Study of interface capturing methods for two-phase flows

Nabil Djati

N°d'ordre NNT : 2017LYSEI052

THESE de DOCTORAT DE L'UNIVERSITE DE LYON
opérée au sein de
l'INSA LYON

Ecole Doctorale ED162
(Mécanique, Energétique, Génie Civil, Acoustique)

Spécialité/ discipline de doctorat : Thermique et Energétique

Soutenue publiquement le 22/06/2017, par :
Nabil DJATI

Study of interface capturing methods for two-phase flows

Devant le jury composé de :

SPELT, Peter	Professeur (EC Lyon)	Président
CHENIER, Eric	Maître de Conférences (U-PEM)	Rapporteur
JOSSERAND, Christophe	Directeur de Recherche CNRS (Ecole Polytechnique)	Rapporteur
DARU, Virginie	Maître de Conférences (ENSAM)	Examinatrice
XIN, Shihe	Professeur (INSA-Lyon)	Directeur de thèse
KNIKKER, Ronnie	Maître de Conférences (INSA-Lyon)	Co-directeur de thèse

Département FEDORA – INSA Lyon - Ecoles Doctorales – Quinquennal 2016-2020

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	**CHIMIE DE LYON** **http://www.edchimie-lyon.fr** Sec : Renée EL MELHEM Bat Blaise Pascal 3e etage secretariat@edchimie-lyon.fr Insa : R. GOURDON	M. Stéphane DANIELE Institut de Recherches sur la Catalyse et l'Environnement de Lyon IRCELYON-UMR 5256 Équipe CDFA 2 avenue Albert Einstein 69626 Villeurbanne cedex directeur@edchimie-lyon.fr
E.E.A.	**ELECTRONIQUE, ELECTROTECHNIQUE, AUTOMATIQUE** **http://edeea.ec-lyon.fr** Sec : M.C. HAVGOUDOUKIAN Ecole-Doctorale.eea@ec-lyon.fr	M. Gérard SCORLETTI Ecole Centrale de Lyon 36 avenue Guy de Collongue 69134 ECULLY Tél : 04.72.18 60.97 Fax : 04 78 43 37 17 Gerard.scorletti@ec-lyon.fr
E2M2	**EVOLUTION, ECOSYSTEME, MICROBIOLOGIE, MODELISATION** **http://e2m2.universite-lyon.fr** Sec : Sylvie ROBERJOT Bât Atrium - UCB Lyon 1 04.72.44.83.62 Insa : H. CHARLES secretariat.e2m2@univ-lyon1.fr	M. Fabrice CORDEY CNRS UMR 5276 Lab. de géologie de Lyon Université Claude Bernard Lyon 1 Bât Géode 2 rue Raphaël Dubois 69622 VILLEURBANNE Cédex Tél : 06.07.53.89.13 cordey@univ-lyon1.fr
EDISS	**INTERDISCIPLINAIRE SCIENCES-SANTE** **http://www.ediss-lyon.fr** Sec : Sylvie ROBERJOT Bât Atrium - UCB Lyon 1 04.72.44.83.62 Insa : M. LAGARDE secretariat.ediss@univ-lyon1.fr	Mme Emmanuelle CANET-SOULAS INSERM U1060, CarMeN lab, Univ. Lyon 1 Bâtiment IMBL 11 avenue Jean Capelle INSA de Lyon 696621 Villeurbanne Tél : 04.72.68.49.09 Fax :04 72 68 49 16 Emmanuelle.canet@univ-lyon1.fr
INFOMATHS	**INFORMATIQUE ET MATHEMATIQUES** **http://infomaths.univ-lyon1.fr** Sec :Renée EL MELHEM Bat Blaise Pascal, 3e étage Tél : 04.72. 43. 80. 46 Fax : 04.72.43.16.87 infomaths@univ-lyon1.fr	M. Luca ZAMBONI Bâtiment Braconnier 43 Boulevard du 11 novembre 1918 69622 VILLEURBANNE Cedex Tél :04 26 23 45 52 zamboni@maths.univ-lyon1.fr
Matériaux	**MATERIAUX DE LYON** **http://ed34.universite-lyon.fr** Sec : Marion COMBE Tél:04-72-43-71-70 –Fax : 87.12 Bat. Direction ed.materiaux@insa-lyon.fr	M. Jean-Yves BUFFIERE INSA de Lyon MATEIS Bâtiment Saint Exupéry 7 avenue Jean Capelle 69621 VILLEURBANNE Cedex Tél : 04.72.43 71.70 Fax 04 72 43 85 28 Ed.materiaux@insa-lyon.fr
MEGA	**MECANIQUE, ENERGETIQUE, GENIE CIVIL, ACOUSTIQUE** **http://mega.universite-lyon.fr** Sec : Marion COMBE Tél:04-72-43-71-70 –Fax : 87.12 Bat. Direction mega@insa-lyon.fr	M. Philippe BOISSE INSA de Lyon Laboratoire LAMCOS Bâtiment Jacquard 25 bis avenue Jean Capelle 69621 VILLEURBANNE Cedex Tél : 04.72 .43.71.70 Fax : 04 72 43 72 37 Philippe.boisse@insa-lyon.fr
ScSo	**ScSo*** **http://recherche.univ-lyon2.fr/scso/** Sec : Viviane POLSINELLI Brigitte DUBOIS Insa : J.Y. TOUSSAINT Tél : 04 78 69 72 76 viviane.polsinelli@univ-lyon2.fr	M. Christian MONTES Université Lyon 2 86 rue Pasteur 69365 LYON Cedex 07 Christian.montes@univ-lyon2.fr

*ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

Abstract

This thesis is devoted to the development and comparison of interface methods for incompressible two-phase flows. It focuses on the selection of robust interface capturing methods, then on the manner of their coupling with the Navier-stokes solver. The level-set method is first investigated, in particular the influence of the advection scheme and the reinitialization step on the accuracy of the interface capturing. It is shown that the volume constraint method for reinitialization is robust and accurate in combination with the conservative fifth-order WENO schemes for the advection. It is found that interface errors increase drastically when the CFL number is very small. As a remedy, reinitializing the level-set field less often reduces the amount of numerical diffusion and non-physical interface displacement. Mass conservation is, however, not guaranteed with the level-set methods. The volume-of-fluid (VOF) method is then investigated, which naturally conserves the mass of the reference fluid. A geometrical consistent and conservative scheme is adopted, then an alternative technique more easily extended to 3D. It is found that both methods give very similar results. The moment-of-fluid (MOF) method, which reconstructs the interface using the reference fluid centroid, is found to be more accurate than the VOF methods. Different coupled level-set and VOF methods are then investigated, namely: CLSVOF, MCLS, VOSET and CLSMOF. It is observed that the level-set method tends to thicken thin filaments, whereas the VOF and coupled methods break up thin structures in small fluid particles. Finally, we coupled the level-set and volume-of-fluid methods with the incompressible Navier-Stokes solver. We compared different manners (sharp and smoothed) of treating the interface jump conditions. It is shown that the VOF methods are more robust, and provide excellent results for almost all the performed simulations. Two level-set methods are also identified that give very good results, comparable to those obtained with the VOF methods.

Keywords: Two-phase incompressible flow simulation, interface capturing, level-set method, volume-of-fluid method, moment-of-fluid method, coupled level-set and volume-of-fluid, ghost fluid method, continuous surface force.

Résumé

Cette thèse est consacrée au développement et à la comparaison des méthodes de suivi d'interface pour les écoulements diphasiques incompressibles. Elle s'intéresse à la sélection de méthodes robustes de suivi d'interface, puis à leur couplage avec le solveur des équations de Navier-Stokes. La méthode level-set est en premier lieu étudiée, en particulier l'influence du schéma d'advection et de l'étape de réinitialisation sur la qualité des résultats du suivi d'interface. Il a été montré que la méthode de réinitialisation avec contrainte de volume est robuste et précise en combinaison avec des schémas conservatifs WENO d'ordre 5 pour l'advection. Il a été constaté que les erreurs du suivi d'interface augmentent de manière abrupte lorsque la condition CFL est trop petite. Comme remède, la réinitialisation du champ level-set effectuée moins souvent réduit la diffusion numérique et le déplacement non-physique de l'interface. La conservation de la masse n'est pas assurée avec les méthodes level-set. Les méthodes VOF (volume-of-fluid) qui conservent naturellement la masse du fluide de référence sont alors étudiées. Une résolution géométrique avec un schéma consistent et conservatif est alors adoptée, ainsi qu'une autre technique alternative plus aisément extensible en 3D. Il a été trouvé que ces deux dernières méthodes donnent des résultats très proches. La méthode MOF (moment-of-fluid), qui reconstruit l'interface en utilisant le centre de masse du fluide de référence, est plus précise que les méthodes VOF. Différentes méthodes couplées entre level-set et VOF sont alors étudiées, notamment: CLSVOF, MCLS, VOSET et CLSMOF. Il a été observé que la méthode level-set tend à épaissir les filaments minces, tandis que VOF et les méthodes couplées les fragmentent en petites particules. Finalement, on a couplé les méthodes level-set et VOF avec le solveur incompressible des équations de Navier-Stokes. On a comparé différentes manières de prise en compte des conditions de saut à l'interface (lisse et raide). Il a été montré que les méthodes VOF sont plus robustes, et donnent d'excellents résultats pour quasiment toutes les simulations. Deux méthodes level-set donnant de très bons résultats, comparables à ceux de VOF, sont aussi identifiées.

Mots clés: Simulation d'écoulement diphasique incompressible, suivi d'interface, méthode level-set, méthode VOF, méthode MOF, méthodes couplées level-set et VOF, ghost fluid method, continuous surface force.

Résumé étendu

Introduction

Les écoulements diphasiques concernent les écoulements avec deux fluides non-miscibles. Ils font intervenir différentes phases et ils sont souvent caractérisés par des rapports de densité et de viscosité élevés. Ces écoulements sont omniprésents dans la nature et dans les procédés industriels. Les vagues marines et les rivières sont des exemples concrets qui mettent en jeu l'écoulement de l'eau en interaction avec une surface libre. Dans les procédés industriels, ces écoulements sont beaucoup plus complexes. Les écoulements à bulles sont souvent rencontrés dans les échangeurs de chaleur. Les jets de combustible dans les moteurs à combustion sont atomisés afin de créer un mélange parfait d'air et de combustible. Cela représente une liste non-exhaustive des écoulements diphasiques. Des phénomènes de changement de phase peuvent avoir lieu dans ces écoulements sous l'effet d'un échange d'énergie. L'étude expérimentale de ces phénomènes est généralement compliquée et donne seulement des corrélations valides dans certaines conditions. La simulation numérique apparait comme une bonne alternative - moins coûteuse - si les outils numériques sont testés, validés et issus de méthodes robustes.

La modélisation de ces écoulements nécessite l'utilisation d'une méthode fiable permettant de localiser avec précision l'interface (frontière physique) qui sépare les deux fluides. Différentes méthodes de suivi d'interface sont proposées dans la littérature. Elles permettent une approche à un fluide lorsqu'elles sont couplées avec le solveur des équations de Navier-Stokes où un seul système d'équations est résolu (les deux fluides sont considérés comme étant un seul fluide avec des propriétés physiques variables). La simulation numérique des écoulements diphasiques dépend donc fortement de la capacité de ces méthodes à localiser l'interface.

L'objectif à long terme de ce travail est l'étude numérique des écoulements diphasiques avec changement de phase qui ont déjà fait l'objet de nombreuses études expérimentales au laboratoire. Cette thèse se focalise sur l'étude et la comparaison des méthodes Eulériennes de suivi d'interface pour les écoulements diphasiques. Ce dernier est réalisé en deux étapes : la première vise à sélectionner des méthodes robustes et fiables capables de prédire avec précision l'évolution de l'interface dans un champ de vitesse imposé ; et la seconde examine et traite le couplage entre les méthodes de suivi d'interface et le solveur de Navier-Stokes.

Dans ce travail, nous nous intéressons aux méthodes Eulériennes de suivi d'interface, plus particulièrement level-set, volume-of-fluid (VOF) et les couplages level-set/VOF.

Chapitre 1 : La méthode level-set

La méthode level-set est une méthode Eulérienne de suivi d'interface et son transport est régit par une équation d'advection d'un scalaire. L'interface est définie par la courbe de niveau zéro

d'une fonction continue : cette fonction est positive dans un fluide, s'annule à l'interface et change de signe dans l'autre fluide. Cette condition est satisfaite par une infinité de fonctions, mais la fonction de distance algébrique a l'avantage de permettre de déterminer avec précision les propriétés géométriques de l'interface (normale à l'interface et courbure par exemple).

Dans ce chapitre, on décrit différents schémas numériques communément utilisés pour la discrétisation spatiale de l'équation de transport de level-set, en particulier les schémas ENO (*Essentially Non-Oscillatory*), WENO (*Weighted ENO*) et HOUC (*High-Order Upstream Central*). Quant à la discrétisation temporelle, elle est réalisée avec le schéma TVD Runge-Kutta d'ordre 3. Le but étant d'identifier les schémas offrant une meilleure stabilité numérique à la méthode et une flexibilité par rapport aux configurations auxquelles elle peut être appliquée. Ainsi, une comparaison entre ces schémas est réalisée sur des cas tests académiques où le champ de vitesse est imposé (cercle de Zalesak, vortex isolé ou serpentin, et vortex isolé avec un champ de vitesse dépendant du temps).

Lorsque le champ level-set est transporté par l'écoulement, la propriété de distance signée à l'interface est plus ou moins altérée suivant la régularité du champ de vitesse qui la transporte. Afin de préserver cette propriété, on résout une équation supplémentaire (Sussman et al., 1994) qui s'appelle l'équation de réinitialisation (ou encore de redistanciation). En théorie, cette équation supplémentaire doit être résolue sur tout le domaine ; mais en pratique, la résolution sur une distance de l'ordre de quelques mailles est suffisante pour rétablir la propriété de distance signée au voisinage de l'interface. La résolution de l'équation de réinitialisation est sensée préserver le niveau zéro. Malheureusement, cette étape à tendance à déplacer l'interface et ainsi génère une variation non-physique de la masse des deux fluides. Plusieurs améliorations ont été proposées dans la littérature. On s'est intéressé à deux méthodes : *sub-cell fix* (Russo and Smereka, 2000) et *volume constraint* (Sussman and Fatemi, 1999). Dans la première méthode, une modification de l'équation de réinitialisation sur les cellules par lesquelles l'interface passe est proposée. Dans la seconde, la conservation de masse du fluide de référence est forcée afin que le volume avant et après réinitialisation soit égal.

Les tests effectués avec et sans réinitialisation du champ level-set ont montré que la réinitialisation est à l'origine du déplacement de l'interface. Afin de limiter ce déplacement non-physique de l'interface, on a réinitialisé le champ level-set avec la méthode sub-cell fix et volume constraint en combinaison avec différents schémas de discrétisation spatiale de l'équation d'advection. La méthode level-set avec volume constraint s'est avérée beaucoup plus robuste et précise, notamment avec les schémas conservatifs WENO d'ordre 5 pour le transport.

Dans le cas du serpentin et des filaments minces, les méthodes level-set engendrent soit un épaississement de ces filaments (et ainsi un gain conséquent de masse) ou un raccourcissement de leur longueur (perte importante de masse).

A l'issue de ce premier chapitre, nous avons sélectionné une méthode level-set (LS) robuste qui consiste en une

- advection avec le schéma WENO d'ordre 5,
- réinitialisation avec volume constraint après chaque advection. Cette équation est résolue sur un temps fictif $\tau = 2\Delta x$ avec un pas $\Delta\tau = 0.5\Delta x$. Le schéma HJ-ENO d'ordre 2 est utilisé pour la discrétisation spatiale de cette équation.

Chapitre 2 : La méthode VOF (Volume-of-fluid)

La méthode VOF (Volume-of-fluid) est une méthode Eulérienne de suivi d'interface qui conserve la masse du fluide de référence. Elle est décrite par la fraction (C) du fluide de référence dans

chaque cellule, ainsi elle vaut 1 si une cellule est pleine et elle est nulle si la cellule est vide. Dans une cellule mixte qui contient les deux fluides, la fraction a une valeur comprise entre 0 et 1. Le transport de ces fractions est régit par une équation d'advection similaire à celle de level-set. Cependant, les méthodes de résolutions ne sont pas les mêmes. La résolution de l'équation d'advection de VOF est souvent réalisée par des méthodes géométriques.

La position de l'interface n'est pas connue avec précision, et dépend de l'algorithme utilisé pour sa reconstruction. La majorité de ces algorithmes de reconstruction de l'interface prédisent une interface discontinue sur la face entre deux cellules voisines. Dans le cadre de ce travail, on utilise la méthode ELVIRA (*Efficient Least-squares Vof Interface Reconstruction Algorithm*). Dans une cellule mixte, l'interface est reconstruite en se basant sur les informations dans un bloc de 3×3 cellules autour de la cellule en question.

On résout l'équation d'advection géométriquement, direction par direction, tout en inversant la direction par laquelle on commence après chaque pas de temps. En deux dimensions par exemple, on commence au premier pas de temps par la direction x puis y, et dans le second pas de temps par la direction y puis x et ainsi de suite. Cette inversion de directions (de commencement dans chaque deux pas de temps successifs) garantie un ordre 2 de la discrétisation en temps.

On a décrit et implémenté un algorithme consistant (qui permet de garder les fractions C entre 0 et 1) et conservatif (qui conserve la masse totale, $\sum C^n = \sum C^{n+1}$) similaire à celui de Scardovelli and Zaleski (2003) et Tryggvason et al. (2011). La complexité de sa mise en œuvre vient du fait que cette advection utilise un mapping Lagrangien implicite dans la première direction et un mapping Lagrangien explicite suivant l'autre direction en deux dimensions. Ses extensions actuelles à 3D ne permettent malheureusement pas de satisfaire les conditions de consistance et de conservation de masse. En revanche, l'algorithme alternatif développé par Weymouth and Yue (2010) (que l'on note VOF-WY) est consistant et conservatif, et est aisément extensible en 3D. La version 2D des deux algorithmes a été intégrée dans notre code.

Dyadechko and Shashkov (2005, 2006, 2008) ont proposé la méthode MOF (*moment-of-fluid*) qui permet de reconstruire l'interface seulement en se basant sur les informations connues sur la cellule en question, à savoir le centre de masse du fluide de référence et sa fraction. Il est à noter que la position des centres de masse doit être recalculée et mise à jour après chaque advection, étape qu'on réalise suivant une manière similaire à celle pour le transport des fractions.

On a présenté les algorithmes de chacune des méthodes, ainsi que ceux de quelques fonctions utiles pour l'élaboration de ces méthodes en 2D. La comparaison entre ces méthodes (VOF, VOF-WY et MOF) a montré qu'elles sont robustes et moins sensibles à la condition CFL que level-set. La méthode VOF (Weymouth and Yue, 2010) est identique à celle de Scardovelli and Zaleski (2003) dans le cas du cercle de Zalesak du moment que le champ de vitesse imposé ne fait subir aux cellules ni compression ni expansion. Dans le cas du serpentin et des filaments minces, à l'inverse de level-set qui les rend plus épais, les méthodes VOF ne gardent pas une interface continue mais causent la fragmentation de ces filaments. La méthode MOF au contraire, du fait qu'elle n'utilise que les informations de la cellule en question pour reconstruire l'interface, garde une structure continue des filaments. Cependant, on a remarqué quelques problèmes de stabilité numérique (détachement non-physique de petites particules fluides) de la méthode MOF lorsque le CFL imposé est trop petit. Ces problèmes pourront probablement être résolus dans une future implémentation plus soignée de MOF.

Chapitre 3 : Méthodes couplées level-set et VOF (volume-of-fluid)

Les avantages et inconvénients de chacune des méthodes level-set et VOF sont bien connus. Bourlioux (1995) a eu l'idée de coupler level-set avec VOF afin de rassembler dans une même

méthode les avantages de chacune, à savoir la conservation de masse de VOF et la bonne représentation de l'interface de level-set.

Sussman and Puckett (2000) a concrétisé ce couplage en publiant le premier article, suivi par un autre article de Son and Hur (2002). La méthode a été appelée CLSVOF (*Coupled level-set and volume-of-fluid*). Dans cette méthode, il y a interaction entre les champs de level-set et des fractions en plusieurs étapes. D'une part, l'advection de VOF nécessite la connaissance de la normale à l'interface qui est déterminée par le biais de level-set. D'autre part, le champ de level-set doit conserver la masse, et ainsi il est réinitialisé par level-set et VOF. Dans ce travail, on a distingué deux versions de CLSVOF : CLSVOF-Sussman (Sussman and Puckett, 2000) et CLSVOF-Son (Son and Hur, 2002). Deux étapes sont réalisées différemment : (1) l'advection de level-set et (2) la méthode de détermination de la normale à l'interface de VOF.

Quelques années plus tard, van der Pijl et al. (2005) ont proposé un couplage entre level-set et VOF qu'ils ont nommé MCLS (*mass conserving level-set*). Le champ de level-set est avancé et réinitialisé par les mêmes schémas numériques que level-set classique ($\phi^n \rightarrow \widetilde{\phi}^{n+1}$) mais avec un ordre moins élevé afin de garantir le caractère lisse de la solution. Le champ VOF (C^n) est déterminé à partir de level-set (ϕ^n) avant l'étape d'advection et avancé dans le temps ($C^n \rightarrow C^{n+1}$). Il est ensuite utilisé pour forcer la conservation de masse du champ level-set à la fin de l'itération (ϕ^{n+1}). Pour cette méthode, on a comparé différentes variantes avec des schémas de différents ordres, avec une réinitialisation volume constraint ou une réinitialisation standard.

