

# Procédure de time-splitting pour la résolution de la fraction volumique dans le cas d'écoulements instationnaires

**Y. ROBERT, A. LEROYER, J. WACKERS,  
M. VISONNEAU et P. QUEUTEY**

Laboratoire de recherche en Hydrodynamique, Énergétique et Environnement  
Atmosphérique (LHEEA), UMR CNRS 6598, École Centrale de Nantes.  
yoann.robert@ec-nantes.fr

## Résumé :

*Les simulations numériques d'écoulements multifluides utilisant des schémas de capture d'interface souffrent de la contrainte sur le pas de temps imposée par la condition Courant-Friedrichs-Lewy (CFL). Nous proposons ici une méthode pour accélérer les calculs, appelée procédure time-splitting. L'équation de convection de la fraction volumique est résolue sur des pas de temps intermédiaires où la condition CFL est vérifiée. Elle permet ainsi l'utilisation d'un pas de temps global plus grand. Les développements ont été validés sur différents écoulements 2D. Nous présentons un cas test modélisant l'écoulement provoqué par une rupture de barrage suivie de l'impact sur un obstacle. Dans cet exemple, un facteur de réduction du temps de calcul allant jusqu'à 3 a été obtenu.*

## Abstract :

*Numerical simulations of multifluid flows using interface capturing schemes suffer from the constraint on the time step imposed by the Courant-Friedrichs-Lewy (CFL) condition. Here we show a method to accelerate the computations, so called time-splitting procedure. The convection equation of the volumic fraction is solved on intermediate time steps where the CFL condition is verified. It allows the use of larger global time steps. The developments have been validated on different 2D flows. We present a test case modelling the flow caused by a dam break followed by the impact on a obstacle. In this example, a reduction factor of the computation time going up to 3 is achieved.*

**Mots clefs : hydrodynamique ; capture d'interface ; time-splitting**

## 1 Introduction

Les simulations numériques d'écoulements multifluides à capture d'interface sont fortement contraintes par la condition de Courant-Friedrichs-Lewy (CFL). Même avec des codes implicites, le nombre de Courant  $Co$  est limité à des valeurs inférieures à l'unité afin de tirer profit des propriétés compressives des schémas et restreindre la diffusion de l'interface [4, 6]. Par conséquent, les pas de temps utilisables deviennent rapidement rédhibitoires étant donné qu'ils sont reliés au maillage par la définition de  $Co$  et que, si celui-ci s'affine, alors le pas de temps doit être diminué en conséquence. Ceci est d'autant plus

vrai quand la surface libre subit des déformations importantes et violentes, comme dans le cas d'un coup d'aviron dans l'eau [2, 5] ou d'un prisme impactant une surface libre [1, 7]. L'équation sujette à cette condition est l'équation de convection pour la fraction volumique  $c$ .

L'objectif des travaux est d'élaborer une stratégie permettant d'accélérer les simulations numériques d'écoulements multifluides instationnaires à surface libre tout en respectant formellement la condition CFL. La procédure présentée ici est une extension de celle déjà développée pour des écoulements tendant vers un état stationnaire [3]. En bref, la méthode autorise l'utilisation d'un pas de temps global plus grand à chaque itération temporelle et la résolution de l'équation de convection de  $c$  est réalisée en plusieurs étapes successives et sur des pas de temps plus petits. Ce compromis permet de diminuer le temps total de calcul, a priori sans perte de précision. Cette technique devrait être d'autant plus efficace pour les calculs massivement parallèles et en particulier pour les maillages fins avec raffinement automatique [7]. Dans ce cas, on réalise une forme indirecte de parallélisation de la résolution en temps : pour appliquer notre stratégie en utilisant un pas de temps plus grand, il faut se donner une marge plus grande sur le nombre de cellules avoisinant la surface libre et donc utiliser un nombre plus important de processeurs<sup>1</sup>. L'idée sous-jacente consiste à déplacer un problème de ressources en temps<sup>2</sup> vers une solution en ressources matérielles, ces dernières devenant de plus en plus accessibles au fil des ans.

Dans ce résumé, quelques explications sur les méthodes seront données dans la section 2 et des premiers résultats montrant les bénéfices seront exposés. Pour finir, la section 4 tirera les conclusions et dressera les perspectives. Dans la suite du document, on emploiera l'expression « procédure *time-splitting* » pour désigner la méthode décrite.

