que sa densité, les composantes de sa vitesse et sa pression. Par ailleurs, la densité du fluide doit forcément être comprise entre les deux densités des fluides considérés pour le modèle diphasique isotherme (donc l'air et l'eau).

4.1.4 Conditions limites

Pour les bords du domaines, il est possible d'imposer plusieurs conditions limites. Elles peuvent être des conditions de sortie, c'est-à-dire que le domaine est ouvert, donc les fluides peuvent en sortir. Elles peuvent aussi être des conditions de miroir, c'est-à-dire que lorsque le fluide arrive contre une paroi, la composante de sa vitesse qui est orthogonale au bord est inversée. Il est aussi possible de mettre des conditions d'adhérence si le modèle est visqueux, ou encore de nombreuses autres conditions.

Dans les cas étudiés, les conditions limites miroirs sont utilisées.

4.1.5 Paramètres de maillage

Dans cette partie du code, il faut tout d'abord définir le maillage initial. Le maillage a un coefficient de raffinement, et plus il est élevé, plus le maillage est fin.

Pour les zones du maillage initial, cela fonctionne comme les conditions initiales sur les fluides. Si on met un maillage sur un autre, c'est le dernier mis qui prévaut.

Une fois que ce maillage initial est mis, il faut choisir jusqu'à quel raffinement de maillage la simulation peut aller durant les calculs (pas plus de quatre pour les ordinateurs à quatre cœurs). Pour savoir où raffiner le maillage, le code calcule l'entropie dans tout le domaine. Plus il y en a à un endroit, plus il faut que le maillage y soit fin.

Dans notre exemple, on a 4 niveaux de raffinements et 3 domaines de maillage initiaux :

- Le domaine de toute la "2d box" avec un niveau de raffinement 0
- Le domaine carré qui parcourt $x \in [0, 1.5]$, $y \in [0, 2.5]$ avec un niveau de raffinement 3
- Le domaine carré qui parcourt $x \in [0, 0.75]$, $y \in [0, 1.75]$ avec un niveau de raffinement 0

Les deux derniers domaines permettent de laisser un raffinement de niveau 3 seulement sur l'interface comme on peut voir sur la figure 2.

4.1.6 Paramètres des solides

Pour mettre des solides dans cette simulation qui utilise un modèle bifluide, il est nécessaire d'utiliser un fluide qui peut bien représenter un solide. Pour cela, une rigidité est imposée au

fluide, qui ne se déformera donc pas durant la simulation. Néanmoins, il est toujours nécessaire que sa densité soit comprise entre celles des deux fluides de référence dans le programme (l'air et l'eau).

Et pour les définir, un fichier objet est affilié à chaque solide, où chacun de ses points initiaux est défini, ainsi que les triangles qui forment ses faces.

4.1.7 Paramètres numériques

Dans cette partie, on trouveras tous les paramètres numériques pour avec des calculs plus ou moins optimisés. Il faut par exemple choisir le pas de temps et le CFL. Dans les cas étudiés, le CFL utilisé est 0.7, et le pas de temps va être choisi suivant le cas.

4.2 Choisir les bons paramètres

Pour certains paramètres, il est nécessaire d'essayer au moins une fois la simulation pour ensuite pouvoir choisir les bons paramètres.

4.2.1 Le pas de temps

Pour choisir notre pas de temps, nous avons besoin de la vitesse du fluide. Pour l'obtenir, on regarde la vitesse de la console affichée. Celle-ci est la somme de la vitesse du fluide et de la vitesse du son choisie. En soustrayant cette vitesse de son, on obtient la vitesse du fluide de l'ordre de $4m/s$. Comme on a $T = \frac{D}{V}$, et a $D = \text{longueur en } x / \text{nb pts en } x = \frac{4}{60}$ d'où $T_{max} = \frac{4}{27 \cdot 4} = 0.037s$. On a donc utilisé la valeur : $T_{max} = 0.025s$.