Sun and Tao (2010) ont proposé le couplage appelé VOSET (*Coupled volume-of-fluid and level-set*). Dans ce dernier, il s'agit simplement de la détermination de la fonction de distance signée à partir de VOF via un processus géométrique itératif. Il est à noter que pour ce couplage, le champ level-set n'est pas transporté mais déduit à partir des fractions. La précision de VOSET dépend alors fortement de celle de l'algorithme utilisé pour la détermination du champ level-set. Pour cette méthode, on a considéré différentes variantes, chacune avec un nombre d'itérations géométriques donné.

Enfin, après l'apparition de la méthode MOF, elle a été couplée avec level-set par Jemison et al. (2013) dans une méthode appelée CLSMOF (*Coupled level-set and moment-of-fluid*). Cette dernière propose d'utiliser soit la reconstruction de MOF, soit celle de CLSVOF. Dans son principe, ce couplage est très semblable à celui de la méthode CLSVOF-Sussman.

Les fractions volumiques dans ces différents couplages sont advectées avec un schéma consistant et conservatif des fractions volumiques. Ceci nous a permis de ne pas utiliser des algorithmes de redistribution des fractions (algorithmes qui permettent de redistribuer les fractions afin que $0 \leq C \leq 1$ dans chaque cellule).

Nous avons comparé ces méthodes couplées sur les cas tests académiques des chapitres 1 et 2. On les a également comparées avec les méthodes VOF, MOF et LS non couplées. La figure 1 reprend la représentation de l'interface du serpentin à $t = 3$. Ces méthodes couplées ont un comportement similaire à celui de la méthode VOF, à savoir la fragmentation des filaments minces. En ce qui concerne la méthode CLSVOF, l'implémentation de Son and Hur (2002) a donné des résultats relativement plus précis que ceux de CLSVOF-Sussman. Les méthodes MOF et CLSMOF préservent la continuité de l'interface du serpentin mieux que les autres méthodes. A cet effet, la localisation de l'interface est plus précise avec les deux méthodes.

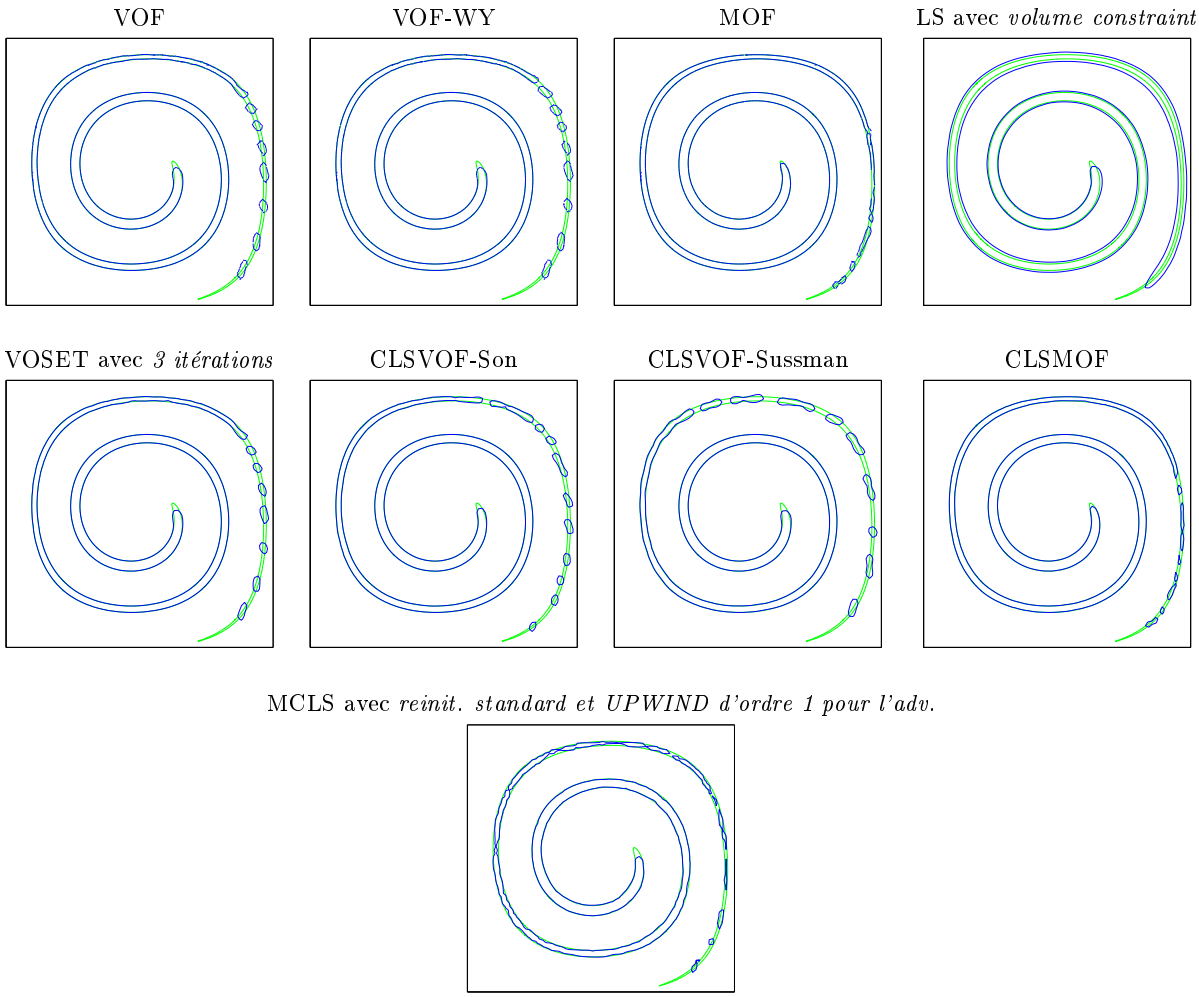

Figure 1: Interface (niveau zéro de level-set) du serpentin à $t = T/2 = 3$ (déformation maximale) sur un maillage de 100×100 avec CFL $= 0.5$. L'interface en vert représente la solution exacte.

Chapitre 4 : Méthodes numériques pour les écoulements diphasiques incompressibles

Dans les chapitres précédents, les méthodes de suivi d'interface développées sont appliquées à des cas tests académiques avec un champ de vitesse imposé (connu). Lorsque ces méthodes sont couplées avec un solveur des équations de Navier-Stokes, elles permettent une approche à un fluide (les deux fluides sont considérés comme un seul fluide avec des propriétés variables). Dans les simulations d'écoulements diphasiques, c'est par le biais du solveur des équations de Navier-Stokes que le champ de vitesse est déterminé et sert ensuite à transporter l'interface. Les propriétés du fluide sont déterminées à partir de la position de l'interface prédite par la méthode de suivi d'interface. Les conditions sur l'interface sont ainsi aisément appliquées du moment que la position de l'interface est connue.

Différentes manières de détermination des propriétés du fluide sont proposées dans la littérature, certaines considèrent une variation lisse au passage à l'interface de la propriété du fluide 1 et celle du fluide 2. D'autres proposent un saut brusque de ces propriétés. La manière de déterminer la courbure de l'interface aussi varie d'une méthode à l'autre.

On a utilisé un maillage décalé (MAC, *Marker-And-Cell*) où les scalaires (pression, propriétés

du fluide, level-set, VOF) sont déterminés au centre de la cellule, et les vitesses sont au centre des faces. La discrétisation temporelle des équations de Navier-Stokes est réalisée par le schéma prédicteur-correcteur Crank-Nicolson. Les termes visqueux sont traités de manière implicite afin de permettre d'utiliser des pas temps plus élevés. La méthode de projection non-incrémentale est utilisée et les termes convectifs sont discrétisés par le schéma ENO d'ordre 2. Nous avons alors gardé cette structure identique, et utilisé différentes approches pour : le suivi de l'interface, la définition des propriétés du fluide et la prise en compte de la tension de surface.

On a ensuite construit 6 solveurs basés sur level-set et 4 sur VOF. Les solveurs basés sur level-set diffèrent principalement dans la manière suivant laquelle les conditions de l'interface sont traitées, à savoir :

- Variation lisse de la pression (CSF) et des propriétés physiques à travers l'interface et un traitement lisse de la force de tension de surface en considérant une interface épaisse. Le solveur LS-CSF utilise cette approche avec une courbure donnée par Chang et al. (1996).
- Saut raide de la pression (GFM) et des propriétés physiques à travers l'interface, et un traitement raide de la tension de surface en utilisant la méthode GFM. Le solveur LS-GFM1 utilise cette approche et est similaire au travail de Son and Dhir (2007).
- Saut raide de pression (GFM) et de la densité, et une viscosité lissée à travers l'interface. Deux solveurs utilisent cette approche : LS-GFMϵ (avec une courbure donnée par Chang et al. (1996)) et LS-GFMk (avec une courbure calculée selon Marchandise et al. (2007))
- Saut raide de la pression (GFM) à travers l'interface avec des propriétés physiques localement moyennées. Les solveurs LS-LA1 (avec une courbure donnée par Vigneaux (2008)) et LS-LA2 (où la courbure est calculée selon Sussman et al. (1999)) utilisent cette approche.

On a aussi construit 4 différentes méthodes VOF (dont une basée sur MOF) avec des propriétés pondérées et déterminées à partir de la fraction volumique. Deux méthodes de calcul de la courbure ont été utilisées : la fonction de hauteur (*Height Function*) de Popinet (2009), notée HF1, et celle de Hernández et al. (2008) notée HF2. De plus, nous avons implémenté un solveur des équations de Navier-Stokes basé sur un maillage centré, similaire à celui du code de calcul Basilisk (Popinet et al.), noté VOF-WYC, afin de pouvoir déterminer si la disposition des variables sur le maillage joue un rôle important sur la qualité des résultats. Il est à noter que le suivi d'interface de cette dernière méthode est réalisé avec la méthode VOF (Weymouth and Yue, 2010).

La confrontation de ces différentes méthodes est réalisée sur diverses simulations : les courants parasites (se développant dans une bulle stationnaire en l'absence de gravité), bulle qui oscille dans un champ de gravité nul, onde capillaire stationnaire, instabilité de Rayleigh-Taylor et la bulle ascendante dans un fluide au repos.

Les résultats obtenus ont montré que les solveurs diphasiques basés sur les méthodes VOF sont beaucoup plus stables et robustes à l'exception du solveur qui utilise VOF avec la fonction de hauteur de Hernández et al. (2008) pour l'estimation de la courbure de l'interface. Les solveurs basés sur level-set sont plus ou moins sensibles au cas traité. On a identifié deux solveurs basés sur level-set (LS-LA2 et LS-GFMϵ), qui donnent approximativement de bons résultats pour toutes les configurations. On a également noté que le solveur basé sur la méthode MOF est beaucoup plus sensible et donne de mauvais résultats lorsque le pas de temps (qui est fixé par la physique qui régit la simulation) est petit.

On présente les résultats de deux différentes simulations : l'onde capillaire stationnaire et la bulle ascendante. Dans la première, une interface sinusoïdale sépare deux fluides 1 et 2 de densité et viscosité uniformes. La figure 2 trace l'évolution de la hauteur de l'interface pour

les différentes méthodes sur différents maillages. La solution analytique Prosperetti (1981) est également tracée. Cette figure montre bien que les solveurs basés sur la méthode VOF sont plus précis que ceux de level-set. Les méthodes VOF prédisent correctement la fréquence d'oscillation de l'interface sur tous les maillages testés.

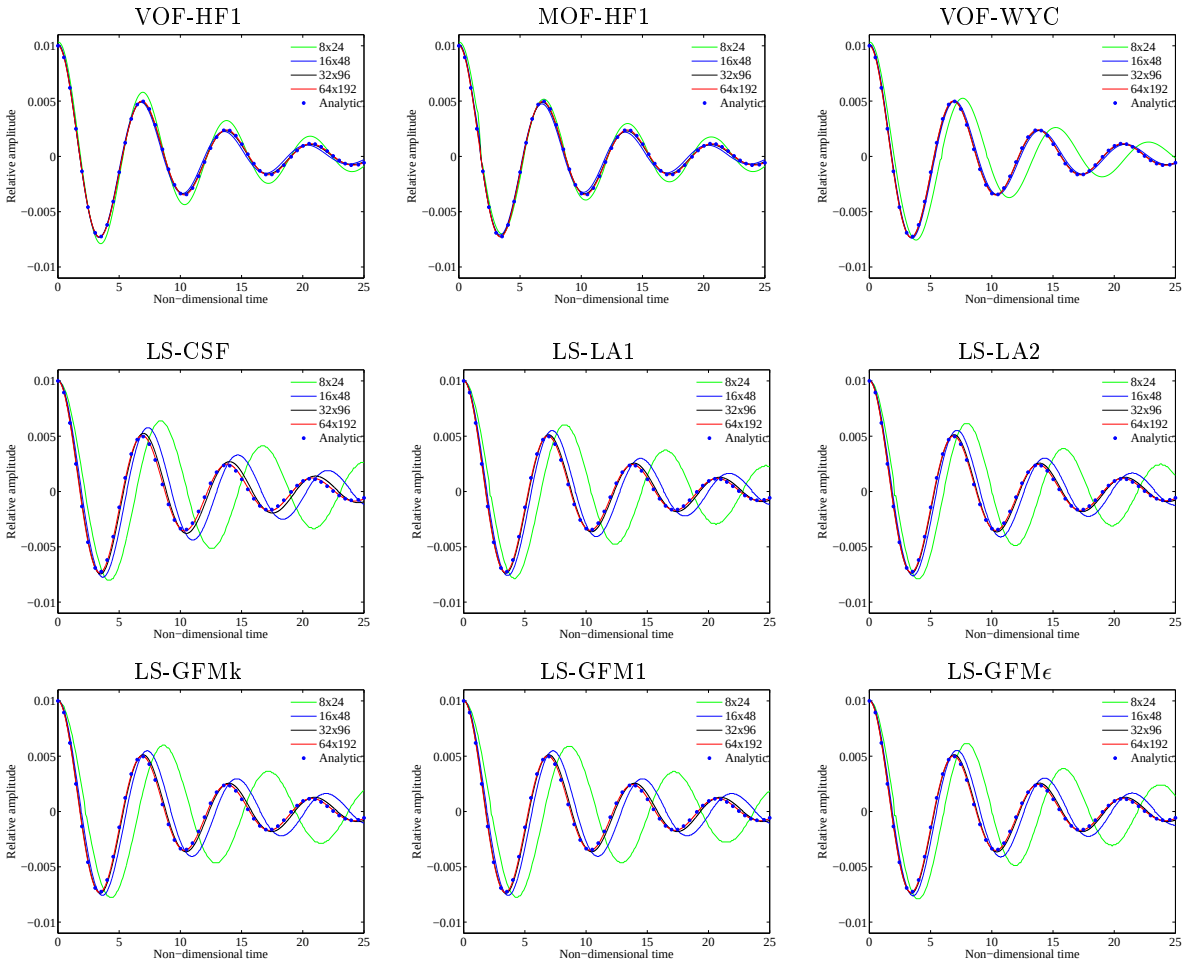

Figure 2: Onde capillaire stationnaire avec une densité et viscosité uniformes : évolution de la hauteur de l'interface à $x = 0$ avec CFL=CFL$_\sigma = 0.5$.

Le deuxième exemple concerne l'ascension d'une bulle initialement au repos (benchmark 2, Hysing et al. (2009)). L'interface au temps t=3 est présentée sur la figure 3 pour 6 différentes méthodes. On voit que la forme de la bulle diffère d'une méthode à une autre, et c'est ce qui est aussi observé par Hysing et al. (2009). Pour ce dernier cas, aucune solution analytique n'est disponible, mais nous avons utilisé le résultat de référence de (Featflow).

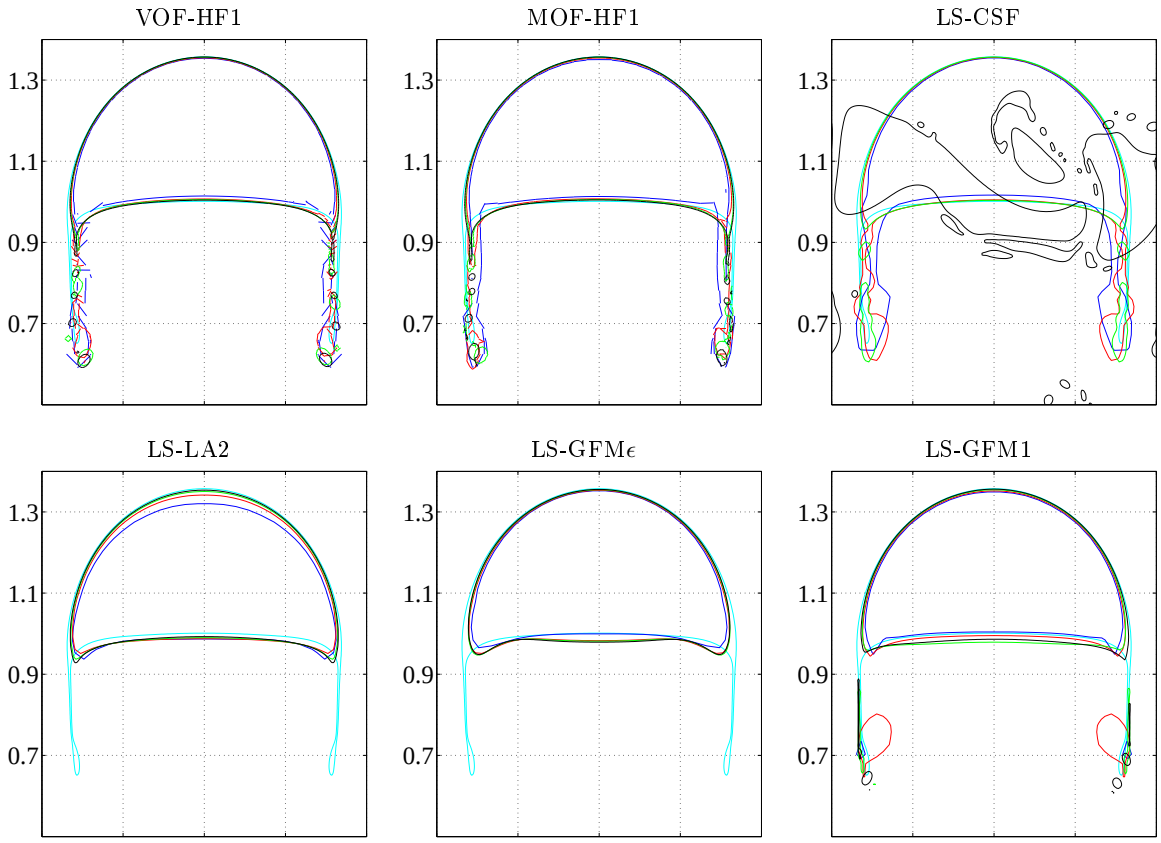

Figure 3: Assension d'une bulle (benchmark 2) : forme de la bulle à $t = 3.0$. Les couleurs font référence au maillage : Bleu (32×64), vert (64×128), rouge (128×256), noir (256×512) et cyan (solution de reference (Featflow)).

Conclusion

Dans ce travail, on a pu identifier une méthode level-set qui est stable et donne de bons résultats sur les cas tests académiques avec un champ de vitesse imposé en 2D cartésiennes. Nous l'avons comparée avec les méthodes VOF, MOF et les méthodes couplées.

La méthode MOF et la méthode couplée CLSMOF ont donné d'excellents résultats pour les problèmes avec champ de vitesse imposé. Cependant, de petites particules fluides se détachent de l'interface lorsque la méthode MOF est utilisée pour suivre l'interface en champ de vitesse donné pour des CFL petits. Ces problèmes ont affecté la stabilité du solveur diphasique dans lequel la méthode MOF est intégrée, ce qui a fortement impacté les résultats. L'utilisation de la méthode CLSMOF à la place de MOF dans ce solveur garantira ainsi de bons résultats vu que dans cette approche, level-set corrige le champ des fractions volumiques et supprime les particules fluides se détachant du fluide de référence. Ceci n'est néanmoins valable que si la prise en compte de l'interface dans le solveur des équations de Navier-Stokes est fiable et précise.

Les travaux futurs incluront la validation du solveur de Navier-Stokes avec les méthodes couplées en 2D, puis par la validation du solveur axisymétrique avec les méthodes VOF et level-set. Ceci va permettre d'étudier, par exemple, les bulles, les gouttes et les écoulements dans des canaux circulaires. L'extension du code de calcul en 3D ouvrira de nouvelles perspectives et permettra de traiter une large gamme de configurations.

Inclure l'équation de l'énergie et les conditions appropriées (pour la prise en compte de la variation de masse du fluide de référence due au changement de phase) va permettre de simuler la croissance d'une bulle lors du changement de phase. D'autres phénomènes tels que l'ébullition en film peuvent être ainsi prédits sur des surfaces plates. La prise en compte du changement de phase sur la surface irrégulière d'un solide nécessitera l'application de bonnes conditions aux limites de vitesse (à l'aide de la méthode de la frontière immergée par exemple).

Les simulations numériques en 3D sont très couteuses, surtout si les calculs sont séquentiels (ce qui est le cas du code développé actuellement). Deux techniques largement utilisées permettent de réduire significativement le coût de calcul : la première consiste à paralléliser le code via MPI; la seconde consiste à utiliser un maillage adaptatif avec raffinement local. Dans ce contexte, l'utilisation des codes open-source tels que Gerris (Popinet) et Basilisk (Popinet et al.) pourrait être une voie intéressante. Ceci permettra d'aborder des problèmes réels et aidera à la compréhension des phénomènes physiques liés au changement de phase.

Remerciements

Je remercie tout d'abord **Shihe XIN**, mon directeur de thèse pour ses conseils, sa disponibilité et sa confiance.