## 2 Méthode

### 2.1 Principe

La procédure *time-splitting* consiste dans une première étape à découper chaque pas de temps original en  $N$  intervalles successifs (cf. figure 1) et à résoudre  $c$  sur ceux-ci. À chacun de ces découpages temporels est donc associée une fraction du pas de temps original, ajustée de manière à obtenir un nombre de Courant respectant la condition CFL. De cette façon, le pas de temps global peut être augmenté et le temps CPU total en sera diminué. Il est alors a priori possible de faire tendre le pas de temps global vers une valeur critique traduisant la limite à ne pas dépasser pour capter la physique du problème, indépendamment de la finesse du maillage. Le surcoût venant des résolutions supplémentaires de la fraction volumique reste assez faible car cette résolution n'est pas la plus gourmande en terme de temps CPU dans les codes RANSE.

### 2.2 Équations

L'équation à résoudre pour les méthodes à capture d'interface utilisant une approche ALE est l'équation de convection pour la fraction volumique  $c$  :

$$\frac{\delta}{\delta t} \int_{\mathcal{V}} c \, d\mathcal{V} + \oint_{\mathcal{S}} c (\vec{U} - \vec{U}_d) \cdot \vec{n} \, d\mathcal{S} = 0, \quad (1)$$

1. Cela doit être fait pour s'assurer que la surface libre reste dans la zone de maillage raffinée, en attendant la prochaine étape de raffinement. En effet, le raffinement automatique de maillage entraîne un surcoût en temps de calcul et n'est pas effectué à toutes les itérations temporelles.

2. résolution séquentielle, par nature, dans notre code de calcul

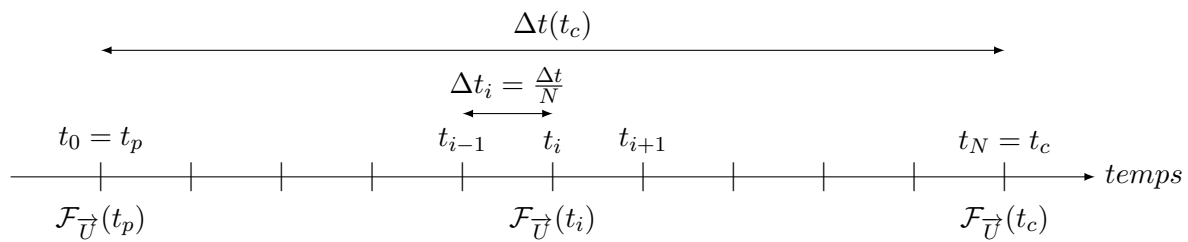


Figure 1 – Découpage temporel associé à l'équation (2).

où  $\mathcal{V}$  est le volume de contrôle limité par une surface fermée  $\mathcal{S}$  se déplaçant à une vitesse  $\vec{U}_d$  de vecteur normal  $\vec{n}$  dirigé vers l'extérieur et  $\frac{\delta}{\delta t}$  représente la dérivée temporelle en suivant le champ de déplacement  $\vec{U}_d$ .

La discrétisation en volumes finis avec un schéma Euler implicite au premier ordre en temps conduit à l'équation discrétisée suivante :

$$\frac{c(t_c) V(t_c) - c(t_p) V(t_p)}{\Delta t} + \sum_{\text{faces } \mathcal{S}_f} c_f(t_c) \left( \mathcal{F}_{\vec{U}}(t_c) - \mathcal{F}_{\vec{U}_d}(t_c) \right) = 0, \quad (2)$$

où  $t_p$  et  $t_c$  désignent respectivement les instants précédent et courant,  $\Delta t = t_c - t_p$  est le pas de temps courant et les volumes des cellules sont désignés par la lettre  $V$ . La reconstruction de la fraction volumique au centre de la face est notée  $c_f$ . Les vecteurs  $\mathcal{F}_{\vec{U}}$  et  $\mathcal{F}_{\vec{U}_d}$  sont respectivement le flux de vitesse et le flux de vitesse de déplacement du maillage à travers la face  $\mathcal{S}$ . Ces trois grandeurs sont exprimées à l'instant courant  $t_c$ . La discrétisation décrite ici, au premier ordre en temps, n'est pas apte à capturer précisément les phénomènes instationnaires (car trop diffusive). Cependant, pour des calculs tendant vers un état stationnaire, une résolution précise en temps n'est pas recherchée donc ce schéma convient bien. En revanche, pour les calculs instationnaires, il est primordial d'avoir une précision suffisante. Un schéma Euler implicite du second ordre en temps est donc utilisé :

