
MODÉLISATION DU TRIMARAN ZEPHYR

-

CODE DE CALCUL CERF

Interactions Fluide-Structure
Seatech 3A - MOCA

2024 - 2025

Auteurs :
Fabien MATHE & Yohan STABLO

Professeur encadrant :
M. Frédéric GOLAY

Contents

Introduction	2
I Initialisation	3
I.1 Modèle physique	3
I.2 Conditions aux limites	3
I.3 Conditions initiales	3
I.4 Maillage	4
I.5 Configuration	5
II Résultats et discussions	6
II.1 Résultats	6
II.2 Discussions	6
Conclusion	8

Introduction

Ce rapport a pour vocation de rendre compte du travail effectué par notre équipe dans le cadre de la prise en main du code de calcul **CERF** (Computation, ERosion & Flows), développé par M. Frédéric GOLAY [[visit his website here](#)].

Pour ce faire, nous avons décidé de simuler les interactions autour du trimaran ZÉPHYR réalisé par l'association HydroSharks [[website](#)] dans différents cas de figure :

- mise à l'eau depuis une rampe ;
- comportement sur une vague ;
- submersion après déferlement.

Nous exposerons donc les conditions et paramétrages de nos simulations avant d'en présenter les résultats afin de pouvoir les discuter.

I Initialisation

I.1 Modèle physique

Le code de calcul CERF résout les équations de Navier-Stokes par méthode des volumes finis. Nous l'utilisons pour résoudre un problème d'écoulement multiphasique isotherme par méthode VOF (Volume Of Fluid).

I.2 Conditions aux limites

Dans toutes les différentes configuration présentées dans la suite de ce rapport, nous utilisons des conditions de type *miroir*. Physiquement, cela représente une domaine fermé où les fluides ne peuvent s'échapper.

I.3 Conditions initiales

I.3.1 Cas du surf de vague

Pour ce qui est du comportement sur une vague, nous avons voulu représenter le bateau "surfant" une vague. Pour ce faire, le bateau est initialement dans le creux d'une vague de 4 m de haut :



Figure I.1: Initialisation du cas de *surf*

Pour ce faire, la vague est définie par une fonction utilisateur sous la forme d'une gaussienne. Sous la gaussienne, la densité est définie à 1000 kg m^{-3} et la pression corrélée à la hauteur d'eau :

```

1 CASE(2)
2 ! Gaussian wave input
3 h1 = 7.d0
4 hw = 1.d0
5 c0 = 1.d0
6 sigma = 1.5d0
7 vprim%v(1) = 1.0d0
8 vprim%v(5) = 1.0d5
9 IF(x .le. 50.d0) THEN
10    zb = hw + (h1 - hw) * exp(-(x - c0)**2.d0/(2.d0*sigma**2.d0))
11    )
12    IF(y .le. zb)THEN
13       vprim%v(1) = 1000.d0
14       vprim%v(5) = 1.d5 + 1000.d0*9.81*(zb - y)
15    ENDIF
ENDIF

```

Listing 1: Fonction utilisateur pour le cas de *surf*

I.3.2 Cas du déferlement

Dans le cas du déferlement de la vague sur le bateau, c'est toujours une vague gaussienne de 4 m haut qui est initialisée, mais cette fois le bateau est placé initialement plus loin, dans la zone de déferlement de la vague :



Figure I.2: Initialisation du cas de déferlement

La fonction utilisateur est la même, seul le fichier *.obj* du bateau est différent.

I.4 Maillage

I.4.1 Cas du surf de vague

Les éléments du maillage maître sont des carrés de taille 50 cm par 50 cm. Cependant, le niveau de raffinement maximal est 4, ce qui donne des mailles de taille 3.125 cm par 3.125 cm.

Lors de l'initialisation, l'interface *air-eau* est raffinée au niveau 3 et le bateau au niveau maximal :

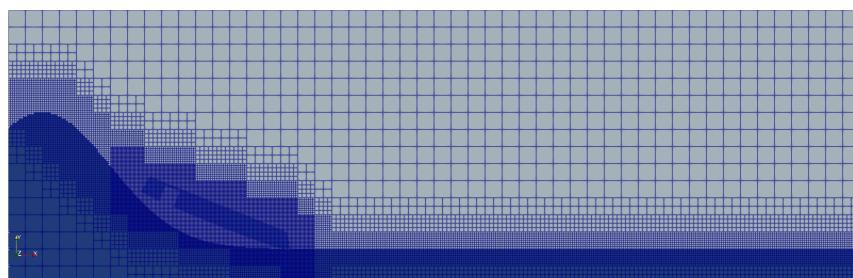


Figure I.3: Maillage initial du cas de surf

Tout ceci se fait grâce à la fonction de maillage utilisateur :

```

1 CASE(2)
2 ! Gaussian wave meshing
3 h1 = 5.d0
4 hw = 1.d0
5 nrb=0
6 IF(x.le.6.d0 .and. y.ge.hw .and. y.ge.(5.d0 - x) .and. y.le
7     .(7.5 - x) .and. y.le.(h1 + hw))nrb=3
8 IF(x.ge.6.d0 .and. y.ge.(hw - 0.5) .and. y.le.(hw + 0.5))nrb
9     =3
! Boat meshing
10 IF(x.ge.3.d0 .and. x.le.9.d0 .and. y.ge.(3.d0 - x*3.d0/9.d0)
11     .and. y.le.(5.d0 - x*3.d0/9.d0))nrb=4