Mes plus vifs remerciements vont à **Ronnie KNIKKER**, mon co-directeur de thèse. Merci pour avoir toujours su me conseiller, m'orienter, m'aider, me soutenir et te rendre disponible!

Je tiens également à remercier **Eric ALBIN** pour les corrections, les conseils, l'aide et les discussions intéressantes.

Je remercie chaleureusement **Eric CHENIER** et **Christophe JOSSERAND** d'avoir accepté de lire en détail ce manuscrit. Je remercie également **Peter SPELT** et **Virginie DARU** qui ont accepté d'évaluer mon travail.

Je remercie aussi les doctorants/Postdoc/ingénieurs/stagiaires du laboratoire et les permanents (**A. Trabelsi**, **M. Amara**, **M. Boutaous**, **S. CIOULACHTJIAN**, ...) pour les moments partagés ensemble. Je remercie également l'équipe du secrétariat (**Florence**, **Christine**, **Corrine**, ...) pour leur disponibilité (eh oui, je viens toujours vous voir à la dernière minute!) et le directeur du laboratoire **Jocelyn BONJOUR**. Mes remerciements vont aussi à l'équipe Nano-Thermique, surtout à **P.-O. CHAPUIS** pour les discussions et conseils (spécialement après 21h ou les weekend! Pardon, le weekend c'est seulement vendredi soir, n'est-ce pas Olivier?).

J'adresse mes remerciement à mes collègues de bureau de m'avoir soutenu (**Kim**, **Mostafa**, **Florine**, **Armand**, ...). Je remercie plus particulièrement **Christophe K.** de m'avoir supporté durant un peu plus que 3 ans, merci pour toutes les discussions variées et enrichissantes sur l'environnement, les moteurs, l'histoire, ... et merci aussi pour les randonnées et sorties en vélo!

Je remercie également l'équipe du département mécanique de l'université Lyon 1 (**Pierre**, **Ivana**, **Delphine**, **Faouzi**, **Séverine**, **Christophe P.**, **Sylvie D.**, **Amandine P.**, ...).

Un grand merci à tous mes amis en France, en Algérie et au Canada.

Je remercie infiniment toute ma famille : mes parents, mes frères (**Salim** et **Abdelkrim**), mes soeurs et tous mes neveux et nièces pour leur amour et soutien inconditionnel. Je tiens aussi à remercier Lycia pour sa patience et son soutien.

Contents

Acronyms

VOF	Volume-of-fluid
MOF	Moment-of-fluid
LS	Level-set
CLSVOF	Coupled level-set and volume-of-fluid
CLSMOF	Coupled level-set and moment-of-fluid
VOSET	Coupled volume-of-fluid and level-set
MCLS	Mass-conserving level-set
GFM	Ghost Fluid Method
CSF	Continuous Surface Force
CSS	Continuous Surface Stress
LA	Local-averaging
HF	Height function
MAC	Marker And Cell
RHS	Right-hand side
CFL	Courant-Friedrichs-Lewy condition
CFLreinit	CFL number that sets when to reinitialize the interface
STD	Standard
VC	Volume constraint

General introduction

Two-phase flows concerning flows with two immiscible fluids are omnipresent in nature and in industrial processes. These flows often involve fluids with different phases (liquid-gas) characterized by high density and viscosity ratios. Sea waves and rivers are concrete examples involving water flow in interaction with a free surface. In the industrial processes and engineering applications, these flows are much more complicated and manifest under different forms. Bubbly flows, for example, are often encountered in heat exchangers. Inkjet printers are jetting devices in which ink is spread through a nozzle. This latter is very similar to what happens in combustion engines, except that the liquid jet is atomized in order to ensure perfect fuel-air mixing. This is a non-exhaustive list of examples. Phase change phenomena can occur under the influence of energy exchange. The experimental study of these phenomena is often fastidious and provides only correlations valid in some conditions. Numerical simulation seems to be a good alternative if the numerical tools are tested, validated and based on robust methods.

The immiscible nature of the involved fluids implies that they are separated by an interface. Many numerical methods devoted to follow the evolution of an arbitrary interface advected by the fluid flow are proposed in literature, called interface capturing/tracking methods. These methods allow a one fluid approach when coupled with the Navier-Stokes solver, i.e. only one set of equations is solved with variable properties (density, viscosity, ...). The numerical simulation of two-phase flows therefore depends strongly on the ability of the interface method to localize the interface.

The interface methods belong to one of these two categories: Lagrangian or Eulerian techniques. Lagrangian methods consist of seeding the interface with (massless) markers (Unverdi and Tryggvason, 1992), which are then advected with the local velocity. In this case we talk about interface tracking or front-tracking. On the other hand, Eulerian methods are based on the resolution of a scalar transport on a fixed Eulerian mesh and in this case we talk about interface capturing or front-capturing. The Lagrangian methods are generally more accurate than the Eulerian methods. However, many difficulties arise especially if the interface separating the two fluids undergoes major topological changes (coalescence and/or breakup). Sometimes, the interface is deformed to such a point that it is compulsory to add more markers in some regions in order to guarantee the accuracy of interface transport. On the other hand, the Eulerian methods naturally take into account the collapse and coalescence of the interface.

In this work, we focused on Eulerian techniques, in particular the level-set and the volume-of-fluid methods. In the level-set (LS) approach, the scalar is the algebraic distance to the interface. This scalar is the volume fraction of a reference fluid (liquid or gas) inside each computational cell in the volume-of-fluid (VOF) approach. The level set method permits to access easily to the geometrical properties of the interface, while the mass of the reference fluid is not fully conserved. From the other hand, the volume-of-fluid method conserves the mass but sometimes small bodies

detach from the reference fluid (often called flotsam and jetsam).

The ultimate objective of this work is to study two-phase flows with phase change. This thesis focuses on the study and comparison of interface capturing methods for two-phase flows, which is split it into two steps. The first one aims to select robust methods able to accurately predict interface transport in imposed velocity field. The second one deals with the investigation of the different manners of coupling the interface methods with the incompressible Navier-Stokes solver.

Many different implementations of the level-set method are proposed in the literature. We recall them in Chapter 1, and compare them in accurately transporting interfaces in fixed velocity fields. The second chapter is devoted to the volume-of-fluid techniques, where a literature review is presented. The moment-of-fluid method (MOF), which in addition to the fraction uses the centroid information is also detailed. Finally, similar test cases to those of the first chapter are performed to evaluate the performances of each method.

Since the level-set method suffers from mass loss/gain, the volume-of-fluid method from the interface representation, Bourlioux (1995) had the idea to couple both methods in one method. Coupled techniques became widely used after the pioneering work of Sussman and Puckett (2000) then Son and Hur (2002). Later, many methods are proposed in the literature such as the mass conserving level-set (MCLS) method, the coupled volume-of-fluid and level-set (VOSET) method and coupled level set and moment-of-fluid (CLSMOF) method. These coupling techniques are presented in Chapter 3, and differences between them are highlighted. A comparison is performed with the standalone methods of Chapters 1 and 2 in order to show the benefits of the coupling.

Chapter 4 deals with the two-phase flow solver. It is based on the incompressible Navier-Stokes equations coupled with the interface method. The fluids are considered Newtonian and immiscible. Different manners of including the effect of the different physical properties and phenomena such as viscosity and surface tension are available in the literature, and the most popular formulations are implemented in our solver. In total, we have tested 11 methods: 4 are coupled with the volume-of-fluid method, 1 with the moment-of-fluid method and 6 with the level-set method. We compared the different implementations in accurately reproducing well known test cases in two-dimensional cartesian grids.

Chapter 1

Level-set method

1.1 Introduction

In the level-set approach, an interface is defined by the zero level-set of a continuous function. This function should change its sign when crossing the interface, i.e. it should be positive (or negative) inside the domain containing the reference fluid and negative (or positive) outside it.

Although an infinity of functions satisfy the previous condition, the algebraic distance function is generally selected for its regularity and other interesting properties. The signed-distance function allows us for example to define an infinity of iso-levels, each one corresponding to a constant distance from the interface.

Given an interface defined by a level-set function ϕ under velocity field $\vec{u} = (u, v)$ (or $\vec{u} = (u, v, w)$ in 3D), the evolution of the interface Γ is governed by the advection equation:

$$\frac{\partial \phi}{\partial t} + \vec{u} \cdot \nabla \phi = 0 \tag{1.1}$$

The level-set function permits the easy access to interface geometrical properties such as the normal to the interface

$$\vec{n} = \frac{\nabla \phi}{|\nabla \phi|}$$

and the curvature

$$\kappa = \nabla \cdot \vec{n}$$

The geometrical properties are easily obtained from the level-set. However, the precision in determining these geometrical properties depends on the accuracy of localizing the zero level-set. As any other numerical method, the level-set suffers from drawbacks. The main disadvantage of the level-set is that the numerical methods that solve Eq. (1.1) do not conserve exactly the mass of the reference fluid. On the contrary, it automatically handles the topological changes that an interface can undergo such as the coalescence and/or the break-up.

In §1.2, we will see how to advect an arbitrary interface Γ defined by the zero level-set. The velocity field is often non-uniform, which disturbs the signed-distance property and introduces irregularities in the level-set function when only Eq. (1.1) is solved. The remedy to this is to solve in regular intervals a supplementary equation called the reinitialization equation, which reset ϕ as the signed-distance function without moving the zero level-set. This step will be detailed in §1.3. Solving the advection and reinitialization equations on the entire domain is numerically expensive, we present therefore in §1.4 a method to solve these equations on a narrow band around the interface. Finally, in §1.5 and §1.6 the different numerical methods for advection and

reinitialization are evaluated and compared.

In fact, we choose the staggered (MAC) grid in which the scalars (the level-set value, ϕ) are stored at the cell centers and the velocity vectors at the center of cell-faces as shown on Fig.1.1. This grid configuration offers simplicity and easiness of using high-order schemes.

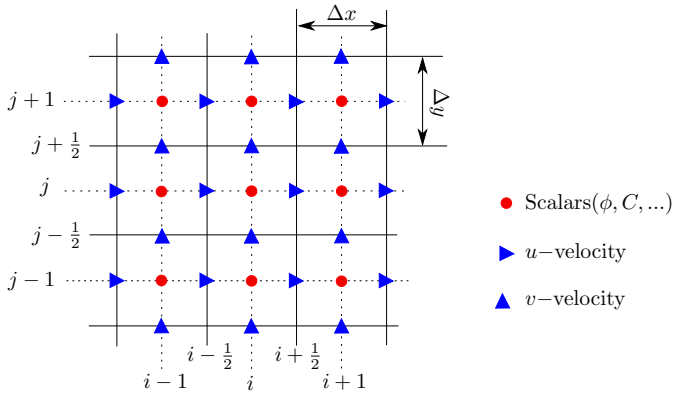

Figure 1.1: Staggered grid for the spatial discretization.

1.2 Numerical resolution of the advection equation

1.2.1 Time discretization

The third-order Runge-Kutta TVD method (Shu and Osher, 1989) is a popular choice in combination with ENO and WENO schemes, we use it in the current work. Let ϕ^n be the solution of Eq. (1.1) after n time steps. The solution at the next step is computed from:

$$\phi^{(1)} = \phi^n + \Delta t L(\phi) \tag{1.2}$$

$$\phi^{(2)} = \frac{3}{4}\phi^n + \frac{1}{4}\phi^{(1)} + \frac{1}{4}\Delta t L(\phi^{(1)}) \tag{1.3}$$

$$\phi^{n+1} = \frac{1}{3}\phi^n + \frac{2}{3}\phi^{(2)} + \frac{2}{3}\Delta t L(\phi^{(2)}) \tag{1.4}$$

where $\phi^{(1)}$ and $\phi^{(2)}$ are intermediate values, ϕ^{n+1} is the solution at time t^{n+1} and L is the spatial discretization of the term $(-\vec{u}\cdot\nabla\phi)$ in Eq. (1.1) which is described in the next sections.

1.2.2 Spatial discretization

The level-set advection equation can be written in the form of Eq. (1.1), which is non-conservative since its resolution is based on point values. In 2D, the discrete form looks like:

$$\frac{\partial \phi_{i,j}}{\partial t} + u_{i,j}\left.\frac{\partial \phi}{\partial x}\right|_{i,j} + v_{i,j}\left.\frac{\partial \phi}{\partial y}\right|_{i,j} = 0 \tag{1.5}$$

where $()_{i,j}$ denotes a quantity defined at the grid point (x_i, y_j).

For incompressible fluids, a conservative formulation can be derived by re-writing Eq. (1.1) in a finite-volume sense as follows:

$$\frac{\partial \phi}{\partial t} + \nabla\cdot(\vec{u}\phi) = 0 \tag{1.6}$$

We can solve this equation directly in a finite-difference method, or in the weak sense using a finite-volume method. The solution $\bar{\phi}_{i,j}$ in a finite-volume sense is the integral value of the function on the cell of area $\Omega_{i,j}$. The Gauss-Ostrogradsky theorem allows the transformation of the integral of the second term into a surface integral representing the net flux of the conserved quantity across cell faces:

$$\int_{\Omega_{i,j}} \frac{\partial \phi}{\partial t} d\Omega + \int_{\Omega_{i,j}} \nabla \cdot (\vec{u}\phi) d\Omega = \Omega_{i,j} \frac{d}{dt} \overline{\phi_{i,j}} + \int_{\partial\Omega_{i,j}} (\vec{u}\phi) \cdot \vec{n}\, dS = 0 \tag{1.7}$$

where $\Omega_{i,j}$ is the volume of the grid cell and $\partial\Omega_{i,j}$ its boundary. In the 2D cartesian grid setting, the discretization of this equation can be written as:

$$\frac{d}{dt}\phi_{i,j} + \frac{F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j}}{\Delta x} + \frac{G_{i,j+\frac{1}{2}} - G_{i,j-\frac{1}{2}}}{\Delta y} = 0$$

where F are the numerical fluxes on the left and right boundaries of $\Omega_{i,j}$, and G on the upper and lower boundaries. For example, the upper and right fluxes are defined as:

$$\begin{cases} F_{i+\frac{1}{2},j} = \displaystyle\int_{y_{i,j-\frac{1}{2}}}^{y_{i,j+\frac{1}{2}}} u(x_{i+\frac{1}{2}}, y)\phi(x_{i+\frac{1}{2}}, y) \quad dy \\ G_{i,j+\frac{1}{2}} = \displaystyle\int_{x_{i-\frac{1}{2},j}}^{x_{i+\frac{1}{2},j}} v(x, y_{j+\frac{1}{2}})\phi(x, y_{j+\frac{1}{2}}) \quad dx \end{cases} \tag{1.8}$$

The advection equation (1.1) is a Hamilton-Jacobi type differential equation. Centered finite-difference schemes are not suitable for Hamilton-Jacobi equations because they do not capture information from the correct direction of propagation, and tend to introduce oscillations in the solution. In the next section, we will explain the Essentially Non-Oscillatory (ENO) and Weighted ENO (WENO) schemes for the level-set advection equation (1.1). Both are commonly used in level-set solvers. ENO schemes were first developed to treat shock waves and discontinuities by Harten et al. (1987), and have been extended to Hamilton-Jacobi equations later in Osher and Sethian (1988) and Osher and Shu (1991).

The schemes we are going to detail hereafter will give approximations to the derivative $\phi_x = \partial\phi/\partial x$ which is computed on a stencil biased to the left or the right, noted as ϕ_x^- and ϕ_x^+, respectively. The choice of which estimation to use depends on the direction of propagation of the characteristic functions of the Hamilton-Jacobi equation. In the case of the level-set advection equation, the characteristic speed is simply the velocity u. In the reinitialization equation, the expressions for the direction of propagation are slightly more complex. Once the direction is known, upwinding consists in selecting the estimation for ϕ_x which is based on a stencil biased in the opposite direction of the characteristics (in the direction where the information comes from). For the advection equation, we have:

$$\frac{\partial \phi}{\partial x} = \phi_x = \begin{cases} \phi_x^- \text{ if } u > 0 \\ \phi_x^+ \text{ if } u < 0 \end{cases} \tag{1.9}$$

There are other ways to estimate ϕ_x from ϕ_x^- and ϕ_x^+, which are not used in this work however. The Lax-Friedrichs schemes use the average between the left and the right derivatives and adds some numerical viscosity. The Roe-Fix scheme chooses between the left and the right derivatives by an upwinding based method and whilst taking into account coefficients for entropy-fix (for the Navier-Stokes equations). Neglecting the entropy-fix coefficients, the Roe-Fix algorithm is reduced also to upwinding. The Godunov's scheme yields the same choice procedure in the

particular case of the level-set advection. For more information on these schemes applied to the level-set equations, see Osher and Fedkiw (2002).

The level-set is treated in a direction-by-direction fashion (as in the method of lines), which means that the above scheme is applied in the same manner to each spatial derivative in Eq. (1.5). Only the expressions for the x-direction are shown in the following, for the sake of brevity, and we drop the subscript j in order to alleviate notations.

1.2.2.1 First-order upwind scheme

It gives a first-order approximation of the term ϕ_x. The derivatives at the grid point i based on the left- and right-biased stencils are given by:

$$(\phi_x^-)_i = \frac{\phi_i - \phi_{i-1}}{\Delta x} \tag{1.10}$$

$$(\phi_x^+)_i = \frac{\phi_{i+1} - \phi_i}{\Delta x} \tag{1.11}$$

1.2.2.2 Hamilton-Jacobi ENO schemes

First, let us introduce the general kth-order ENO scheme, for $k = 1, 2$ or 3. The values of ϕ_x^- and ϕ_x^+ are calculated using the ENO scheme proposed by Osher and Sethian (1988) then generalized by Osher and Shu (1991), which proceeds as follows:

Step 1. We first compute the so-called undivided differences of degree $j \leq k$, noted V_i^j. Starting with the value ϕ_i on a grid cell i, the undivided difference of degree zero is given by

$$V_i^0 \equiv \phi_i \tag{1.12}$$

and the undivided differences of degree $j \geq 1$ are subsequently computed from the recurrent relation

$$V_i^j \equiv V_{i+1}^{j-1} - V_i^{j-1} \tag{1.13}$$

The j-th undivided difference may be interpreted as an approximation of the j-th derivative such that $V_i^j \simeq (\Delta x)^j (\partial^j \phi / \partial x^j)$, based on a stencil of $j+1$ points with x_i as the left-most point.

Step 2. For a given grid point i, the selection procedure for the ENO stencil starts with the first-order polynomial approximation of ϕ based on the values of ϕ on the two-point stencil $[x_i, x_{i+1}]$:

$$P^1(x) = V_i^0 + V_i^1 (x - x_i)/\Delta x \tag{1.14}$$

An approximation of ϕ_x is then calculated from

$$(\phi_x^+)_i \simeq \frac{\partial P^1}{\partial x} = \frac{1}{\Delta x} V_i^1 = \frac{\phi_{i+1} - \phi_i}{\Delta x} \tag{1.15}$$

Step 3. The order of accuracy is then progressively increased by adding one data point on the left or right side of the stencil. Let us denote k the order of the new polynomial function $P^k(x)$ and m the first (left-most) grid point of the new stencil. We started the procedure in Eq. (1.14) with $m = i$ and $k = 1$. The choice of the new stencil is based on the minimization of the absolute value of the undivided difference that will be added to the approximation:

$$\begin{aligned} &k = k + 1 \\ &\text{if} \quad |V_m^k| > |V_{m-1}^k| \quad \text{then} \quad m = m - 1 \end{aligned}$$

In this way, the approximation will always be based on the stencil containing the smoothest data set.

Step 4. At the level k, the stencil contains $k+1$ grid points $[x_m, ..., x_{m+k}]$ on which a unique polynomial $P^k(x)$ of order k can be fitted, from which a new approximation $(\phi_x^+)_i = (\partial P^k/\partial x)_i$ of order k can be computed. Several expressions for $(\phi_x^+)_i$ can be found in the literature. Since $P^k(x)$ is unique, they all lead to the same results. The formulation used in this work is based on a linear combination of the first-degree undivided differences (Shu, 1997):

$$(\phi_x^+)_i = \frac{1}{\Delta x}\sum_{j=0}^{k-1} C_{r-1,j}\, V^1_{i-r+j} \tag{1.16}$$

$$(\phi_x^-)_{i+1} = \frac{1}{\Delta x}\sum_{j=0}^{k-1} C_{rj}\, V^1_{i-r+j} \tag{1.17}$$

where $r = m - i$ and the coefficients C_{rj} are given by

$$C_{rj} = \sum_{m=j+1}^{k} \frac{\sum_{\substack{l=0\\ l\neq m}}^{k}\prod_{\substack{q=0\\ q\neq m,l}}^{k}(r-q+1)}{\prod_{\substack{l=m\\ l\neq s}}^{m+k-1}(m-l)} \tag{1.18}$$

An expression for C_{rj} for a non-uniform grid, based on the first-degree *divided* differences, can be found in Shu (1997).

Note that, even though the higher-degree undivided differences do not appear in Eqs. (1.16) and (1.17), they are still needed in the ENO selection procedure. The order of accuracy is reduced near the boundary points ($i = 1$ and $i = N$) to avoid the use of non-existing grid points:

$$k = \min(k_{max}, \min(i, N-i))$$

where k_{max} is the order of accuracy for inner grid points and N is the number of cells. The quantities $(\phi_x^-)_1$ and $(\phi_x^+)_N$ are not computed and set to zero. This will generally provide a smooth treatment of the level-set near the boundary and avoid generation of unwanted oscillations. Note that in the examples shown in this work, the interface never reaches the boundary of the domain. Correct implementation of boundary points and interfaces connecting with the boundary are postponed to future work.