$$e_c c(t_c) V(t_c) + e_p c(t_p) V(t_p) + e_q c(t_q) V(t_q) + \sum_{\text{faces } \mathcal{S}_f} c_f(t_c) \left( \mathcal{F}_{\vec{U}}(t_c) - \mathcal{F}_{\vec{U}_d}(t_c) \right) = 0, \quad (3)$$

où  $e_c$ ,  $e_p$  et  $e_q$  sont les coefficients du schéma sus-cité pour la dérivée temporelle et  $t_q$  est l'instant précédent  $t_p$ . L'utilisation de celui-ci pose de nouvelles difficultés, en particulier pour reconstruire  $c_f$  sur le pas de temps global. Comme mentionné précédemment, la résolution de (3) est réalisée par une succession de découpages temporels sur des intervalles  $\Delta t_i = \frac{\Delta t}{N}$  (cf. figure 1). Les flux sont interpolés entre leurs valeurs aux instants  $t_p$  et  $t_c$ . Le schéma de Crank-Nicolson est utilisé pour avoir une précision à l'ordre 2 en temps sur les pas de temps intermédiaires. Une méthode pour reconstruire  $c_f$  à la fin de cette résolution fractionnée est mise en œuvre. En effet, cette variable est requise pour calculer, par exemple, le terme  $\rho \mathcal{F}$  de l'équation de quantité de mouvement.

### 3 Résultats

La procédure *time-splitting* a été implémentée dans le code *ISIS-CFD*, développé par l'équipe DSPM (Dynamique des Systèmes Propulsifs Marins) du LHEEA, puis validés sur plusieurs cas-tests en 2D, pour des problèmes tendant vers des états stationnaires puis avec des écoulements purement instationnaires : simple advection d'un carré, foil submergé (Duncan), rupture de barrage avec obstacle ou encore

propagation d'une houle. Les solutions sont très proches (voire identiques) de celles obtenues sans utiliser la procédure de découpage temporel pour la résolution de  $c$ . Dans ces premiers essais, la baisse du temps CPU escomptée se confirme : un facteur de diminution allant jusqu'à 4 a été atteint. Nous nous limiterons, dans ce résumé, à la présentation des résultats pour le cas test *rupture de barrage avec obstacle*.

### 3.1 Rupture de barrage avec obstacle

Un des exemples les plus intéressants utilisé pour la validation d'écoulements à surface libre est la *rupture de barrage avec obstacle* [6]. À l'état initial (cf. figure 2), une colonne d'eau rectangulaire (146 cm par 292 cm) est située dans le coin inférieur gauche d'un domaine carré (de côté 584 cm) dont les bords sont des parois solides sauf le bord supérieur qui est à l'air libre. La gravité entraîne la chute de la colonne d'eau et celle-ci vient heurter un obstacle rectangulaire (24 cm par 48 cm).

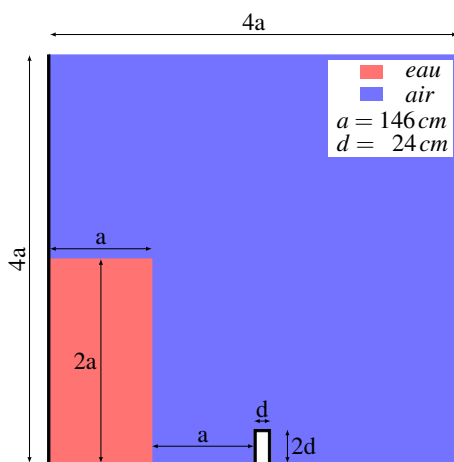


Figure 2 – Schéma de la configuration initiale du cas test *rupture de barrage avec obstacle*.