```

Listing 2: Maillge utilisateur pour le cas de *surf*

I.4.2 Cas du déferlement

Le maillage maître est le même que le cas précédent, avec le même niveau de raffinement maximal, ce qui donne l'initialisation suivante :

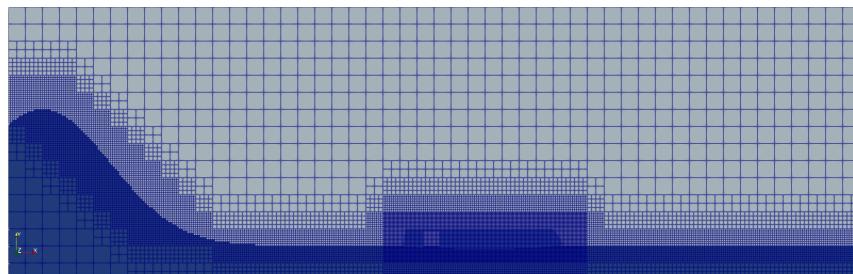


Figure I.4: Maillage initial du cas de déferlement

Avec la fonction de maillage utilisateur :

```
1 CASE(3)
2   ! Gaussian wave meshing
3   h1 = 7.d0
4   hw = 1.d0
5   nrb=0
6   IF(x.le.6.d0 .and. y.ge.hw .and. y.ge.(6.d0 - x*6.d0/4.d0) .
7     and. y.le.(10.d0 - x*10.d0/7) .and. y.le.(h1 + hw))nrb=3
8   IF(x.ge.6.d0 .and. y.ge.(hw - 0.5) .and. y.le.(hw + 0.5))nrb
9     =3
! Boat meshing
10  IF(x.ge.11.d0 .and. x.le.17.d0 .and. y.ge.0.5 .and. y.le.2.d0
11    )nrb=4
```

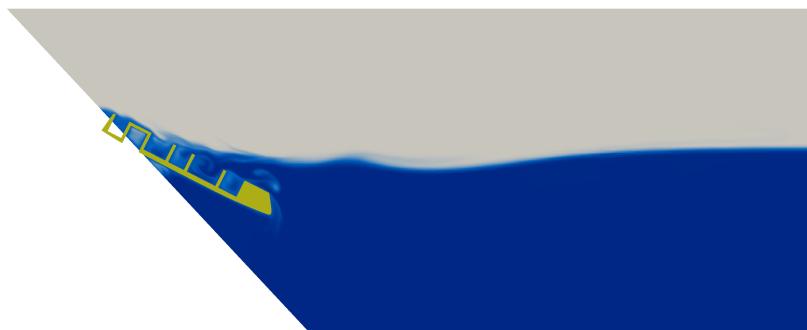
Listing 3: Maillage utilisateur pour le cas de déferlement

I.5 Configuration

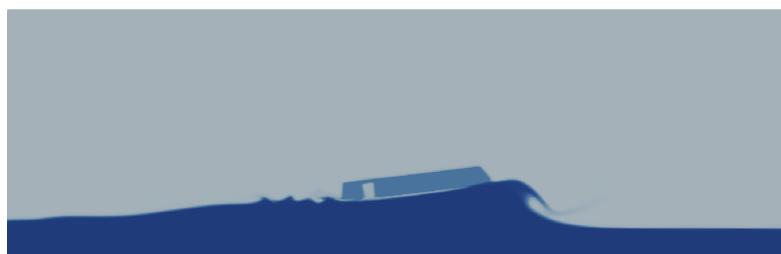
II Résultats et discussions

II.1 Résultats

Après avoir lancé les calculs de nos simulations nos résultats sont disponibles dans les vidéos présentent ici : [\[vidéos\]](#).



(a) Mise à l'eau, exemple de résultat



(b) Surf d'une vague, exemple de résultat



(c) Déferlement, exemple de résultat

Figure II.1: Exemple de résultats en cours de calcul pour les différents cas

II.2 Discussions

Plusieurs points peuvent être critiqués sur nos simulations.

Problème de maillage

La première critique à formuler concerne le maillage. Effectivement, pour capter correctement les différentes interfaces (*air-eau* et *solide-liquides*), il faudrait un maillage très fin, avec un niveau de raffinement extrême.

Cependant, si nous avions raffiné plus, le temps de calcul nécessaire à la résolution de nos cas aurait explosé, déjà que pour simuler 8 sec (déferlement) il nous a fallu plus de 1 h.

Malheureusement, ce manque de finesse dans le maillage introduit un nouveau problème : les objets, qui sont des liquides figés, diffusent, c-à-d que le bateau "fond". En d'autres termes, le maillage n'est pas assez fin et le temps de remaillage déjà trop long pour capturer l'interface entre solide et liquides et des bouts du bateau sont alors détachés de l'objet solide, donnant cette impression qu'il est en train de fondre :

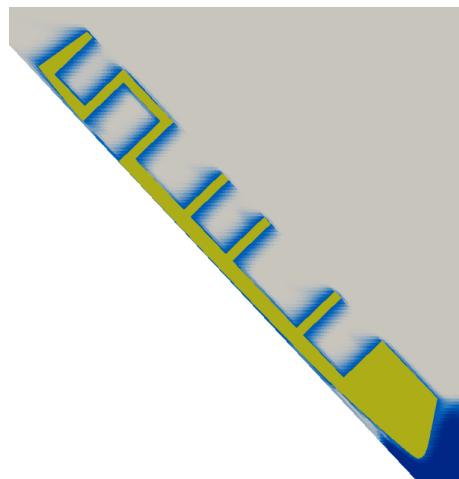


Figure II.2: Illustration du problème de diffusivité des objets solides

Conclusion