1.2.2.3 Hamilton-Jacobi WENO schemes

The Weighted ENO schemes are due to Liu et al. (1994) then improved by Jiang and Shu (1996). It has been later extended to Hamilton-Jacobi equations by Jiang and Peng (2000). The procedure starts with the computation of the divided differences, although only those of degree one are required here:

$$V_i^1 = \frac{\phi_{i+1} - \phi_i}{\Delta x} \tag{1.19}$$

The derivatives ϕ_x are evaluated in a cell-by-cell manner. For the i-th cell, the WENO approximation uses the symmetric stencil $[x_{i-(k-1)}, ..., x_{i+(k-1)}]$ where $k = 2$ for the third-order scheme

and $k = 3$ for the fifth-order scheme. The left- and right approximations of ϕ_x are computed from a linear combination of a set of kth-order polynomial approximations:

$$(\phi_x^+)_i = \sum_{r=1}^{k} \omega_r^+ (\phi_x^+)_i^r \tag{1.20}$$

$$(\phi_x^-)_{i+1} = \sum_{r=1}^{k} \omega_r^- (\phi_x^-)_i^r \tag{1.21}$$

where $(\phi_x^\pm)_i^r$ for $r = 0...k-1$ are the kth-order polynomial approximations, based on different stencils, and $\omega_r^\pm$ are the weights which provide a final order of accuracy of $2k-1$ for smoothly varying functions. For $k = 2$, the polynomials are:

$$\begin{aligned}
(\phi_x^+)_i^1 &= -\frac{1}{2}V_{i+1}^1 + \frac{3}{2}V_i^1 \quad \text{with stencil } [x_i, ..., x_{i+2}] \\
(\phi_x^+)_i^2 &= \frac{1}{2}V_i^1 + \frac{1}{2}V_{i-1}^1 \quad \text{with stencil } [x_{i-1}, ..., x_{i+1}]
\end{aligned}$$

We only provide the expressions for $(\phi_x^+)_i$. Those for $(\phi_x^-)_i$ can be obtained from symmetry considerations around the grid point i. For $k = 3$, the polynomials are:

$$\begin{aligned}
(\phi_x^+)_i^1 &= \frac{1}{3}V_{i+2}^1 - \frac{7}{6}V_{i+1}^1 + \frac{11}{6}V_i^1 \quad \text{with stencil } [x_i, ..., x_{i+3}] \\
(\phi_x^+)_i^2 &= -\frac{1}{6}V_{i+1}^1 + \frac{5}{6}V_i^1 + \frac{1}{3}V_{i-1}^1 \quad \text{with stencil } [x_{i-1}, ..., x_{i+2}] \\
(\phi_x^+)_i^3 &= \frac{1}{3}V_i^1 + \frac{5}{6}V_{i-1}^1 - \frac{1}{6}V_{i-2}^1 \quad \text{with stencil } [x_{i-2}, ..., x_{i+1}]
\end{aligned}$$

Figure 1.2b shows the three sub-stencils used to compute these third-order accurate derivatives.

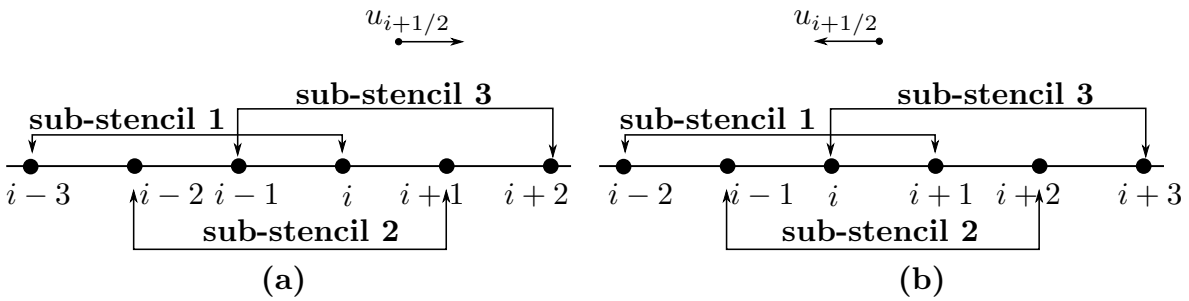

Figure 1.2: 5-th order Hamilton-Jacobi WENO scheme stencils for the derivative approximation at the grid point i, (a) $(\phi_x^-)_i$ and (b) $(\phi_x^+)_i$.

Next, the weights are defined by :

$$\omega_r^+ = \frac{\widetilde{\omega}_r^+}{\sum_{s=1}^{k} \widetilde{\omega}_s^+} \quad \text{with} \quad \widetilde{\omega}_r^+ = \frac{d_r}{(\varepsilon + \beta_r)^2} \tag{1.22}$$

We verify that $\sum_{r=1}^{k} \omega_r^+ = 1$. The coefficients β_r are smoothness indicators and the small parameter ε avoids division by zero and is set to

$$\varepsilon = 10^{-6} \max\{(V_{i-2}^1)^2, (V_{i-1}^1)^2, (V_i^1)^2, (V_{i+1}^1)^2, (V_{i+2}^1)^2\} + 10^{-16}$$

as recommended by Fedkiw et al. (2000). The weights d_r are determined such that the highest order of accuracy $(2k-1)$ for smoothly varying functions is obtained, leading to

$$d_1 = \frac{1}{3}, \quad d_2 = \frac{2}{3}$$

for $k = 2$ and

$$d_1 = \frac{1}{10}, \quad d_2 = \frac{3}{5}, \quad d_3 = \frac{3}{10}$$

for $k = 3$. The smoothness indicators β_r allow the scheme to decrease the weights of those stencils where the function shows sharp variations. For $k = 2$, they are given by :

$$\begin{aligned} \beta_1 &= (V^1_{i+1} - V^1_i)^2 \\ \beta_2 &= (V^1_i - V^1_{i-1})^2 \end{aligned} \tag{1.23}$$

For $k = 3$, they are written :

$$\begin{aligned} \beta_1 &= \frac{13}{12}(V^1_{i+2} - 2V^1_{i+1} + V^1_i)^2 + \frac{1}{4}(V^1_{i+2} - 4V^1_{i+1} + 3V^1_i)^2 \\ \beta_2 &= \frac{13}{12}(V^1_{i+1} - 2V^1_i + V^1_{i-1})^2 + \frac{1}{4}(V^1_{i+1} - V^1_{i-1})^2 \\ \beta_3 &= \frac{13}{12}(V^1_i - 2V^1_{i-1} + V^1_{i-2})^2 + \frac{1}{4}(3V^1_i - 4V^1_{i-1} + V^1_{i-2})^2 \end{aligned} \tag{1.24}$$

1.2.2.4 High-order upstream central (HOUC) schemes

Nourgaliev and Theofanous (2007) have used high-order upwind finite-differences schemes to compute the left and the right derivatives in the level-set equations, and called it the HOUC schemes. The choice of the fixed stencil and coefficients (without weighting coefficients and smoothness indicators) was motivated by the reduction in computational time and increase in order of accuracy with respect to the WENO schemes. To test the HOUC schemes we use the upwinding as in Eq. (1.9) to choose between the left and the right derivatives. The derivatives based on the left-biased stencil are given by:

$$(\phi^-_x)_i|_{HOUC3} = \frac{1}{6}\phi_{i-2} - \phi_{i-1} + \frac{1}{2}\phi_i + \frac{1}{3}\phi_{i+1} + O(\Delta x^3) \tag{1.25}$$

$$(\phi^-_x)_i|_{HOUC5} = -\frac{1}{30}\phi_{i-3} + \frac{1}{4}\phi_{i-2} - \phi_{i-1} + \frac{1}{3}\phi_i + \frac{1}{2}\phi_{i+1} - \frac{1}{20}\phi_{i+2} + O(\Delta x^5) \tag{1.26}$$

$$(\phi^-_x)_i|_{HOUC7} = \frac{1}{140}\phi_{i-4} - \frac{1}{15}\phi_{i-3} + \frac{3}{10}\phi_{i-2} - \phi_{i-1} + \frac{1}{4}\phi_i + \frac{3}{5}\phi_{i+1} - \frac{1}{10}\phi_{i+2} + \frac{1}{105}\phi_{i+3} + O(\Delta x^7) \tag{1.27}$$

$$(\phi^-_x)_i|_{HOUC9} = -\frac{1}{630}\phi_{i-5} + \frac{1}{56}\phi_{i-4} - \frac{2}{21}\phi_{i-3} + \frac{1}{3}\phi_{i-2} - \phi_{i-1} + \frac{1}{5}\phi_i + \frac{2}{3}\phi_{i+1} - \frac{1}{7}\phi_{i+2} + \frac{1}{42}\phi_{i+3} - \frac{1}{504}\phi_{i+4} + O(\Delta x^9) \tag{1.28}$$

$$(\phi_x^-)_i|_{HOUC11} = \frac{1}{2772}\phi_{i-6} - \frac{1}{210}\phi_{i-5} + \frac{5}{168}\phi_{i-4} - \frac{5}{42}\phi_{i-3} + \frac{5}{14}\phi_{i-2} - \phi_{i-1} + \frac{1}{6}\phi_i + \frac{5}{7}\phi_{i+1} - \frac{5}{28}\phi_{i+2} + \frac{5}{126}\phi_{i+3} - \frac{1}{168}\phi_{i+4} + \frac{1}{2310}\phi_{i+5} + O(\Delta x^{11}) \quad (1.29)$$

where the stencil length depends on the considered order. The right-biased stencil based derivatives are obtained by symmetry. Note that the coefficients for the 11th-order scheme are derived here from Taylor expansions and are slightly different from those provided by Nourgaliev and Theofanous (2007).

1.2.2.5 ENO schemes for conservation equations

The ENO schemes were initially proposed for hyperbolic conservation equations (Harten et al., 1987; Shu and Osher, 1989) in order to compute the numerical fluxes at the cell boundaries. Both finite-volume and finite-difference schemes have been proposed (Shu, 1997). The level-set advection equation (1.6), although not derived from a conservation law, can be written as a conservation equation. In one-dimensional space we have for example

$$\frac{\partial \phi}{\partial t} + \frac{\partial f}{\partial x} = 0 \quad (1.30)$$

where $f = u\phi$ is the numerical flux. The main idea of the finite-difference ENO scheme for this type of conservation equations is to start from point values at cell centers (ϕ_i, $f_i = u_i\phi_i$, etc.) and to express the numerical fluxes at the left and at the right cell faces ($\widehat{f}_{i-1/2}$ and $\widehat{f}_{i+1/2}$) in such a way that its first-order divided differences represents a high-order approximation of the flux spatial derivative. Eq. (1.30) is therefore approximated by the semi-discrete equation [1]

$$\frac{\partial \phi_i(t)}{\partial t} + \frac{\widehat{f}_{i+1/2} - \widehat{f}_{i-1/2}}{\Delta x} = 0 \quad (1.31)$$

In the Hamilton-Jacobi ENO scheme, we have expressed the left and the right derivatives as a function of the first-order undivided differences. In the conservative formulation (1.31), we express the right and the left fluxes by the same formulas as in Eqs. (1.16-1.17) by replacing the first-order undivided differences with the zero-th-order undivided differences. The starting stencil would be then $\left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right]$, and if we want to increase the order of polynomials we add either the left point $x_{i-\frac{3}{2}}$ or the right one $x_{i+\frac{3}{2}}$ to the stencil by choosing the smoothest resulting stencil as explained in the " Step 3 " of standard ENO.

Given a grid point i, the expression of the numerical fluxes based on the right/left stencils are respectively:

$$\widehat{f}^-_{i+\frac{1}{2}} = \sum_{j=0}^{k-1} C_{rj} f_{i-r+j} \quad (1.32)$$

$$\widehat{f}^+_{i+\frac{1}{2}} = \sum_{j=0}^{k-1} C_{(r-1)j} f_{i-(r-1)+j} \quad (1.33)$$

where the constants C_{rj} are given by Eq. (1.18). For the advection equation (1.31), simple

[1] This scheme guarantees primary conservation of the variable ϕ, since the numerical fluxes in the interior of the domain cancel out when the cell values ϕ_i are summed, i.e.

$$\frac{\partial}{\partial t}\left(\sum_{i=1}^{N} \Delta x \phi_i\right) = \widehat{f}_{1/2} - \widehat{f}_{N+1/2}$$

upwinding is still used to choose between the numerical fluxes based on left- and right-handed stencils:

$$\widehat{f}_{i+\frac{1}{2}} = \begin{cases} \widehat{f}^{-}_{i+\frac{1}{2}} & \text{if} \quad u_{i+\frac{1}{2}} > 0 \\ \widehat{f}^{+}_{i+\frac{1}{2}} & \text{if} \quad u_{i+\frac{1}{2}} < 0 \end{cases}$$

Note that the cell-centered velocity, u_i, is used to construct the numerical fluxes, whereas values on cell faces, $u_{i+\frac{1}{2}}$, are needed to select the upwind direction. In case the cell-centered values are not available (staggered arrangement), we use averaging, $\bar{u}_i = (u_{i-\frac{1}{2}} + u_{i+\frac{1}{2}})/2$.

The level-set advection equation is not derived from a conservation law, and therefore the ENO scheme described above can be implemented in different conservative and non-conservative ways. The three formulations that have been tested in this work (with their corresponding abbreviations in brackets) are:

[Jia] The first one corresponds to the original conservative formulation, which expresses the advection equation in semi-discrete form as (Jiang and Shu, 1995):

$$\frac{\partial \phi_i(t)}{\partial t} + \frac{\widehat{(\phi\bar{u})}_{i+\frac{1}{2}} - \widehat{(\phi\bar{u})}_{i-\frac{1}{2}}}{\Delta x} = 0 \tag{1.34}$$

where the velocity $\bar{u}$ is the interpolated velocity at the cell center. For example, the flux $\widehat{(\phi\bar{u})}_{i+\frac{1}{2}}$ is obtained by taking $f_i = (\bar{u}\phi)_i$ in Eqs. (1.32) and (1.33).

[Est] Estivalezes (2002) proposed a slightly different formulation for the flux derivative:

$$\frac{\partial \phi_i(t)}{\partial t} + \frac{u_{i+\frac{1}{2}}\widehat{(\phi)}_{i+\frac{1}{2}} - u_{i-\frac{1}{2}}\widehat{(\phi)}_{i-\frac{1}{2}}}{\Delta x} = 0 \tag{1.35}$$

which does not require the interpolation of the velocity. This scheme is still conservative, but the accuracy is reduced to second-order since the velocity is not included in the high-order reconstruction step.

[Adv] In the current study, we add a third non-conservative (advective) formulation which is written as:

$$\frac{\partial \phi_i(t)}{\partial t} + \bar{u}_i\frac{\widehat{(\phi)}_{i+\frac{1}{2}} - \widehat{(\phi)}_{i-\frac{1}{2}}}{\Delta x} = 0 \tag{1.36}$$

where the velocity should again be interpolated from the cell faces to the cell centers. Here, we have used the advection equation (1.1) as a starting point (rather than the conservative formulation (1.6)) and the ENO scheme is merely used to compute the spatial derivative of the level-set function. The accuracy of the ENO scheme is preserved in the calculation of the spatial derivative, but the cell averaging of the velocity introduces a second-order error.

1.2.2.6 WENO schemes for conservation equations

The conservative WENO is a convex combination of the approximations of conservative ENO on possible stencils and it is derived following the same manner as the non-conservative WENO scheme in §1.2.2.3. The main difference is that the approximation of the derivatives becomes an approximation of the fluxes, and the zeroth-order undivided differences are used instead of the first-order undivided differences. After having expressed the numerical fluxes, the derivative comes from the first-order undivided differences of fluxes. Figure 1.3 below shows how the stencils

are used to approximate the level-set flux at grid points $(i+\frac{1}{2})$ where velocity is positive (Fig. 1.3a) and negative (Fig. 1.3b).

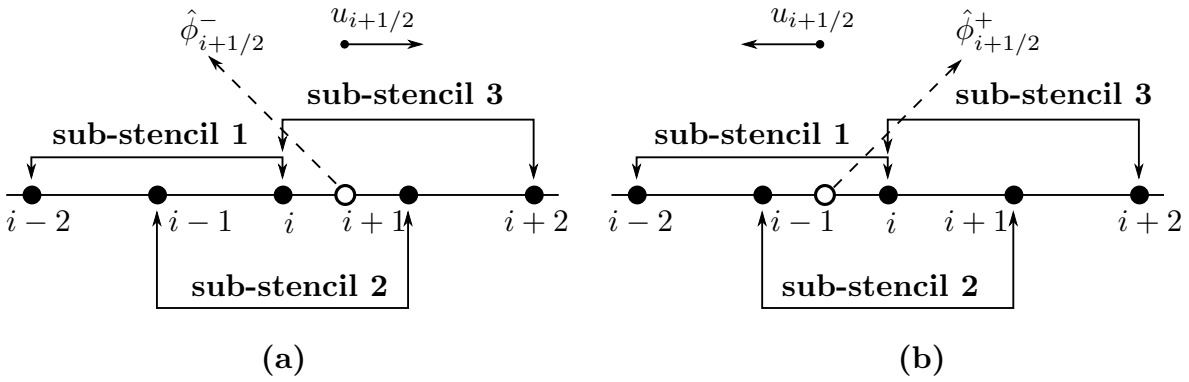

Figure 1.3: Stencils of the conservative 5th-order WENO scheme for the flux approximation.

As for the conservative ENO schemes, we have tested three different formulations for the 5th-order WENO scheme (Eq. 1.34, 1.35 and 1.36). The second formulation (1.35) has been used by Tanguy (2004), Zuzio (2010) and Voronetska (2012) in combination with the 5th-order WENO scheme.

1.2.2.7 HOUC schemes for conservation equations

The conservative HOUC scheme is built here in the same manner as the conservative ENO scheme. The difference is that the ENO scheme chooses the smoothest stencil, while the HOUC scheme uses a fixed stencil. For a given grid cell (i,j), the numerical flux at $(i+1/2,j)$ is computed using a stencil centered at that grid point. These fluxes based on the left and the right stencils are given, as in the conservative ENO schemes, by Eqs. (1.32) and (1.33) where the coefficients $C_{r,j}$ are given by Eq. (1.18). For the fifth-order reconstruction, for example, we set $r=2$ and the flux at $i+1/2$ of a given quantity f is:

$$\widehat{f}^-_{i+\frac{1}{2}} = \frac{3}{128}f_{i-2} - \frac{20}{128}f_{i-1} + \frac{90}{128}f_i + \frac{60}{128}f_{i+1} - \frac{5}{128}f_{i+2} \tag{1.37}$$

$$\widehat{f}^+_{i+\frac{1}{2}} = \frac{3}{128}f_{i+3} - \frac{20}{128}f_{i+2} + \frac{90}{128}f_{i+1} + \frac{60}{128}f_i - \frac{5}{128}f_{i-1} \tag{1.38}$$

where f could be either ϕ or $\phi\bar{u}$. The derivative is then computed using one of the formulations (1.34), (1.35) and (1.36).

1.2.3 Semi-Lagrangian advection

The semi-Lagrangian discretization of the level-set advection, proposed by (Wang et al., 2012a), consists in determining the interpolating point associated with each grid point (i,j). It is obtained by tracing it back along the characteristics from time t^{n+1} to t^n. The interpolating point is obtained using a two-step procedure (Wang et al., 2012a):

$$\begin{cases} x^{(1)} &= x_{i,j} - u(x_{i,j}, t^{n+1})\Delta t \\ x^{*} &= \dfrac{x_{i,j} + x^{(1)}}{2} \\ x^{(2)} &= x_{i,j} - u(x^{*}, t^{n+1/2})\Delta t \end{cases} \tag{1.39}$$

Then, by interpolating the level-set ϕ^n on the point $x^{(2)}$ we obtain the level-set value which would be on the grid point $x_{i,j}$ at time t^{n+1}. The interpolation of the level-set is performed using the bilinear or the bicubic interpolation as detailed in §A.1. Near the domain boundaries, the bilinear interpolation is preferred since the stencil is not large enough to perform a bicubic interpolation. The bicubic interpolation could also generate a non-physical high curvature of the interface. As in Wang et al. (2012a), we use the bicubic interpolation only if:

$$-\beta < (\phi_{bicubic} - \phi_{bilinear}) < \beta$$

with $\beta = \Delta x/20$. The velocity is also interpolated using the bilinear interpolation.

1.3 Level-set reinitialization

After having advected the level-set, the signed-distance property is disturbed. In order to maintain $\phi(\vec{x}, t)$ close to a signed-distance function, the following reinitialization equation is solved:

$$\frac{\partial \phi}{\partial \tau} + S(\phi_0)(|\nabla \phi| - 1) = 0 \tag{1.40}$$

where τ is a pseudo-time, $\phi_0 = \phi(\tau = 0)$ is the initial level-set field before reinitialization and S is the sign function. The exact signed-distance function is recovered when this equation is solved up to steady state, but in practice it is sufficient to stop the computation at some pseudo-time τ_{max} which is of the order of $n\Delta x$, where n depends on the stencils used. Mukherjee and Kandlikar (2007) for example considered 10 iterations for the reinitialization equation with the third-order TVD Runge-Kutta method. The fictitious time step suggested by the same authors is $\Delta\tau = \Delta x/2$. They reinitialized the level-set after each advection step.

In order to avoid numerical difficulties, a smoothed version of the sign function is generally used. We use the definition proposed by Sussman et al. (1994):

$$S(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + \delta^2}} \tag{1.41}$$

where δ is the thickness of smoothing region, generally taken as the grid size $\delta = \Delta x$. Other formulations of the sign function exist, in particular we may integrate in this function the so-called sub-cell fix proposed by Russo and Smereka (2000) (see §1.3.1), which reduces the displacement of the interface by the reinitialization procedure and thereby improves mass conservation.