Pour ces simulations, une condition de glissement est appliquée à toutes les parois solides. Comme le cas est bidimensionnel, une condition de type miroir est utilisée pour les faces avant et arrière du domaine. Une condition de type Dirichlet est appliquée sur la face supérieure : la pression hydrostatique est imposée.

Deux types de maillage ont été utilisés. Dans la première configuration, le maillage est fixe au cours du calcul et assez fin. Dans la moitié inférieure du domaine, on a 192 cellules dans la direction horizontale et 96 dans la direction verticale. Le maillage devient très grossier au fur et à mesure que l'altitude augmente pour arriver à des dimensions de mailles égales à celles obtenues par découpage du domaine en 24 par 24. Dans la seconde configuration, on emploie le raffinement automatique de maillage, appelé AGR (Automatic Grid Refinement). Le maillage initial est assez grossier avec 58 cellules dans les deux directions, excepté dans une zone plus raffinée autour de l'obstacle. Les cellules autour de la surface libre sont raffinées au cours du calcul avec une taille cible de 1,25 mm (distance normale à la surface libre), sur un certain nombre de couches de cellules  $N_1$  (dépendant du cas souhaité) et avec une fréquence d'un raffinement tous les 3 pas de temps.

Les calculs ont été faits sur le même ordinateur et sur un seul processeur. Des cas utilisant la procédure *time-splitting* ou non ont été réalisés. Des nombres  $N_s$  de découpage temporels (appelés *splits*)

différents ont été testés. Pour simplifier, si  $N_s = 0$ , alors cela veut dire que la procédure *time-splitting* n'est pas utilisée. Augmenter  $N_s$  devrait faire diminuer le temps de calcul  $t_{CPU}$  mais il faut s'assurer que les résultats ne se dégradent pas, par exemple pour la composante horizontale de l'effort maximal sur l'obstacle  $F_{x_{max}}$ . Ne possédant pas de valeur expérimentale ou numérique de référence, on prendra arbitrairement la valeur maximale obtenue (cas 4 du tableau 1) comme référence. En utilisant la procédure *time-splitting*, il faut augmenter  $N_l$  pour avoir une marge de sécurité suffisante et le nombre de cellules moyen<sup>3</sup>  $N_{c, moy}$  sera donc plus important. On pourra également observer le coût  $C_{tot}$  pour chaque calcul, désignant le temps CPU total moyen par cellule.

Le pas de temps global est adapté à chaque itération temporelle de manière à avoir un nombre de Courant  $Co$  inférieur à une valeur limite. Pour les calculs n'utilisant pas la procédure *time-splitting*, cette limite vaut 0,3. Pour les autres, elle est fixée à  $0,3 N_s$ .

Les résultats des 6 cas sont synthétisés dans le tableau 1. L'évolution de l'écoulement simulé entre l'instant initial et  $t = 0,6 s$  est représenté en figure 3. Les images sont issues du cas 5. On retrouve bien les étapes décrites dans [6]. En quelques mots, on peut dire que l'on retrouve tout d'abord une fine couche issue de la colonne d'eau venant impacter l'obstacle. Une sorte de langue d'eau se forme au dessus de l'obstacle, puis se dirige vers la droite et vient s'écraser sur la paroi opposée du domaine, emprisonnant de l'air en dessous d'elle. Une langue secondaire se forme alors en haut à droite de l'obstacle, piégeant à son tour une zone d'air.

Les trois premiers cas utilisent la configuration avec maillage fixe. On peut principalement noter que l'utilisation de la procédure *time-splitting* permet de diminuer le temps de calcul. Un facteur d'environ 3 a été obtenu dans le cas où  $N_s$  vaut 10. Comme le nombre de mailles ne varie pas, on retrouve ce même facteur pour le coût  $C_{tot}$ . En revanche, on remarque tout de même une baisse sur la valeur de l'effort  $F_{x_{max}}$  (N). Il est possible que le pas de temps global ait été choisi trop grand par rapport à la physique de l'écoulement. L'utilisation d'un pas de temps trop large entraîne également un emprisonnement d'air entre l'eau et l'obstacle au moment de l'impact, ce qui diminue de ce fait l'effort.