4.2.2 Le sharpening

Le SHAR (sharpening) correspond au raidissement d'interface. Ce coefficient permet au maillage de "rattraper" le déferlement de la vague. Mais surtout d'affûter le maillage pour avoir des interface plus fines entre les fluides. Car sur les images le jaune et vert entre les fluides n'est pas un mélange de fluide, mais de la diffusion, qui peut être empêchée grâce au sharpening. Sans SHAR, le maillage peut donc ne pas avoir le temps de "se déplacer" aux endroits où il est nécessaire d'avoir un maillage plus fin comme à l'interface : on peut voir ceci sur la figure 3.

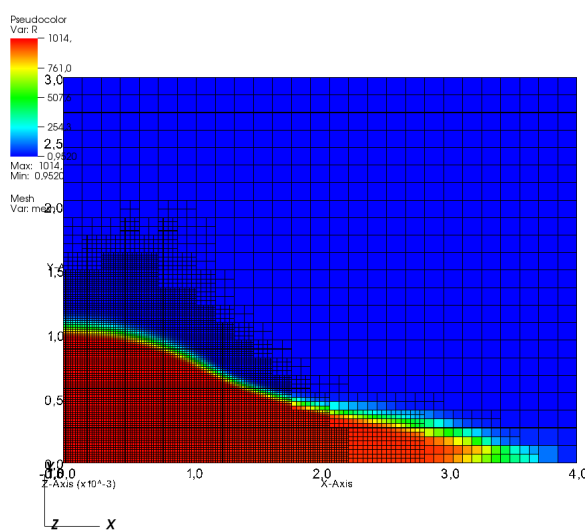


FIGURE 3 – Simulation `dam4.inp` sans paramètre de SHAR

En ajoutant un coefficient SHAR=0.05, on obtient la simulation suivante figure 4.

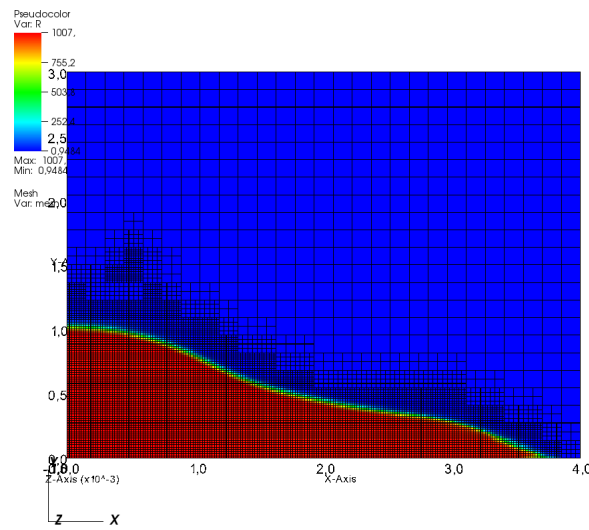


FIGURE 4 – Simulation `dam4.inp` avec paramètre de SHAR=0.05

4.2.3 Influence VCDE/VCRA

Les critères VCDE (Mesh coarsening parameter) et VCRA (Mesh refinement parameter) permettent de raffiner plus ou moins le maillage sur des zones données. Pour se faire, on ouvre les fichiers blocs et on affiche le "CRIT". En faisant varier le minimum et le maximum, on arrive à retirer des zones qui n'ont pas besoin d'être raffinées de manière très fines comme la zone bleu qu'on peut voir sur la figure 5 à gauche. Si l'on se base sur les paramètres de la figure de gauche où on a pris 0.5 en minimum et 1 en maximum, on règle VCDE=min et VCRA=max et on obtient la figure de droite.