In this work, the third-order TVD Runge-Kutta scheme is also employed for reinitialization, except for the case where the alternative semi-Lagrangian advection is used. For this latter, a semi-Lagrangian reinitialization without volume enforcement is used according to Wang et al. (2012a). For the Eulerian approach, let ϕ^n be the solution of Eq. (1.40) after n pseudo-time steps, then the solution at the next step is computed from:

$$\phi^{(1)} = \phi^n + \Delta\tau S(\phi_0)(1 - |\nabla \phi^n|) \tag{1.42}$$

$$\phi^{(2)} = \frac{3}{4}\phi^n + \frac{1}{4}\phi^{(1)} + \frac{1}{4}\Delta\tau S(\phi_0)(1 - |\nabla \phi^{(1)}|) \tag{1.43}$$

$$\phi^{n+1} = \frac{1}{3}\phi^n + \frac{2}{3}\phi^{(2)} + \frac{2}{3}\Delta\tau S(\phi_0)(1 - |\nabla \phi^{(3)}|) \tag{1.44}$$

where $\phi^{(1)}$ and $\phi^{(2)}$ are intermediate values and ϕ^{n+1} is the solution at time τ^{n+1}.

The term $|\nabla\phi|$ in Eq. (1.40) is determined from $|\nabla\phi| = (\phi_x^2 + \phi_y^2)^{1/2}$ where the values of ϕ_x^2

and ϕ_y^2 are computed using the Godunov method (Osher and Fedkiw, 2003), resulting in:

$$\phi_x^2 = \max\left(\max(\phi_x^-, 0)^2, \min(\phi_x^+, 0)^2\right) \quad \text{for} \quad S(\phi_0) > 0 \tag{1.45}$$

$$\phi_x^2 = \max\left(\min(\phi_x^-, 0)^2, \max(\phi_x^+, 0)^2\right) \quad \text{for} \quad S(\phi_0) < 0 \tag{1.46}$$

where ϕ_x^- and ϕ_x^+ are approximations of the derivative $\partial\phi/\partial x$ based on left- and right-hand sided stencils. Similar expressions are used for ϕ_y^2. The spatial discretization, i.e.: the determination of the spatial derivatives, is performed usually using the Hamilton-Jacobi schemes. We have tested the following schemes:

- 1st-order Upwind.
- 2nd-, 3rd- and 4th-order HJ-ENO.
- 3rd- and 5th-order HJ-WENO.

One of the main disadvantages of the reinitialization procedure is that it generally causes a slight displacement of the interface and thereby a change of the mass of the fluids on either side. Several fixes have been proposed in literature that try to limit the movement of the interface.

The sub-cell fix method for example by Russo and Smereka (2000) uses a first-order discretization for cells near the interface. Later, the volume constraint method by Sussman and Fatemi (1999) adds a supplementary term to the reinitialization equation in order to correct the mass errors during the reinitialization step. Solving this latter is a two step procedure: first, an intermediate level-set is computed, then the mass correction is applied. A global mass correction scheme was proposed by Yap et al. (2006), then enhanced by Zhang et al. (2010), is in principle similar to the volume constraint method. In this context, other methods could be found for example in Chang et al. (1996) and Zhang (1998) in which authors proposed to add a supplementary term for the mass correction which takes into account the interface curvature.

In the current work, we have tested only the sub-cell fix and the volume constraint methods.

1.3.1 Sub-cell fix

The reinitialization should make the level-set as a signed-distance function. However, it moves slightly the zero level which causes non-physical change of the reference fluid mass. To overcome this drawback, Russo and Smereka (2000) proposed a solution called the sub-cell fix. The remedy to the interface displacement is the use of the sign function for the reinitialization on the whole domain, then for the grid cells near the interface a truly upwind discretization is employed near the interface based on the distance to the interface. A modification of the sub-cell fix reinitialization is proposed in Sun et al. (2010), but is not used in this work. Instead, we have used the original method by Russo and Smereka (2000).

First of all, we should locate cells cut by the interface. More specifically, we say that the interface passes through a cell $(i,j) \in \sum_{\Delta x}$, if $\phi_{i,j}^0\phi_{i-1,j}^0 < 0$ or $\phi_{i,j}^0\phi_{i+1,j}^0 < 0$ or $\phi_{i,j}^0\phi_{i,j-1}^0 < 0$ or $\phi_{i,j}^0\phi_{i,j+1}^0 < 0$. Then, the minimum distance separating the interface from the grid node (i,j), at these interface neighboring grid cells is computed by:

$$D_{i,j} = \frac{\phi_{i,j}^0}{|\nabla\phi_{i,j}^0|} \qquad \text{with } |\nabla\phi_{i,j}^0| = \sqrt{\phi_x^2 + \phi_y^2}$$

where the gradient norm is evaluated by

$$\phi_x = \max\left\{\frac{|\phi_{i+1,j}^0 - \phi_{i-1,j}^0|}{2\Delta x}, \frac{|\phi_{i+1,j}^0 - \phi_{i,j}^0|}{\Delta x}, \frac{|\phi_{i,j}^0 - \phi_{i-1,j}^0|}{\Delta x}\right\}$$

and

$$\phi_y = \max\left\{\frac{|\phi^0_{i,j+1} - \phi^0_{i,j-1}|}{2\Delta y}, \frac{|\phi^0_{i,j+1} - \phi^0_{i,j}|}{\Delta y}, \frac{|\phi^0_{i,j} - \phi^0_{i,j-1}|}{\Delta y}\right\}$$

The corresponding smoothed signed-distance in this region is a function of the distance $D_{i,j}$:

$$S(\phi_0) = \frac{D_{i,j}}{\Delta x}$$

Let us consider the first sub-step of RK3 time discretization of Eq. (1.42) which is the Euler discretization. The spatial discretization of the reinitialization Eq. (1.40) should be modified on the region near the interface, and it takes the form:

$$\phi^{n+1} = \phi^n - \frac{\Delta\tau}{\Delta x}\left[\text{sign}(\phi_0)|\phi^n| - D_{i,j}\right] \tag{1.47}$$

One should note that $\text{sign}(S(\phi_0)) = \text{sign}(\phi_0)$.

Finally, we resume the sub-cell fix reinitialization in these two steps:

1. Compute the smoothed signed-distance function

$$S(\phi_0) = \begin{cases} \dfrac{D_{i,j}}{\Delta x} & \text{if } \ (i,j) \in \sum_{\Delta x} \\ \dfrac{\phi_0}{\sqrt{\phi_0^2 + \Delta x^2}} & \text{otherwise} \end{cases} \tag{1.48}$$

2. Advance the reinitialization equation with a pseudo time step $\Delta\tau$

$$\phi^{n+1} = \begin{cases} \phi^n - \Delta\tau\left(\text{sign}(\phi_0)\dfrac{|\phi^n|}{\Delta x} - S(\phi_0)\right) & \text{if } \ (i,j) \in \sum_{\Delta x} \\ \phi^n - \Delta\tau S(\phi_0)(|\nabla\phi^n| - 1) & \text{otherwise} \end{cases} \tag{1.49}$$

   In the above reinitialization equation, the spatial discretization is only first-order near the interface and we are free to use higher-order schemes for grids not on the interface.

If time advancement is done by a multi-step scheme, sub-cell fix should be applied to each sub-time-step.

Based on the fact that reinitialization at each physical time step introduces artificial displacement of the interface, sub-cell fix reinitialization is not performed after each advection but after many advection steps (Russo and Smereka, 2000). This would introduce less errors. Instead of solving the reinitialization equation for a fictitious time τ_{max} after each advection, we can solve it on a fictitious time of $cst \times \tau_{max}$ where $cst > 1$ after k advection steps.

1.3.2 Volume constraint

A different approach to improve mass conservation during the reinitialization procedure is proposed by Sussman and Fatemi (1999). This modification is mainly based on adding a new term to the reinitialization Eq. (1.40). This term is a mass correction, and it preserves the zero level-set whilst redistancing. The so-called reinitialization with constraint could be expressed by the equation:

$$\frac{\partial\phi}{\partial\tau} + \text{sign}\left(\phi_0\right)\left(|\nabla\phi| - 1\right) - \lambda f(\phi) = 0 \tag{1.50}$$

where $f(\phi)$ is chosen as : $f(\phi) \equiv H'(\phi_0)|\nabla\phi_0|$. The derivative of the Heaviside function is used in the added term to correct the zero level-set without disturbing the distance function property

away from the interface. λ is a constant. The Heaviside function is a step function taking either zero or unity. In this method, we rather use the smeared-out Heaviside on a narrow band of thickness Δx on each side of the interface, denoted $H_{\Delta x}(\phi)$ and defined as:

$$H_{\Delta x}(\phi) = \begin{cases} 1 & \text{if } \phi > \Delta x \\ 0 & \text{if } \phi < -\Delta x \\ \frac{1}{2} + \frac{\phi}{2\Delta x} + \frac{1}{2\pi}\sin\left(\frac{\pi\phi}{\Delta x}\right) & \text{otherwise} \end{cases} \tag{1.51}$$

The resulting smooth sign function is then:

$$\text{sign}_{\Delta x}(\phi) = 2(H_{\Delta x}(\phi) - 1/2)$$

The main idea of this method is that when discretized, the area of each computational cell should remain unchanged. The area is the integral of the Heaviside function in space:

$$A_{i,j} = \int_{\Omega_{i,j}} H_{\Delta x}(\phi) d\vec{x}$$

where $\Omega_{i,j}$ includes all the points (x, y) delimited by $x_{i-1/2} < x < x_{i+1/2}$ and $y_{i-1/2} < y < y_{i+1/2}$. The constraint that $A_{i,j}$ remains unchanged is equivalent to:

$$\int_{\Omega_{i,j}} H'_{\Delta x}(\phi_0)\frac{\partial\phi}{\partial\tau} d\vec{x} = 0 \tag{1.52}$$

So to apply the local constraint to the reinitialization equation, one should first calculate the intermediate reinitialized level-set field at the time $n+1$, denoted as $\tilde{\phi}^{n+1}$, from the field at the step n, i.e.: ϕ^n. This is done by solving the Eq. (1.50) while ignoring the added term, and this comes to solving the standard reinitialization equation given by Eq. (1.40).
Now, let us find an expression of λ. By replacing $\partial\phi/\partial\tau$ with its expression given by Eq. (1.50) in Eq. (1.52), and by the use of the intermediate level-set $\tilde{\phi}^{n+1}$ definition we get :

$$\int_{\Omega_{i,j}} H'_{\Delta x}(\phi_0)\left[\frac{\tilde{\phi}^{n+1} - \phi_0}{\Delta\tau} + \lambda_{i,j} f(\phi_0)\right] d\vec{x} = 0$$

and thus, it follows that:

$$\lambda_{i,j} = \frac{-\int_{\Omega_{i,j}} H'(\phi_0)\left(\frac{\tilde{\phi}^{n+1}-\phi_0}{\Delta\tau}\right) d\vec{x}}{\int_{\Omega_{i,j}} H'^2(\phi_0)|\nabla\phi_0| d\vec{x}} \tag{1.53}$$

These integrals are computed using a nine-point quadrature in 2D, i.e.: we use the information of the level-set on the grid point of interest and we take into account the information of the neighboring grids. For a grid point not on the edges of the discretized domain, the volume integral is computed by means of:

$$\int_{\Omega_{i,j}} g\, d\vec{x} \approx \frac{h^2}{24}\left(16 g_{i,j} + \sum_{m,n=-1;(m,n)\neq(0,0)}^{1} g_{i+m,j+n}\right)\Delta x\Delta y$$

If we want to implement a 3D or an axisymmetric version of this constraint, one should first adapt the volume integral given by the previous quadrature.

Since we have already evaluated $\tilde{\phi}^{n+1}$, the gradient projection step during which the constraint is applied is simply

$$\phi^{n+1} = \tilde{\phi}^{n+1} + \Delta\tau\lambda H'_{\Delta x}(\phi_0)|\nabla\phi_0| \tag{1.54}$$

When the third-order Runge-Kutta scheme is used for temporal discretization of the reinitialization equation, the gradient projection step is performed only once after the last sub-time step. In the sub-cell fix approach, however, the correction is applied after each sub-time step.

1.4 Local level-set method

The level-set advection and reinitialization equations are expensive from a computational effort point of view when solved on the whole computational domain. Peng et al. (1999) have developed a method to solve these equations in a narrow band around the interface.

First, let β and γ be two constants comparable to Δx with $\gamma > \beta > 0$. The largest constant, γ is used to define two tubes (narrow bands) N and T as follows:

$$\begin{aligned} T &= \{(x,y) : |\phi(x,y)| \leq \gamma\} \\ N &= \{(x,y) : |\phi(x \pm \delta x, y \pm \delta y)| \leq \gamma\} \text{ for } \delta x < \Delta x \text{ and } \delta y < \Delta y \end{aligned} \tag{1.55}$$

Assuming that the level-set function is approximately a signed-distance function, the tube T contains all cells whose centers are within a distance of γ to the interface. The tube N contains the tube T, but also the neighboring cells.

Then, let us define the cut-off function (Peng et al., 1999):

$$c(\phi) = \begin{cases} 1 & \text{if } |\phi| \leq \beta \\ (2|\phi| + \gamma - 3\beta)\dfrac{(|\phi| - \gamma)^2}{(\gamma - \beta)^3} & \text{if } \beta < |\phi| \leq \gamma \\ 0 & \text{if } |\phi| > \gamma \end{cases} \tag{1.56}$$

This cut-off function is equal to 1 in the narrow band of distance β on each side of the interface, and then smoothly decreases to zero towards the outer boundary of the tube T. This function is integrated into the advection equation (1.1) which is rewritten as

$$\frac{\partial\phi}{\partial t} + c(\phi)\vec{u}\cdot\nabla\phi = 0 \tag{1.57}$$

The objective is to smoothly damp out the advection near the edges of the tube T to avoid generation of disturbances at this location. It is clear that the advection equation now only needs to be solved in the tube T, since the advection velocity is zero outside. Note that the exact advection equation is solved for all points with $|\phi| \leq \beta$.

The reinitialization equation is solved in the tube N. Since this tube is slightly larger than tube T, the disturbances near the outer boundary of the latter are reduced to a minimum. Once the reinitialization step is completed, the values of the signed-distance function in the tube T are conserved, while they are set to a constant value outside the tube. In other words, the following correction is applied:

$$\phi(x,y) = \begin{cases} -\gamma & \text{if } \phi(x,y) < -\gamma \\ \phi(x,y) & \text{if } |\phi(x,y)| \leq \gamma \\ \gamma & \text{if } \phi(x,y) > \gamma \end{cases} \tag{1.58}$$

In order to preserve the accuracy of the numerical schemes near the interface, the values of β

and γ need to be defined as a function of the stencil size used for the approximation of the spatial derivatives of Eq. (1.57). Peng et al. (1999) recommanded $\beta = 3\Delta x$ and $\gamma = 6\Delta x$ for the third-order ENO and the fifth-order WENO schemes. In our computation, these two constants are set to $\beta = 6\Delta x$ and $\gamma = 9\Delta x$ to allow the use of high-order schemes.

1.5 Testing the level-set method

As shown in the previous sections, many variants of the level-set method can be built depending on the numerical schemes employed for the spatial discretization of the advection and the reinitialization equations. Different reinitialization methods can also be obtained by selecting one of the already described methods to reduce the interface displacement while reinitializing. The general algorithm of the level-set method is given by Algorithm 1.1, where Nreinit is the number of advection steps to perform before reinitializing the level-set.

Algorithm 1.1 Level-set method

Input: ϕ^n, $\vec{u}^{n+1/2}$, Δt, $\tau_{\max}$, $\Delta\tau$, Nreinit
  $\phi^{n+1} \leftarrow$ Advect level-set Eq. (1.1) using 3rd-order TVD Runge-Kutta (ϕ^n, $u^{n+1/2}$, Δt)
  if $(\mathrm{mod}(n, \mathrm{Nreinit}) = 0)$ **then**
    $\phi^{n+1} \leftarrow$ Reinitialize level-set (ϕ^{n+1}, $\tau_{\max}$, $\Delta\tau$)
  end if
  $\vec{u}^{n+3/2} \leftarrow$ Advance flow equations (ϕ^{n+1}, $\vec{u}^{n+1/2}$, Δt)
  return ϕ^{n+1}, $\vec{u}^{n+3/2}$

This section is presented as follows: we start first by providing definitions for the interface error, then we describe some reference test cases in order to compare and validate the methods. The idea then is to first validate our implementation of the level-set method with the results provided by Enright et al. (2002). This level-set is given by Algorithm 1.1 with these specifications:

Level-set: The reinitialization is performed using Eq. (1.40) with the definition of the sign function given by Eq. (1.41). This equation is solved on a band of $\tau_{\max} = 10\Delta x$ around the interface, with a fictitious time step $\Delta\tau = \Delta x$. The level-set is reinitialized at the end of each advection step (Nreinit = 1). The fifth-order HJ-WENO scheme is used for the spatial discretization of the advection and the reinitialization equations.

Next, we vary the scheme for the advection of the level-set, and we compare it to the level-set without reinitialization in order to show the benefits/drawbacks of the reinitialization procedure.

Then, we test the level-set with the sub-cell fix reinitialization. For this we vary the advection scheme, then the CFL number in order to investigate its stability. Some of the suited schemes for the advection of the level-set with the sub-cell fix will be identified. For these identified schemes, we investigate the influence of the reinitialization scheme on the behavior of the method. The sub-cell fix is given by Algorithm 1.1 with the corresponding specifications:

Sub-cell fix : The reinitialization is performed using the sub-cell fix method Eq.(1.49) with the definition of the sign function in Eq. (1.48). The fictitious time is set to $\tau_{max} = 10\Delta x$ with a time step $\Delta\tau = \Delta x/2$. The reinitialization is performed after the first level-set advection step, then after each Nreinit = 10 advection steps.

The study of the level-set with volume constraint reinitialization is performed in the same manner as for the sub-cell fix. The level-set with volume constraint reinitialization is also given by Algorithm 1.1, with:

Volume constraint: The reinitialization is performed using Eq. (1.40) with the definition of the sign function in Eq. (1.41). The volume constraint Eq. (1.54) is then solved to enforce the level-set mass conservation. The fictitious time is $\tau_{max} = 2\Delta x$ with $\Delta\tau = \Delta x/2$. The reinitialization is performed after each advection step (Nreinit = 1).

For the temporal discretization of the advection and the reinitialization equations, we have used the third-order TVD Runge-Kutta scheme. Concerning the spatial discretization, all the different schemes detailed in the previous sections were tested in order to identify the advantages and disadvantages of each scheme. The schemes and abbreviations used to advect the level-set equation are:

- 1st-order UPWIND (UPWIND1) §1.2.2.1
- 2nd-order Hamilton-Jacobi ENO (HJ-ENO2) §1.2.2.2
- 3rd- and 5th-order Hamilton-Jacobi WENO (HJ-WENOk, where k is the order) §1.2.2.3
- 3rd-, 5th-, 7th-, 9th- and 11th-order HOUC (HOUCk, where k is the order) §1.2.2.4
- 2nd-, 3rd- and 4th- order conservative ENO (ENOk-(Jia, Adv and Est), where k is the order) §1.2.2.5
- 5th-order conservative WENO- (WENO5-(Jia, Adv and Est)) §1.2.2.6
- 5th-order conservative HOUC- (HOUC5-(Jia, Adv and Est)) §1.2.2.7

and for the reinitialization equation:

- 1st-order UPWIND (UPWIND1)
- 2nd-, 3rd- and 4th-order Hamilton-Jacobi ENO (HJ-ENO2, HJ-ENO3, HJ-ENO4)
- 3rd- and 5th-order Hamilton-Jacobi WENO (HJ-WENO3, HJ-WENO5)

In addition to these schemes, the semi-Lagrangian advection (Wang et al., 2012a) detailed on §1.2.3 is also tested, always in combination with the semi-Lagrangian reinitialization.

1.5.1 Definition of interface errors

The accuracy of the interface location and the contained area of the reference fluid are two important comparison criteria. The contained area (directly related to the total mass) is defined by the integral of the Heaviside over the whole domain. The mass loss/gain error is then given by:

$$\begin{aligned} E_M &= \frac{\text{mass} - \text{Exact mass}}{\text{Exact mass}} \\ &= \frac{\int_\Omega H(\phi_C)d\Omega - \text{Exact mass}}{\text{Exact mass}} \end{aligned} \tag{1.59}$$

The exact mass may be computed either analytically from the initial condition or numerically from the initial level-set field, while the mass at a given time of the simulation is the integral of

the Heaviside over all the domain. The Heaviside integral is estimated using the bilinear or the bicubic interpolation as described in Appendix A.1.

The positioning error E_1 is expressed as:

$$E_1 = \frac{\int_\Omega |H(\phi_E) - H(\phi_C)| \, d\Omega}{\int_\Omega H(\phi_E) d\Omega} \tag{1.60}$$

where subscripts E and C refer to exact and computed solutions. Note that the exact solution is defined either by the exact level-set field if this latter is available, or by a reference solution. The positioning error E_1 is computed as described in Appendix A.1. Note that E_1 and E_M are computed using the bicubic interpolation for mixed cells which are subdivided into m^2 sub-cells with $m = 1000$, except otherwise specified.

The convergence rate based on the positioning error, E_1, is defined as:

$$\mathcal{O} = \frac{\log\left(E_1^{(r)}/E_1^{(c)}\right)}{\log\left(\Delta x^{(r)}/\Delta x^{(c)}\right)} \tag{1.61}$$

where superscripts (r) and (c) mean that the quantity is computed for refined and coarse grid, respectively.

We define also the mean absolute deviation of the level-set from a signed-distance function as:

$$E_{|\nabla\phi|} = \frac{\sum_{k=1}^{N_i} ||\nabla\phi|_k - 1|}{N_i} \quad \text{with} \quad |\nabla\phi| = \sqrt{\phi_x^2 + \phi_y^2} \tag{1.62}$$

where N_i is the number of cells cut by the interface and the $x-$ and $y-$derivatives are computed using the 2nd-order central differences.