Pour les autres cas, le raffinement automatique de maillage a été utilisé. Les gains en temps de calcul sont nettement plus faibles. Cela s'explique en particulier par un nombre moyen de cellules plus grands avec facteur allant jusqu'à 1,8 pour le cas 6. Cependant, il est intéressant de constater que, pour ce même cas, le coût est deux fois inférieur au cas 4 n'utilisant pas la procédure *time-splitting*. Compte tenu de la taille cible spécifiée pour le raffinement environ deux fois inférieur à la dimension des cellules du maillage fixe, la précision sur les valeurs de l'effort maximal est augmentée. Il faut aussi noter que, bien que le nombre de cellules moyen est plus faible que pour les cas à maillage fixe, les temps de calculs CPU sont plus importants. Cela est dû au coût de la procédure de raffinement automatique. Cela ne remet pas en cause l'intérêt d'une telle procédure. Étant donné la faible valeur de taille cible pour les cellules à générer et la loi d'adaptation des pas de temps en fonction d'un nombre de Courant limite, les pas de temps deviennent très petits.

---

3. Le nombre de cellules varie au cours du calcul quand le raffinement automatique de maillage est utilisé.

| Cas | $N_s$ | AGR | $N_l$ | $t_{CPU}$ (s) | ratio $t_{CPU}$ | $N_{c, moy}$ | ratio $N_{c, moy}$ | $C_{tot}(ms/pt)$ | ratio $C_{tot}$ | $F_{x_{max}}$ (N) | Erreur $F_{x_{max}}$ (%) |
|-----|-------|-----|-------|---------------|-----------------|--------------|--------------------|------------------|-----------------|-------------------|--------------------------|
| 1   | 0     | ×   | ×     | 1905          | -               | 20210        | -                  | 94.3             | -               | 183               | 0.5                      |
| 2   | 5     | ×   | ×     | 824           | 2.31            | 20210        | 1                  | 40.8             | 2.31            | 179               | 2.7                      |
| 3   | 10    | ×   | ×     | 646           | 2.95            | 20210        | 1                  | 32.0             | 2.95            | 175               | 4.9                      |
| 4   | 0     | ✓   | 1     | 3514          | -               | 5774         | -                  | 540.9            | -               | 184               | -                        |
| 5   | 10    | ✓   | 5     | 2545          | 1.38            | 8039         | 1.39               | 280.1            | 1.93            | 178               | 3.3                      |
| 6   | 10    | ✓   | 10    | 3285          | 1.07            | 10590        | 1.83               | 263.9            | 2.05            | 179               | 2.7                      |

Tableau 1 – Résumé des résultats obtenus pour un temps de simulation physique de 0,2 s.