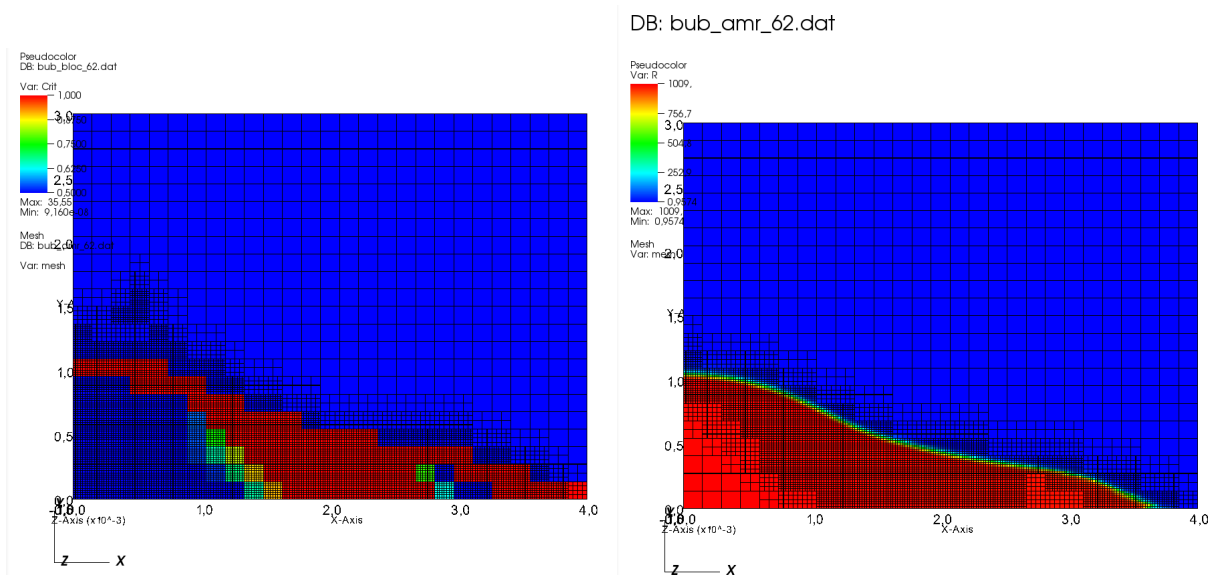


FIGURE 5 – Maillage avec ou sans critères VCDE/VCRA sur `dam4.inp`

On remarque que ce maillage est plus optimisé car il est mieux raffiné aux interfaces sur la figure 5 que la figure 4.

5 Projet

Pour notre projet, nous avons décidé d'effectuer la simulation suivante figure 6 :

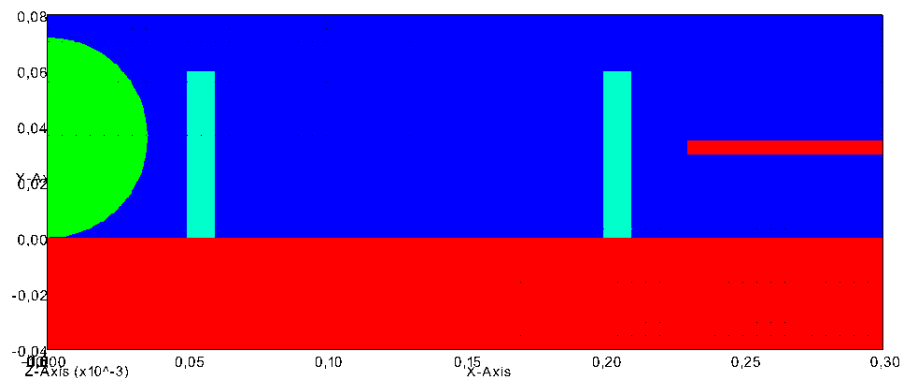


FIGURE 6 – Première image de notre simulation

L'idée est de placer deux "solides" sur la surface de l'eau puis de les faire basculer d'un côté par une vague créée par un cylindre ayant un mouvement vertical vers le bas ; de l'autre par un jet d'eau arrivant horizontalement sur le "solide".

5.1 Condition initiales

Comme on peut le voir figure 6, nous avons introduit les conditions initiales suivantes :

- Un bassin d'eau avec une interface en $y = 0$.
- Deux planches en équilibre à l'interface de l'eau :
Pour cela nous avons récupéré un parallélépipède régulier et nous avons changé les coordonnées afin d'avoir les deux planches comme on peut voir 6.
- Un cylindre avec un mouvement vertical permettant la création d'une vague :
Pour cela nous avons récupéré le fichier objet de la sphère sur `bbarm_solide`.
- Un jet d'eau horizontal : Nous lui mettons une vitesse initiale de $-1m/s$.