1.5.2 Zalesak disk problem

Zalesak's rotating disk (Zalesak, 1979) test evaluates the ability of interface methods in transporting rigid objects. It is different from the circle translation/rotation test case (Pilliod and Puckett, 2004) due to the presence of sharp angles; each interface capturing method will handle differently these discontinuities. The Zalesak disk simulation is therefore a good measure of the numerical diffusivity of interface methods.

The slotted disk (Fig. 1.4) is defined by a radius $R = 0.15$, a slot width $w = 0.05$ and of a distance $h = 0.05$. This disk is placed in a 1×1 domain and initially centered at $(0.5, 0.25)$. The imposed velocity field is that of a solid rotation around the origin $(0.5, 0.5)$:

$$\begin{cases} u = \dfrac{(y - 0.5)\,\pi}{3.14} \\ v = \dfrac{(0.5 - x)\,\pi}{3.14} \end{cases}$$

The CFL value is based on the maximum velocity given by:

$$\text{CFL} = \max(|u|, |v|)\,\frac{\Delta t}{\Delta x} = \frac{\pi \Delta t}{6.28 \Delta x}$$

A full rotation of the disk is accomplished after 6.28 time units and the exact position of the interface is known at any time by rotating the axis since it undergoes a solid rotation.

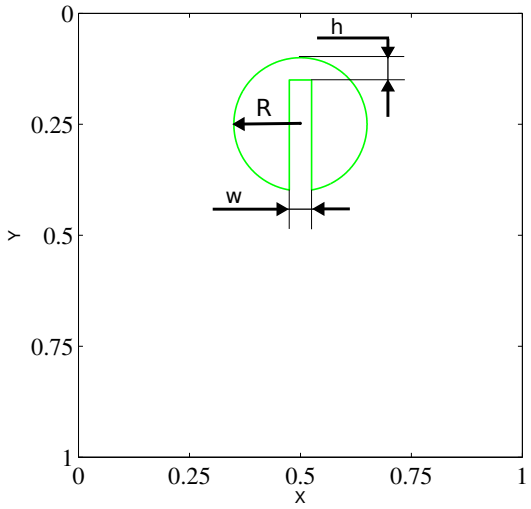

Figure 1.4: Zalesak disk initial configuration.

1.5.3 Shearing vortex problem

While the Zalesak's disk problem is a good indicator of diffusion errors in an interface capturing method, it does not test the ability of an Eulerian scheme to accurately resolve thin filaments which can occur in stretching and tearing flows (Bell et al., 1989). In this shearing vortex

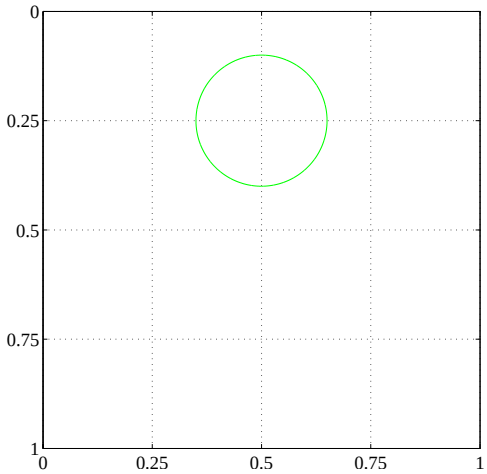

Figure 1.5: Single vortex initial configuration.

problem, the initial configuration is a circle of radius 0.15 centered at $(0.5, 0.25)$ in a 1×1 domain (see Fig. 1.5). The imposed velocity field in the case of the single vortex is derived from the stream function

$$\Psi = \frac{1}{\pi} \sin^2(\pi x) \sin^2(\pi y) \tag{1.63}$$

given by

$$\begin{cases} u = -\dfrac{\partial \Psi}{\partial y} \\ v = \dfrac{\partial \Psi}{\partial x} \end{cases}$$

The use of the analytical formula of the velocity in the discrete domain leads to a non zero discrete divergence. We rather use the central differences to numerically derive the velocity field as follows:

$$\begin{cases} u_{i+1/2,j} = -\dfrac{\Psi_{i+1/2,j+1/2} - \Psi_{i+1/2,j-1/2}}{\Delta y} \\ v_{i,j+1/2} = \dfrac{\Psi_{i+1/2,j+1/2} - \Psi_{i-1/2,j+1/2}}{\Delta x} \end{cases}$$

from which it is easily shown that the discrete velocity divergence, $(u_{i+1/2,j} - u_{i-1/2,j})/\Delta x + (v_{i,j+1/2} - v_{i,j-1/2})/\Delta y$ is equally zero.

Computations are performed up to a time $t = T/2 = 3$ at which point the interface reaches maximum stretching; then, the velocity sign is reversed which causes the return to the initial position which would be a circle at time $t = T = 6$ in the ideal case.

For the computation of the mass error E_M and the positioning error E_1 at the beginning and the end of the simulation, the exact interface is known which is a circle. However at an intermediate time step, no analytic formula gives the exact interface of the vortex. In the current work, we use an accurate Lagrangian method to estimate the interface position at a given time as detailed in Appendix A.2.

1.5.4 Time-reversed single vortex problem

The time-reversed single vortex is similar to the shearing vortex problem. The initial interface form and configuration are identical. What changes is the stream function from which the velocity field is derived, and the period T during which the interface is transported. The stream function given by the Eq. (1.63) is multiplied by $\cos(\pi t/T)$, which gives:

$$\Psi = \frac{1}{\pi} \sin^2(\pi x) \sin^2(\pi y) \cos\left(\frac{\pi t}{T}\right) \tag{1.64}$$

In order to conserve mass, the velocity field should be divergence free, hence the discrete velocity fields are derived in the same manner as explained for the shearing vortex problem. The flow inversion at $t = T/2$ is directly ensured by the cosine term. Due to this cosine term, the velocity magnitude is also decreasing in time until $\cos(\pi t/T) = 0$, then it increases again until $\cos(\pi t/T) = 1$. The period T is set to 2, which make this test less severe since the interface undergoes less stretching. The exact position of the interface is determined in the same manner as for the shearing vortex problem, using a Lagrangian method.

1.6 Results and discussions

First, we start by comparing our level-set implementation with results found in literature, more specifically with those of Enright et al. (2002) for the Zalesak disk (Fig. 1.4). In this work, the level-set advection Eq. (1.1) and reinitialization Eq. (1.40) are discretized in time using TVD-RK3 and in space using HJ-WENO5 scheme. For advection, the CFL is 0.5. The only difference between the present implementation (with the same advection and reinitialization schemes) and the one presented in Enright et al. (2002) is that in our case the level-set reinitialization Eq. (1.40) is solved on the entire domain, and in the work of Enright et al. (2002) on the narrow band with the use of a fast marching method (Chopp, 2001; Sethian, 1999) on the rest of the domain.

The mass error E_M and positioning error E_1 for different grid sizes are summarized on Table 1.1. In the work of Enright et al. (2002), these errors were computed using the bilinear interpolation with 1000^2 sub-cells. We have used the same method for this comparison.

Table 1.1: Zalesak disk: comparison of level-set method with the results of Enright et al. (2002). The HJ-WENO5 scheme is used for both advection and reinitialization with Nreinit = 1 and CFL = 0.5.

	Enright et al. (2002)			Level-set: our program		
Grid	$E_M(\%)$	$E_1(\%)$	$\mathcal{O}$	$E_M(\%)$	$E_1(\%)$	$\mathcal{O}$
After one revolution						
50^2	-100.00	100.00	–	-46.12	60.97	–
100^2	$+5.30$	15.14	2.7	$+5.03$	13.68	2.2
200^2	-0.54	1.98	2.9	-0.43	1.64	3.1
After two revolutions						
50^2	-100.00	100.00	–	-59.35	69.96	–
100^2	$+9.0$	22.08	2.2	$+9.39$	20.45	1.8
200^2	-0.82	2.73	3.0	-0.63	2.07	3.3

We remark that, although we observe the same order of magnitude and tendency for the errors, the current implementation provides slightly more accurate results, especially for the grid 50^2 where the interface vanishes in the implementation of Enright et al. (2002). These differences are most probably due to the reinitialization method which is slightly different.

In the next subsection, we vary the scheme for the advection of the level-set and we compare the resulting interface from each advection with its corresponding when we do not perform reinitialization.

1.6.1 Effect of advection scheme and reinitialization

We now investigate the effect of the advection scheme and compare at the same time the level-set with and without reinitialization. The results with reinitialization are obtained with Nreinit = 1, $\tau_{\max} = 10\Delta x$ and $\Delta\tau = \Delta x$. No sub-cell fix or volume constraint is applied. The HJ-WENO5 scheme is used for the spatial discretization of the reinitialization Eq. (1.40) except for the semi-Lagrangian level-set where the reinitialization is of the semi-Lagrangian type. The CLF number for advection is set to 0.5.

Zalesak disk

In the Zalesak disk test case, the velocity u depends only on the y-coordinate ($u_{k,j} = u_{l,j}$, $\forall (k,l,j) \in [1,n]^3$). Similarly, the velocity in the y-direction depends only on the x-coordinate. Therefore, the three formulations of the advection scheme (see §1.2.2.5) are strictly equivalent:

$$\frac{u_{i+\frac{1}{2},j}\widehat{\phi}_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}\widehat{\phi}_{i-\frac{1}{2},j}}{\Delta x} = u_{i,j}\frac{\widehat{\phi}_{i+\frac{1}{2},j} - \widehat{\phi}_{i-\frac{1}{2},j}}{\Delta x} = \frac{\widehat{\phi u}_{i+\frac{1}{2},j} - \widehat{\phi u}_{i-\frac{1}{2},j}}{\Delta x}$$

Hence, there is no need to specify which formulation is used in this test case.

The resulting interface is presented in Fig. 1.6 for both the level-set with (in blue) and without reinitialization (in red), for the Zalesak disk after a complete revolution (at time $t = 6.28$). We observe that reinitialization prevents the interface from vanishing when the advection of the level-set is ensured by the first-order upwind scheme, and it does not smooth the entire slab for the HJ-ENO2, ENO2 and the HJ-WENO3 schemes. However, for higher-order schemes, the reinitialization causes a deterioration of the interface representation with respect to the method without reinitialization. This latter is due to:

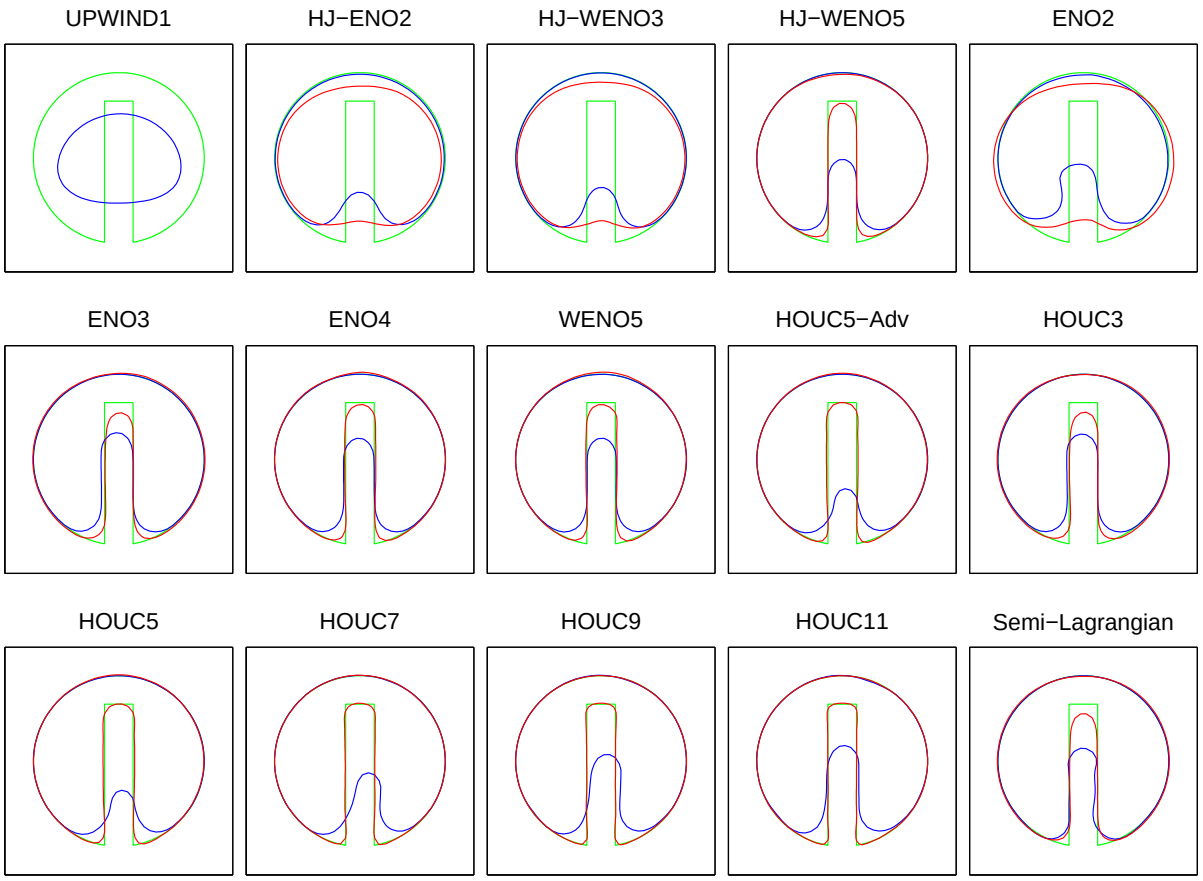

Figure 1.6: Level-set with and without reinitialization: Zalesak disk after a full rotation for CFL = 0.5 and a 100^2 grid. Results obtained with different advection schemes: Green: Exact interface ; Red: without reinitialization ; Blue: HJ-WENO5 reinitialization with $\Delta\tau = \Delta x$, $\tau_{\max} = 10\Delta x$ and Nreinit = 1.

- to relatively large fictitious time step ($\Delta\tau = \Delta x$).
- to the use of the fifth-order HJ-WENO scheme in the reinitialization process.
- to the relatively long reinitialization time ($\tau_{\max} = 10\Delta x$).
- to the fact that the level-set is reinitialized after each advection.

The mass and positioning errors corresponding to each advection scheme are reported on Table 1.2 for the level-set with and without reinitialization. In this test case, the interface positioning error, $E_1(\%)$, is probably the most relevant. The mass error $E_M(\%)$ sometimes show very small values when positive and negative error contributions compensate (for the HJ-ENO2 and WENO5 schemes for example). When considering the error $E_1(\%)$, the main contribution for the cases without reinitialization is due the rounding of the interface near the sharp corners. The use of high-order schemes, in particular the HOUCk scheme with $k \geq 7$, seems to reduce this error. However, high-order schemes use larger stencils and are much more sensitive to small perturbations in the level-set field. The effect of reinitialization is therefore greater, in particular for the HOUC schemes with fixed stencil. We can almost imagine that the level-set without reinitialization is better than the level-set with reinitialization. However, without reinitialization the deviation of the level-set from the signed-distance function, $E_{|\nabla\phi|}$, is much larger.

Table 1.2: Comparison of level-set errors with and without reinitialization for the Zalesak disk after a full rotation on a 100^2 grid. Advection: CFL = 0.5. Reinitialization: HJ-WENO5 with $\Delta\tau = \Delta x$, $\tau_{\max} = 10\Delta x$ and Nreinit = 1.

Advection scheme	No reinitialization			Reinitialization		
	$E_M(\%)$	$E_1(\%)$	$E_{\lvert\nabla\phi\rvert}$	$E_M(\%)$	$E_1(\%)$	$E_{\lvert\nabla\phi\rvert}$
UPWIND1	−100.00	100.00	–	−53.47	80.21	2.26×10^{-3}
HJ-ENO2	−0.28	36.73	0.404	4.99	23.26	3.20×10^{-3}
HJ-WENO3	6.66	29.84	0.338	7.29	19.69	3.23×10^{-3}
HJ-WENO5	−0.87	2.95	0.133	5.12	13.70	3.34×10^{-3}
ENO2	14.71	31.89	0.385	−2.04	22.21	3.43×10^{-3}
ENO3	3.59	6.53	0.172	−0.30	10.52	4.86×10^{-3}
ENO4	0.22	4.04	0.106	1.13	10.71	3.71×10^{-3}
WENO5	−0.01	4.13	0.105	1.58	10.44	3.83×10^{-3}
HOUC5-Adv	−1.23	3.27	0.078	10.00	18.28	5.20×10^{-3}
HOUC3	3.37	5.97	0.160	0.37	10.13	4.09×10^{-3}
HOUC5	−1.24	3.26	0.078	10.00	18.28	5.20×10^{-3}
HOUC7	−0.80	1.83	0.066	7.02	18.13	4.99×10^{-3}
HOUC9	−0.31	1.10	0.054	2.91	15.24	4.66×10^{-3}
HOUC11	−0.18	0.88	0.047	0.56	13.31	4.09×10^{-3}
Semi-Lagrangian	3.27	5.60	0.165	6.97	10.08	8.95×10^{-3}

Shearing vortex

The interface at times $t = T/2 = 3$ for the shearing vortex test case are plotted in Fig. 1.7 and the corresponding interface errors are summarized on Table 1.3. Note that since the u-velocity now depends on x, and the v-component on y, the various formulations of the advection equation (noted Adv, Est and Jia, see §1.2.2.5) do not lead to the same results, although the differences here are rather small.

Similar to the Zalesak disk, we remark that the interface vanishes before $t = T/2$ without reinitialization when the 1st or 2nd-order advection schemes are used. It vanishes also for the 3rd-order HJ-WENO or the semi-Lagrangian schemes. Reinitialization of the level-set clearly improves the results for these schemes.

Higher-order schemes perform well without reinitialization, in particular the HOUC schemes, but reinitialization now tends to decrease the accuracy of these methods, as it is in the case for the Zalesak disk. Reinitialization tends to thicken the filaments at $t = T/2$, thereby increases the positioning error and the total mass.

The HJ-schemes with and without reinitialization generally show poor accuracy in this test case, possible due to the slightly larger stencils which do not capture the thin filaments well.

In Fig. 1.8, the results for the interface at time $t = T = 6$ are shown, obtained after having inverted the flow direction in order to come back to the initial configuration. We obtain generally the same conclusions as for the results at $t = T/2 = 3$.

Table 1.3: Comparison of level-set errors with and without reinitialization for the shearing vortex test case on a 100^2 grid. Advection: CFL = 0.5. Reinitialization: HJ-WENO5 with $\Delta\tau = \Delta x$, $\tau_{\max} = 10\Delta x$ and Nreinit = 1.