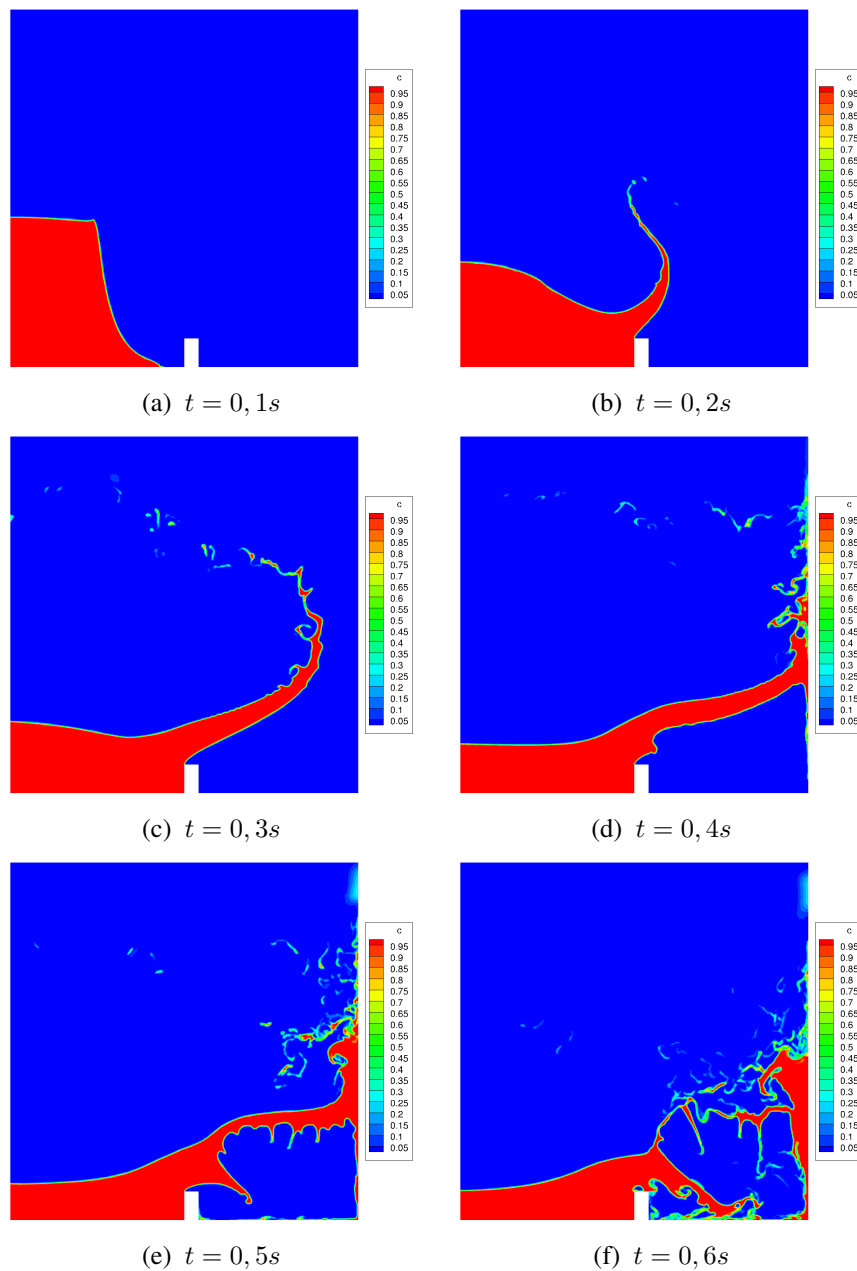


Figure 3 – Évolution de l'écoulement simulé.

## 4 Conclusion et perspectives

Les premiers résultats sont encourageants et confirment les espoirs placés dans l'élaboration de la procédure *time-splitting*. Sur des maillages fixes, le temps CPU pour la simulation du cas test *rupture de barrage avec obstacle* a été réduit d'un facteur 3. Dans le cas de calculs sur un seul processeur, le temps CPU n'est que très peu réduit quand on utilise le raffinement automatique de maillage. Ceci s'explique par la nécessité de prévoir une marge de sécurité conséquente en nombre de mailles autour de la surface libre. Cependant, le coût d'un calcul, en terme de temps CPU total moyen par cellule, est réduit de moitié. Il apparaît alors que la procédure *time-splitting* devrait être très efficace sur des calculs massivement parallèles.

## Références

- [1] A. Hay, A. Leroyer, and M. Visonneau. H-adaptive navier–stokes simulations of free-surface flows around moving bodies. *Journal of Marine Science and Technology*, 11(1) :1–18, 2006.
- [2] A. Leroyer, S. Barré, J.-M. Kobus, and M. Visonneau. Influence of free surface, unsteadiness and viscous effects on oar blade hydrodynamic loads. *Journal of Sports Sciences*, 28(12) :1287–1298, 2010.
- [3] A. Leroyer, J. Wackers, P. Queutey, and E. Guilmineau. Numerical strategies to speed up CFD computations with free surface - application to the dynamic equilibrium of hulls. *Ocean Engineering*, 38(17–18) :2070–2076, 2011.
- [4] F. Moukalled and M. Darwish. Transient schemes for capturing interfaces of free–surface flows. *Numerical Heat Transfer, Part B : Fundamentals*, 61(3) :171–203, 2012.
- [5] Y. Robert, A. Leroyer, S. Barré, F. Rongère, P. Queutey, and M. Visonneau. Fluid mechanics in rowing : the case of the flow around the blades. In *The 2014 conference of the International Sports Engineering Association*, 2014.
- [6] O. Ubbink. *Numerical prediction of two fluid systems with sharp interfaces*. PhD thesis, Imperial College, University of London, 1997.
- [7] J. Wackers, G. Deng, A. Leroyer, P. Queutey, and M. Visonneau. Adaptive grid refinement for hydrodynamic flows. *Computers & Fluids*, 55 :85–100, 2012.