5.2 Paramètres de maillage

Pour notre maillage, nous l'avons discrétisé de la manière suivante comme on peut voir ci-dessous figure 7

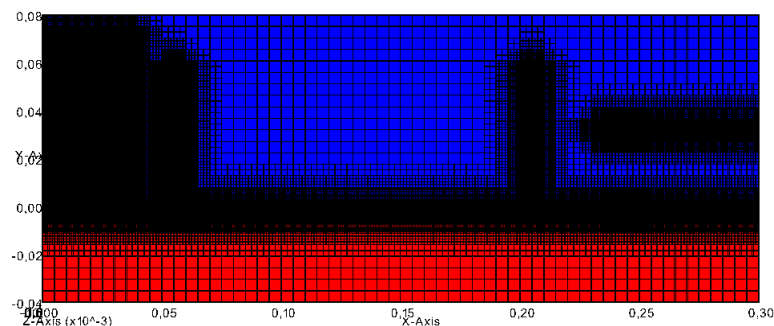


FIGURE 7 – Première image du maillage de notre simulation

On remarque qu'on a plusieurs zones particulières :

- L'ensemble du domaine maillé avec un niveau de raffinement 4.

- L'interface maillé avec une niveau de raffinement 4.
- Les planches maillés avec une niveau de raffinement 4.
- La sphère maillé avec un niveau de raffinement 4.

5.3 Paramètres numériques

5.3.1 TF

Pour calculer notre Tmax, nous avons besoin de savoir la vitesse maximale du fluide. Il faut savoir que dans notre problème, elle est maximale dans le sens des x due au fait que nous avons un jet horizontale et le vague crée par la sphère a aussi une vitesse selon x.

On reprend donc la partie 4.2.1 et on calcul directement : $Tmax = \frac{L}{v_x \cdot nx} = \frac{0.30}{2 \cdot 60} = 0,0025s$.

On prend donc Tmax=0.0025s.

5.3.2 Le sharpening

Nous avons dans un premier temps utilisé un SHAR à 0.005 mais le calcul se bloquait à cause d'une vitesse de fluide trop grande. En effet, il aurait fallu un très petit pas de temps, qui, couplé au sharpening rendait le calcul trop lourd. Nous avons donc décidé de retirer le SHAR.

5.3.3 VCDE/VCRA

Comme nous avons procédés dans la partie 4.2.3, nous avons tracés le CRIT afin d'optimiser notre maillage pour le rendre optimal au niveau des interfaces. Nous avons donc pris un VCDE=0.05 et un VCRA=0.2.

6 Conclusion

Ainsi, une fois les hypothèses physiques transformées en équations, la première étape qui a été nécessaire pour construire ce code est de les résoudre numériquement. La méthode des volumes finis est la plus naturelle, mais il faut également choisir les bonnes méthodes numériques pour les flux ainsi que l'intégration temporelle. Ensuite, l'autre principale problème dans ce code est d'obtenir un maillage adaptatif. Pour cela, ses créateurs ont utilisé l'entropie.

Pour pouvoir utiliser ce code, il n'a pas fallu que nous en comprenions chaque sous-routine, mais plutôt comment utiliser chaque paramètre, et ce qu'ils représentent. De plus, il a fallu comprendre ce qui est possible ou pas avec le programme pour respecter le physiquement correct. Ensuite, nous avons appliqué ce code sur un projet visant à déséquilibrer deux planches de bois placées initialement à la surface de l'eau. Cela nous a permis d'observer la dynamique des fluides régit par les équations d'Euler bi-fluide, ainsi que leurs interactions avec des structures solides. Pour pousser le projet plus loin, il aurait pu être intéressant d'affiner encore le maillage, d'ajouter des éléments comme des bulles d'air par exemple. Cependant, les calculs prenant beaucoup de temps, nous nous sommes arrêtés à la version en pièce jointe. Afin d'avoir une autre approche sur notre projet, il aurait été intéressant de créer de nombreux scénarios différents, mais les temps de calcul énormes nous ont limité.