Advection scheme	Time t = 3						Time t = 6					
	No reinitialization			Reinitialization			No reinitialization			Reinitialization		
	$\boldsymbol{E_M}(\%)$	$\boldsymbol{E_1}(\%)$	$\boldsymbol{E_{\lvert\nabla\phi\rvert}}$	$\boldsymbol{E_M}(\%)$	$\boldsymbol{E_1}(\%)$	$\boldsymbol{E_{\lvert\nabla\phi\rvert}}$	$\boldsymbol{E_M}(\%)$	$\boldsymbol{E_1}(\%)$	$\boldsymbol{E_{\lvert\nabla\phi\rvert}}$	$\boldsymbol{E_M}(\%)$	$\boldsymbol{E_1}(\%)$	$\boldsymbol{E_{\lvert\nabla\phi\rvert}}$
UPWIND1	−100.00	100.00	–	−100.00	100.00	–	−100.00	100.00	–	−100.00	100.00	–
HJ-ENO2	−100.00	100.00	–	−64.74	71.65	0.16	−100.00	100.00	–	−100.00	100.00	–
HJ-WENO3	−100.00	100.00	–	−60.67	66.96	0.14	−100.00	100.00	–	−100.00	100.00	–
HJ-WENO5	−46.25	48.84	3.78	−56.18	59.01	0.14	−63.52	63.52	0.44	−94.03	94.03	3.35×10^{-2}
ENO2-Adv	−100.00	100.00	–	57.86	89.42	2.12×10^{-2}	−100.00	100.00	–	53.31	95.42	6.20×10^{-3}
ENO2-Est	−100.00	100.00	–	59.78	90.30	2.06×10^{-2}	−100.00	100.00	–	55.67	96.58	5.98×10^{-3}
ENO2-Jia	−100.00	100.00	–	49.97	84.08	2.17×10^{-2}	−100.00	100.00	–	46.35	92.75	6.16×10^{-3}
ENO3-Adv	−34.80	39.39	3.93	33.68	65.13	2.89×10^{-2}	−67.67	67.67	0.55	31.77	72.14	6.30×10^{-3}
ENO3-Est	−36.73	40.12	3.85	34.34	65.35	2.86×10^{-2}	−67.39	67.39	0.55	32.43	72.40	6.26×10^{-3}
ENO3-Jia	−35.12	39.19	3.98	37.56	67.53	2.91×10^{-2}	−64.40	64.40	0.51	35.46	75.16	6.06×10^{-3}
ENO4-Adv	−14.89	26.02	5.14	21.69	56.84	3.74×10^{-2}	−24.67	28.06	0.29	17.92	63.78	6.49×10^{-3}
ENO4-Est	−15.20	26.43	5.22	22.20	56.94	3.63×10^{-2}	−23.26	25.82	0.29	18.09	63.72	6.54×10^{-3}
ENO4-Jia	−13.21	26.83	5.46	24.31	58.08	3.55×10^{-2}	−23.17	25.69	0.28	20.41	65.69	5.86×10^{-3}
WENO5-Adv	−7.58	23.89	5.63	14.53	50.79	4.23×10^{-2}	−14.38	21.93	0.24	7.97	57.25	6.01×10^{-3}
WENO5-Est	−8.50	27.49	5.55	15.07	50.89	4.12×10^{-2}	−13.67	20.69	0.24	8.57	57.32	5.96×10^{-3}
WENO5-Jia	−10.91	27.63	5.66	17.94	52.10	3.91×10^{-2}	−18.96	27.33	0.19	11.01	58.83	5.83×10^{-3}
HOUC5-Adv	0.06	23.60	6.38	−18.24	45.16	0.12	−2.08	18.70	0.12	−54.13	56.53	7.52×10^{-3}
HOUC5-Est	−2.19	22.91	6.28	−17.66	44.47	0.12	−2.08	18.54	0.12	−54.07	55.76	7.36×10^{-3}
HOUC5-Jia	0.29	22.74	6.40	−16.20	43.07	0.11	−2.08	18.12	0.13	−53.89	54.97	7.29×10^{-3}
HOUC3	−24.07	37.66	4.21	28.87	59.28	3.33×10^{-2}	−46.47	53.90	0.404	26.05	66.16	6.26×10^{-3}
HOUC5	0.06	23.60	6.38	−18.24	45.16	0.12	−2.08	18.70	0.12	−54.13	56.53	7.52×10^{-3}
HOUC7	1.35	22.07	6.88	−34.23	49.84	0.17	2.55	14.79	0.09	−95.71	95.71	4.34×10^{-2}
HOUC9	1.60	20.85	7.03	−58.67	62.77	0.17	3.75	11.21	0.08	−100.00	100.00	–
HOUC11	2.08	19.94	7.14	−67.56	73.38	0.14	3.83	8.84	0.07	−100.00	100.00	–
Semi-Lag	−100.00	100.00	–	53.35	78.64	2.40×10^{-2}	−100.00	100.00	–	43.17	77.91	7.09×10^{-3}

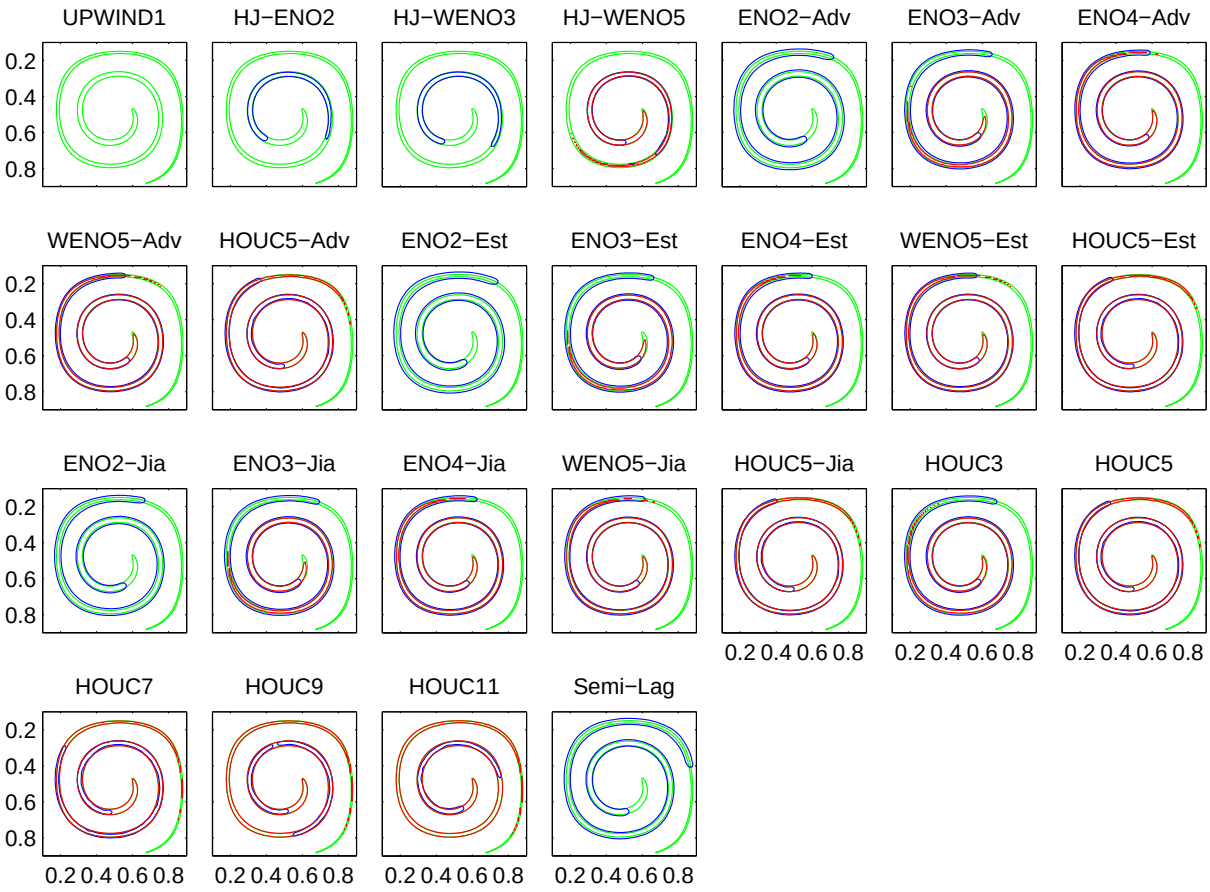

Figure 1.7: Level-set with and without reinitialization: shearing vortex case on a 100^2 grid and CFL $= 0.5$, interface at $t = T/2 = 3$ for different advection schemes. Green: Exact interface; Red: without reinitialization; Blue: HJ-WENO5 reinitialization with $\Delta\tau = \Delta x$, $\tau_{\max} = 10\Delta x$ and Nreinit $= 1$.

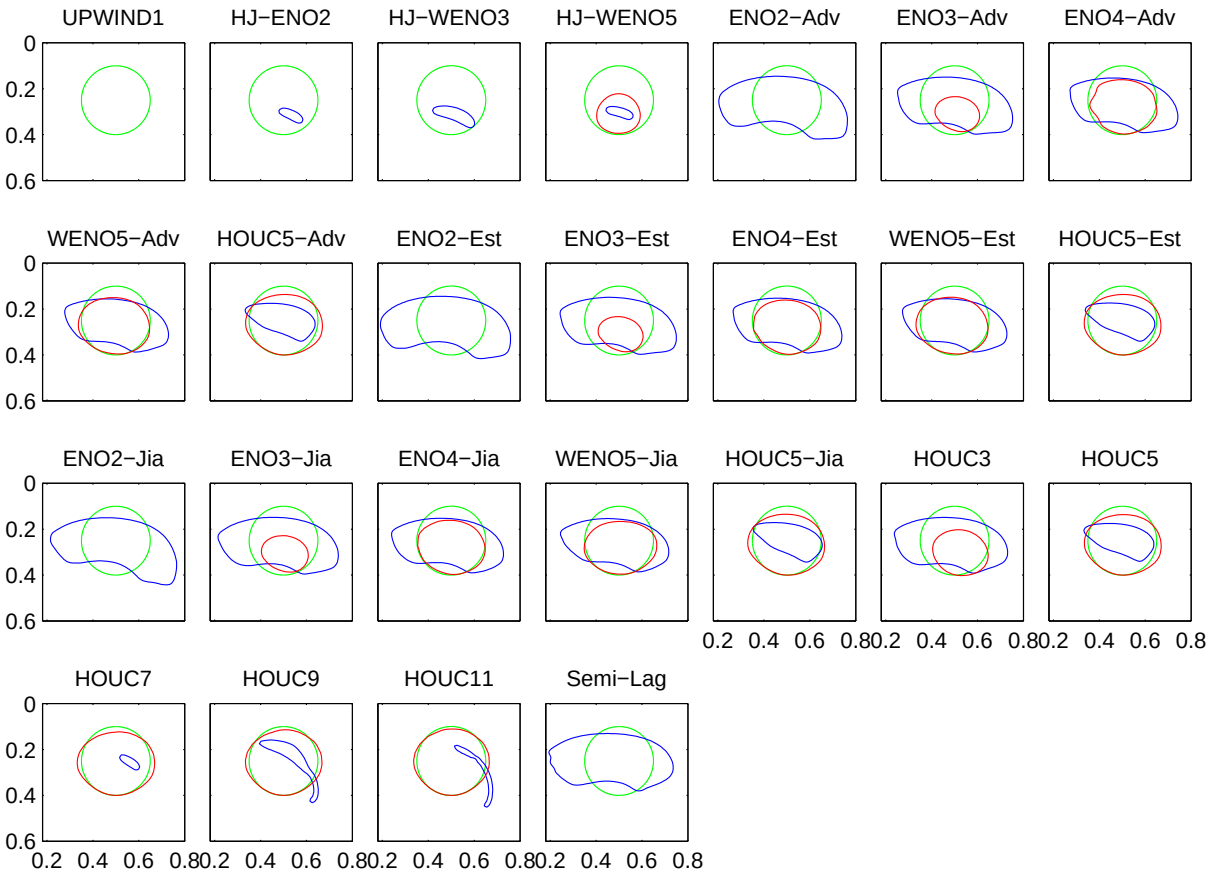

Figure 1.8: Level-set with and without reinitialization: shearing vortex case on a 100^2 grid and CFL $= 0.5$, interface at $t = T = 6$ for different advection schemes. Green: Exact interface; Red: without reinitialization; Blue: HJ-WENO5 reinitialization with $\Delta\tau = \Delta x$, $\tau_{\max} = 10\Delta x$ and Nreinit $= 1$.

Time-reversed single vortex

The results for the time-reversed single vortex at $t = T/2 = 1$ are presented in Fig. 1.9 and Table 1.4. The interface is not as deformed as in the shearing vortex problem, and the numerical results are closer to the reference solution. The interface obtained is almost independent from the advection scheme, except for the UPWIND1 for which the mass loss is quite important. Fig. 1.10 shows that for $t = T$ the UPWIND1 in combination with the non-reinitialized level-set causes the loss of the total mass of the vortex. For high-order schemes, the non-reinitialized level-set fits the reference interface better than when reinitializing the level-set. We can also remark on Table 1.4 that the level-set without reinitialization is characterized by a low $E_{|\nabla\phi|}$ at time $t = T$ since the interface has not been much stretched at $t = T/2$.

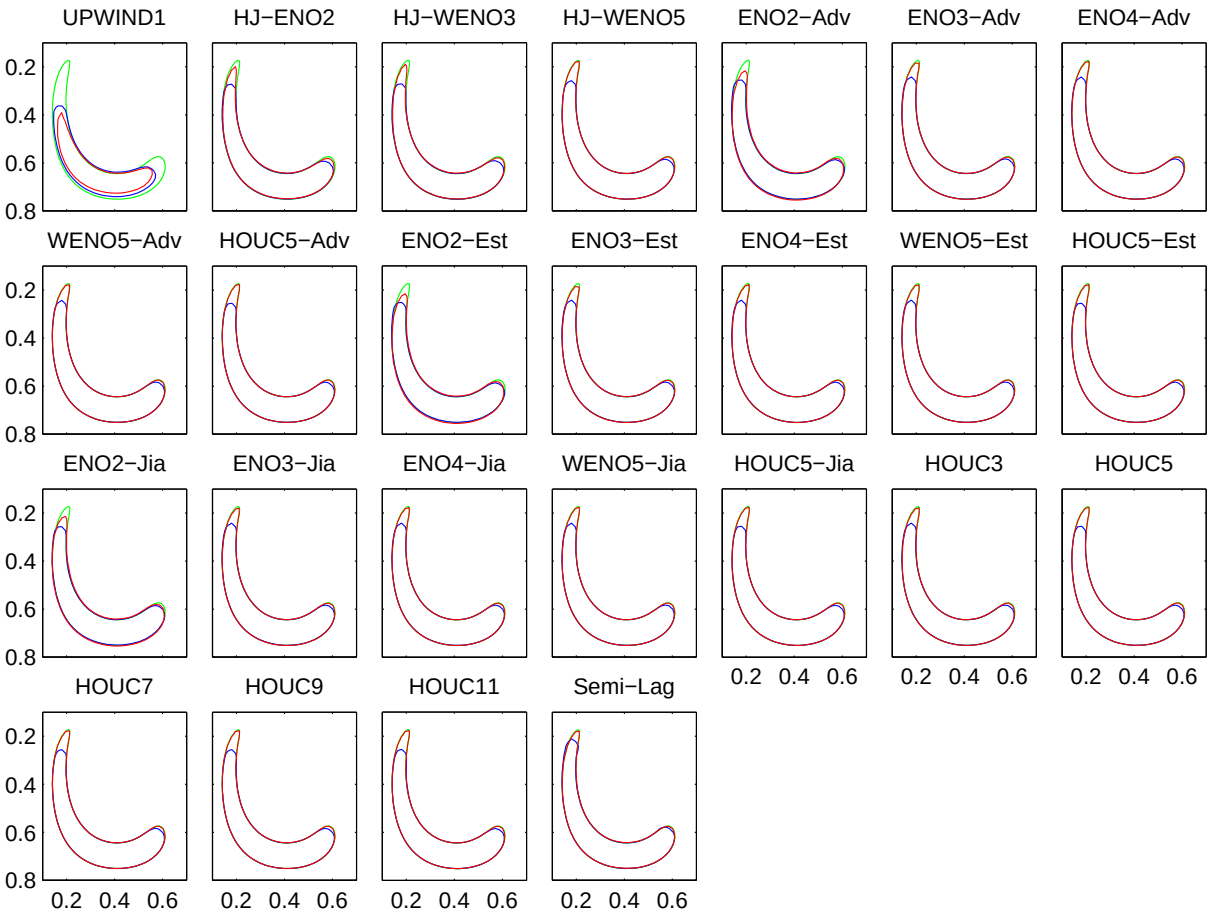

Figure 1.9: Level-set with and without reinitialization: time-reversed single vortex interface at $t = T/2 = 1$ with CFL $= 0.5$, a 64^2 grid and different advection schemes. Green: Exact interface; Red: without reinitialization; Blue: HJ-WENO5 reinitialization with $\Delta\tau = \Delta x$, $\tau_{\max} = 10\Delta x$ and Nreinit $= 1$.

Table 1.4: Comparison of level-set errors with and without reinitialization for the time-reversed single vortex test case on a 64^2 grid. Advection: CFL = 0.5. Reinitialization: HJ-WENO5 with $\Delta\tau = \Delta x$, $\tau_{\max} = 10\Delta x$ and Nreinit = 1.

Advection scheme	Time t = 1						Time t = 2					
	No reinitialization			Reinitialization			No reinitialization			Reinitialization		
	$\boldsymbol{E}_M(\%)$	$\boldsymbol{E}_1(\%)$	$\boldsymbol{E}_{\lvert\nabla\phi\rvert}$	$\boldsymbol{E}_M(\%)$	$\boldsymbol{E}_1(\%)$	$\boldsymbol{E}_{\lvert\nabla\phi\rvert}$	$\boldsymbol{E}_M(\%)$	$\boldsymbol{E}_1(\%)$	$\boldsymbol{E}_{\lvert\nabla\phi\rvert}$	$\boldsymbol{E}_M(\%)$	$\boldsymbol{E}_1(\%)$	$\boldsymbol{E}_{\lvert\nabla\phi\rvert}$
UPWIND1	−44.02	45.39	0.425	−26.22	31.29	4.95×10^{-3}	−100.00	100.00	–	−53.62	56.42	1.19×10^{-2}
HJ-ENO2	−4.71	6.34	1.21	−8.04	8.75	4.46×10^{-3}	−10.78	12.01	1.55×10^{-1}	−14.68	15.19	1.04×10^{-2}
HJ-WENO3	−2.96	4.45	1.25	−7.34	7.97	4.55×10^{-3}	−6.96	8.11	1.23×10^{-1}	−12.50	12.82	1.06×10^{-2}
HJ-WENO5	−0.23	0.72	1.36	−5.82	6.37	4.68×10^{-3}	−0.59	0.73	2.17×10^{-2}	−10.09	10.22	1.04×10^{-2}
ENO2-Adv	1.57	8.80	1.36	−5.06	7.06	4.82×10^{-3}	1.46	15.14	1.38×10^{-1}	−9.27	10.76	1.06×10^{-2}
ENO2-Est	2.07	8.67	1.36	−4.67	6.93	5.41×10^{-3}	2.79	15.45	1.26×10^{-1}	−8.93	10.51	1.07×10^{-2}
ENO2-Jia	2.11	7.61	1.39	−5.75	7.04	4.35×10^{-3}	−0.43	11.75	1.30×10^{-1}	−10.92	12.11	1.05×10^{-2}
ENO3-Adv	0.70	1.46	1.41	−4.32	5.39	3.95×10^{-3}	1.24	2.35	4.28×10^{-2}	−7.43	7.76	1.07×10^{-2}
ENO3-Est	0.61	1.53	1.40	−4.25	5.35	4.07×10^{-3}	1.41	2.45	4.24×10^{-2}	−7.36	7.71	1.07×10^{-2}
ENO3-Jia	0.68	1.50	1.41	−4.31	5.47	4.05×10^{-3}	1.11	2.32	4.56×10^{-2}	−7.69	7.98	1.07×10^{-2}
ENO4-Adv	0.29	0.70	1.40	−4.48	5.34	7.01×10^{-3}	0.61	0.82	1.89×10^{-2}	−7.66	7.85	1.07×10^{-2}
ENO4-Est	0.10	0.81	1.40	−4.43	5.30	7.05×10^{-3}	0.35	1.03	2.28×10^{-2}	−7.61	7.81	1.07×10^{-2}
ENO4-Jia	0.32	0.71	1.40	−4.41	5.38	7.18×10^{-3}	0.58	0.83	2.05×10^{-2}	−7.67	7.87	1.08×10^{-2}
WENO5-Adv	0.17	0.61	1.40	−4.52	5.32	7.03×10^{-3}	0.35	0.59	1.80×10^{-2}	−7.80	7.91	1.07×10^{-2}
WENO5-Est	0.05	0.66	1.39	−4.48	5.27	7.07×10^{-3}	0.36	0.61	1.90×10^{-2}	−7.74	7.86	1.07×10^{-2}
WENO5-Jia	0.15	0.68	1.40	−4.45	5.33	7.22×10^{-3}	0.44	0.77	1.97×10^{-2}	−7.81	7.93	1.07×10^{-2}
HOUC5-Adv	−0.01	0.53	1.39	−5.57	6.09	5.65×10^{-3}	−0.07	0.47	1.16×10^{-2}	−9.94	10.03	1.05×10^{-2}
HOUC5-Est	−0.14	0.58	1.38	−5.54	6.03	5.62×10^{-3}	−0.06	0.47	1.12×10^{-2}	−9.90	10.00	1.05×10^{-2}
HOUC5-Jia	0.02	0.61	1.39	−5.69	6.21	5.57×10^{-3}	−0.01	0.49	1.11×10^{-2}	−9.98	10.08	1.05×10^{-2}
HOUC3	0.42	1.24	1.40	−4.32	5.32	5.38×10^{-3}	1.21	2.18	3.82×10^{-2}	−7.48	7.74	1.07×10^{-2}
HOUC5	−0.01	0.53	1.39	−5.57	6.09	5.65×10^{-3}	−0.07	0.47	1.16×10^{-2}	−9.94	10.03	1.05×10^{-2}
HOUC7	0.07	0.66	1.38	−5.49	6.35	5.19×10^{-3}	−0.08	0.21	7.95×10^{-3}	−9.80	10.10	1.05×10^{-2}
HOUC9	0.08	0.64	1.39	−5.37	6.48	6.50×10^{-3}	−0.01	0.18	5.39×10^{-3}	−9.66	10.03	1.05×10^{-2}
HOUC11	0.06	0.59	1.39	−5.23	6.56	6.17×10^{-3}	0.02	0.13	4.63×10^{-3}	−9.58	9.99	1.05×10^{-2}
Semi-Lag	0.42	1.88	1.29	−1.53	3.59	7.73×10^{-3}	0.69	2.33	5.85×10^{-2}	−2.91	4.19	1.12×10^{-2}

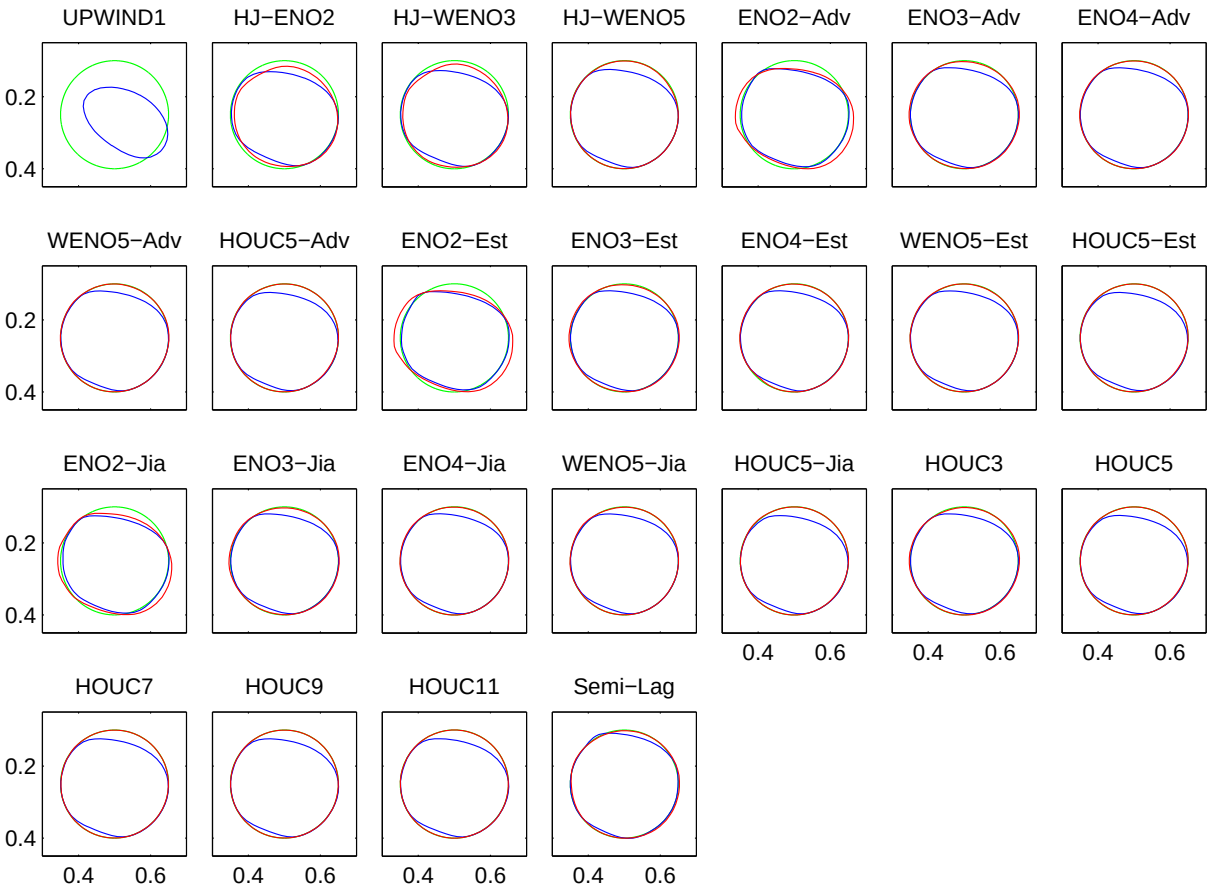

Figure 1.10: Level-set with and without reinitialization: time-reversed single vortex interface at $t = T = 2$ with CFL $= 0.5$, a 64^2 grid and different advection schemes. Green: Exact interface; Red: without reinitialization; Blue: HJ-WENO5 reinitialization with $\Delta\tau = \Delta x$, $\tau_{\max} = 10\Delta x$ and Nreinit $= 1$.

As a conclusion, for the high-order spatial discretization of the advection equation, the interface representation and advection is more accurate when the level-set is not reinitialized. However, the level-set then diverges from the signed-distance function ($|\nabla\phi| \neq 1$). The reinitialization step helps in keeping the level-set as close as possible to a signed-distance function, and improves the accuracy for low-order schemes. Nevertheless, it deteriorates the accuracy for high-order schemes, in particular the HOUCk schemes with $k \geq 7$. There are two possible contributions to this deterioration: displacement of the interface and introduction of perturbations in the level-set field during reinitialization. The sub-cell fix and the volume constraint reinitializations are two methods helping to prevent the interface displacement while reinitializing. In the next subsections, we study the influence of the different schemes for the advection and the reinitialization on both methods, the sub-cell fix and the volume constraint, on different grid sizes and CFL numbers.

1.6.2 Sub-cell fix

We now include in our tests the sub-cell fix, described in §1.3.1. First, we investigate the influence of the spatial discretization scheme for advection on this method. Then, we verify the sensibility of the method to the CFL number variations. Finally, we study the convergence rate of the method as a function of the grid size. The influence of the reinitialization scheme will be also investigated. In this section, the reinitialization equation is solved during a fictitious time $\tau_{max} = 10\Delta x$ with a step of $\Delta\tau = \Delta x/2$.

1.6.2.1 Effect of the advection scheme

The reinitialization equation is discretized in space using the HJ-ENO2 scheme and is solved after each Nreinit $= 10$ advection steps. The CFL number for advection based on the maximum velocity is set to 0.5. A 100^2 grid is adopted for both test cases, the Zalesak disk and the shearing vortex, while a 64^2 grid is used for the time-reversed single vortex problem.

The resulting interfaces are plotted on

- Fig. 1.11 for the Zalesak disk after a full rotation
- Figs. 1.12 and 1.13 for the single vortex at times $t = T/2 = 3$ and $t = T = 6$, respectively.
- Figs. 1.14 and 1.15 for the time-reversed single vortex at $t = T/2 = 1$ and $t = T = 2$, respectively.

and the errors for all cases are summarized on Table 1.5.

For the Zalesak disk, we remark that the slot has been entirely smoothed when the method is advected using the UPWIND1 scheme. On the contrary, HJ-ENO2 and HJ-WENO3 keeps the circular form but the slot is almost half its exact height. The HOUC and conservative ENO/WENO schemes lead to a deformation of the circular part of the disk and the slab. This is due to the displacement of the interface during the reinitialization procedure. The HJ-WENO5 scheme deforms less the interface and predicts its position better than the other schemes. We see that the Zalesak disk mass is conserved up to $\pm 3\%$ for all advection schemes except for the 1st- and 2nd-order schemes and the HJ-WENO3 scheme.

In Fig. 1.12, the shearing vortex result at $t = T/2 = 3$ shows important deformations: the filament thickness at time $T/2$ is too small and the level-set method has the tendency either to make it thicker (in this case we gain mass) or to let it vanish (and in this case we loose mass). The errors are much larger than for the Zalesak disk, the mass gain/loss is 35% at best for the HOUC11 advection scheme. The Hamilton-Jacobi schemes have the tendency of making the

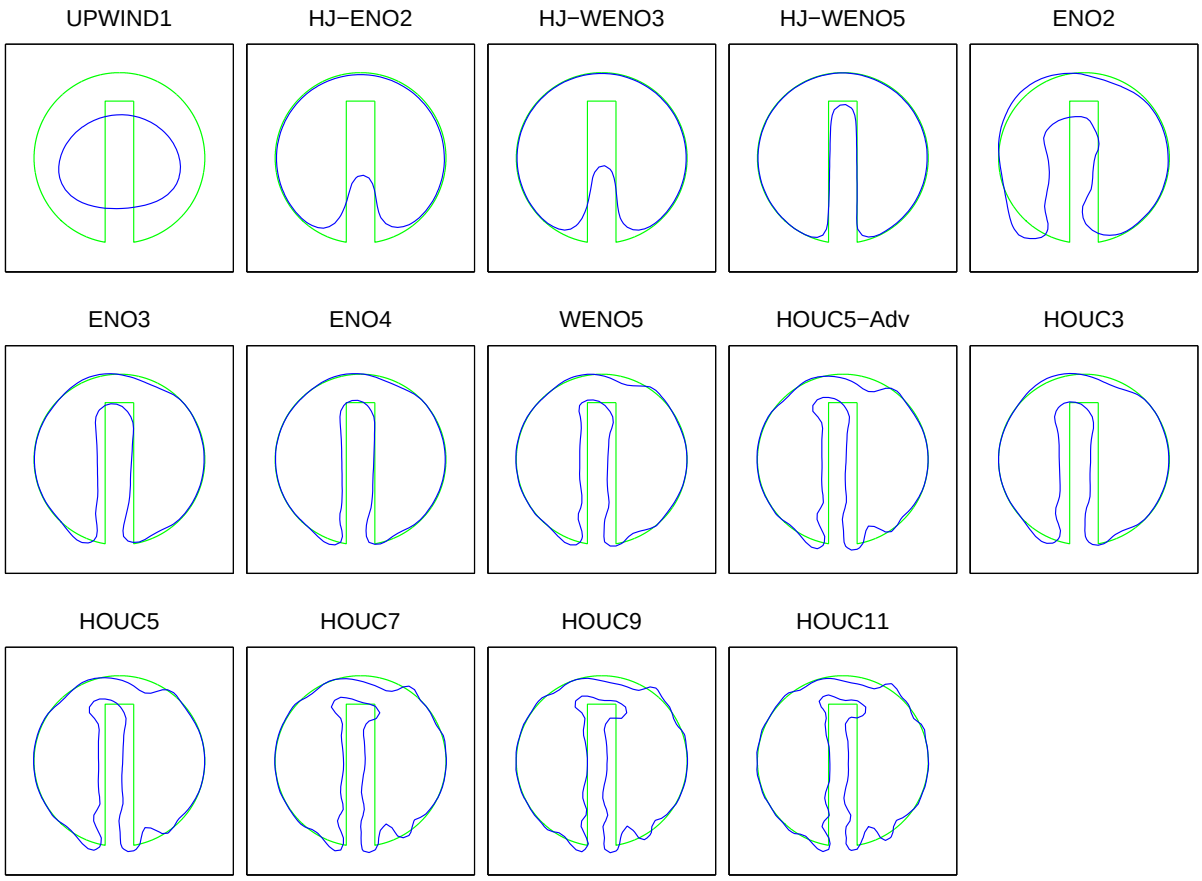

Figure 1.11: Zalesak disk after a full rotation on a 100^2 grid with CFL $= 0.5$ and different advection schemes. Sub-cell fix method with HJ-ENO2 reinitialization. $\Delta\tau = \Delta x/2$, $\tau_{\max} = 10\Delta x$ and Nreinit $= 10$ Green: Exact interface ; Blue: Sub-cell fix method.

vortex loosing mass, while the other schemes increase the mass of the vortex. High-order HOUC schemes seem to be more accurate for this simulation, at time $T/2$ as well as at time T (Fig. 1.13). The interface mass increases between the start of the simulation and time $T/2$, then decreases until the end of the simulation for HOUC schemes of 5th-order or higher; this is due to the fact that these schemes use a fixed stencil, therefore the same stencils are used when advecting the vortex forward and backward in time. At time T, the interface form when advected with one of the HOUC schemes is very close to what presented by Nourgaliev and Theofanous (2007).

The time-reversed single vortex simulation is similar to the shearing vortex with a small period T. The mass errors at $t = T/2 = 1$ (Fig. 1.14) are lower than 1% for all the advection schemes except the Hamilton-Jacobi type. At the end of computation (Fig. 1.15), only the fifth-order HJ-WENO scheme fits better the interface.

In Table 1.5, the mass error E_M is always smaller than the positioning errors except for the shearing vortex simulation at $T/2$ with some advection schemes. This is due only to the fact that if the mass increases/decreases, it will be taken into account in the positioning error. The HJ-WENO5 scheme produces the smallest mass and positioning errors for all the simulations, but it does not reproduce thin structures very well.

Table 1.5: Sub-cell fix: Zalesak disk, shearing vortex and time-reversed single vortex interface errors.

Adv. scheme	**Zalesak disk Grid 100^2**		**Shearing vortex Grid 100^2**				**Time-reversed single vortex Grid 64^2**			
	t = 6.28		t = T/2 = 3		t = T = 6		t = T/2 = 1		t = T = 2	
	$\boldsymbol{E_M}(\%)$	$\boldsymbol{E_1}(\%)$	$\boldsymbol{E_M}(\%)$	$\boldsymbol{E_1}(\%)$	$\boldsymbol{E_M}(\%)$	$\boldsymbol{E_1}(\%)$	$\boldsymbol{E_M}(\%)$	$\boldsymbol{E_1}(\%)$	$\boldsymbol{E_M}(\%)$	$\boldsymbol{E_1}(\%)$
Error of the reference field										
	0.01	–	1.88	–	5.10^{-4}	–	0.007	–	5.10^{-4}	–
UPWIND1	−51.93	80.00	−100.00	100.00	−100.00	–	−23.30	28.60	−48.67	51.91
HJ-ENO2	3.30	20.83	−88.81	91.84	−100.00	100.00	−3.17	3.67	−6.40	7.13
HJ-WENO3	4.92	17.00	−84.98	88.09	−100.00	100.00	−2.45	2.74	−4.79	5.18
HJ-WENO5	−0.58	4.47	−59.43	62.05	−82.54	82.66	−1.04	1.21	−2.12	2.29
ENO2-Adv	−5.42	29.42	150.20	150.79	170.42	172.98	0.11	4.89	0.97	5.73
ENO2-Est	−5.42	29.42	152.78	153.02	172.90	174.95	0.42	5.05	1.15	5.40
ENO2-Jia	−5.42	29.42	141.06	141.72	162.36	165.17	−0.92	5.10	−0.33	4.66
ENO3-Adv	−3.46	12.83	94.39	93.21	109.59	110.64	0.15	2.11	0.10	2.58
ENO3-Est	−3.46	12.83	93.93	92.94	108.48	109.70	0.24	2.08	0.13	2.53
ENO3-Jia	−3.46	12.83	95.09	93.96	111.97	112.83	−0.10	2.73	0.57	2.72
ENO4-Adv	−2.24	8.74	78.74	77.30	92.32	93.23	0.02	1.56	−0.10	2.30
ENO4-Est	−2.24	8.74	79.06	77.64	92.86	93.63	0.09	1.52	−0.13	2.31
ENO4-Jia	−2.24	8.74	79.80	78.31	94.35	94.89	0.28	1.89	0.87	2.18
WENO5-Est	0.16	15.27	71.99	70.37	76.64	77.31	0.23	1.89	0.53	2.41
WENO5-Adv	0.16	15.27	72.30	70.68	77.57	78.16	0.17	1.94	0.59	2.40
WENO5-Jia	0.16	15.27	73.81	72.23	77.85	78.37	0.29	2.10	0.37	2.50
HOUC5-Est	−0.05	20.99	60.09	58.36	37.55	43.76	−0.54	2.18	−1.15	3.54
HOUC5-Adv	−0.05	20.99	60.41	58.69	39.85	44.94	−0.63	2.30	−0.95	3.59
HOUC5-Jia	−0.05	20.99	61.33	59.53	36.73	45.12	−0.45	2.05	−1.34	3.72
HOUC3	−2.05	19.54	90.69	89.50	100.66	101.53	−0.04	2.37	0.22	2.59
HOUC5	−0.05	20.99	60.41	58.69	39.85	44.94	−0.63	2.30	−0.95	3.59
HOUC7	0.69	19.42	47.22	45.96	19.39	37.97	−0.74	2.21	−0.97	3.91
HOUC9	1.93	17.84	39.74	39.85	10.93	36.85	−0.64	2.12	−0.85	3.76
HOUC11	2.97	16.62	34.83	36.34	6.77	36.52	−0.59	2.08	−0.74	3.51

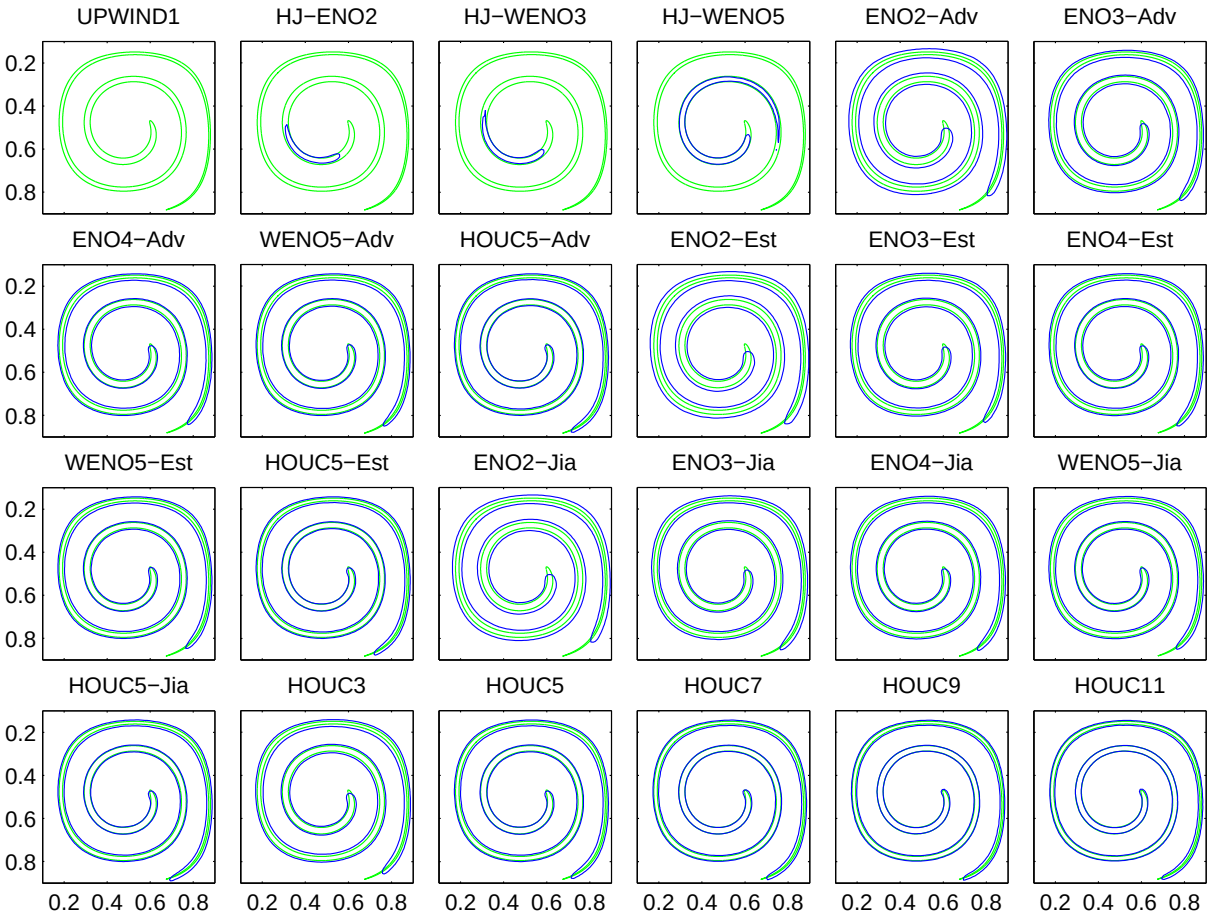

Figure 1.12: Shearing vortex at $t = 3$ on a 100^2 grid with CFL $= 0.5$. Sub-cell fix method with the HJ-ENO2 reinitialization scheme for different advection schemes. $\Delta\tau = \Delta x/2$, $\tau_{\max} = 10\Delta x$ and Nreinit $= 10$. Green: Exact interface ; Blue: Sub-cell fix method.

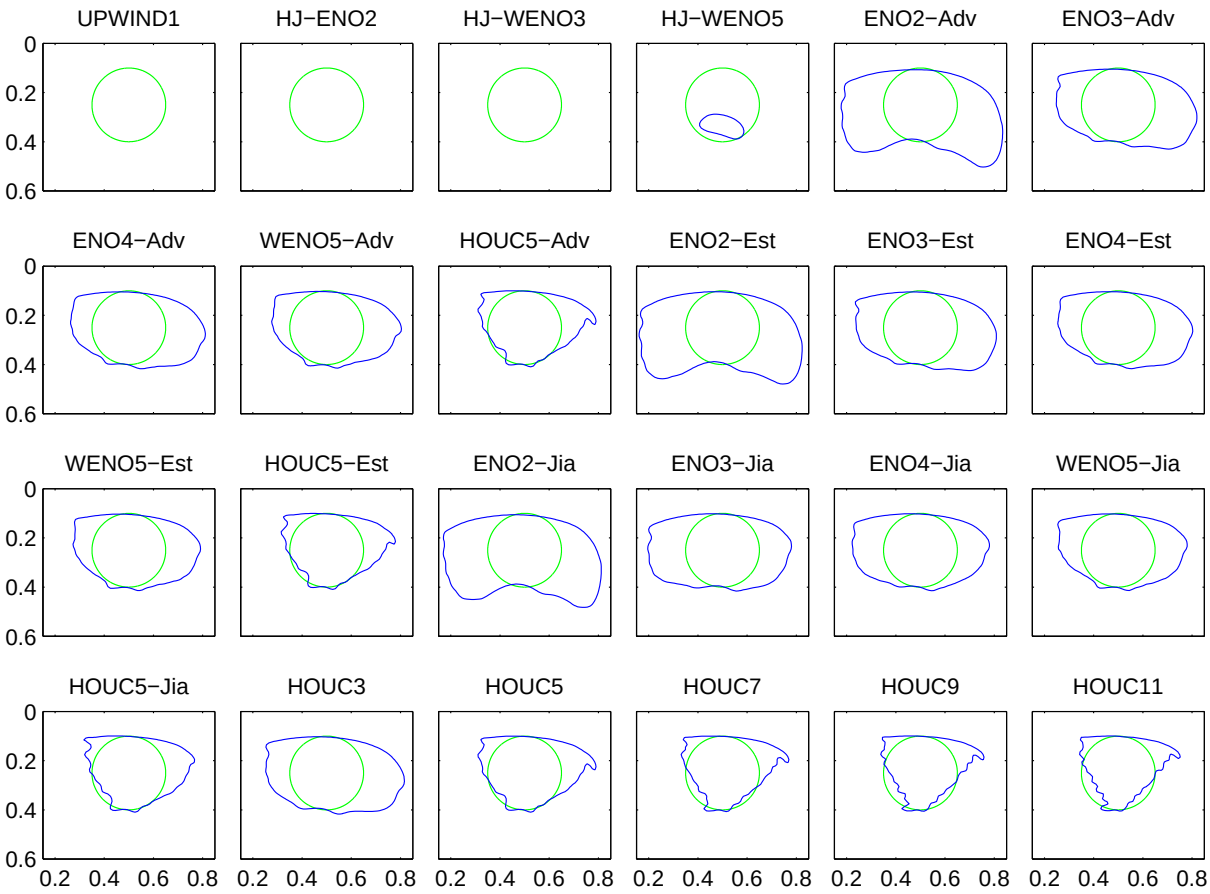

Figure 1.13: Shearing vortex at $t = 6$ on a 100^2 grid with CFL $= 0.5$. Sub-cell fix method with the HJ-ENO2 reinitialization scheme for different advection schemes. $\Delta\tau = \Delta x/2$, $\tau_{\max} = 10\Delta x$ and Nreinit $= 10$. Green: Exact interface ; Blue: Sub-cell fix method.

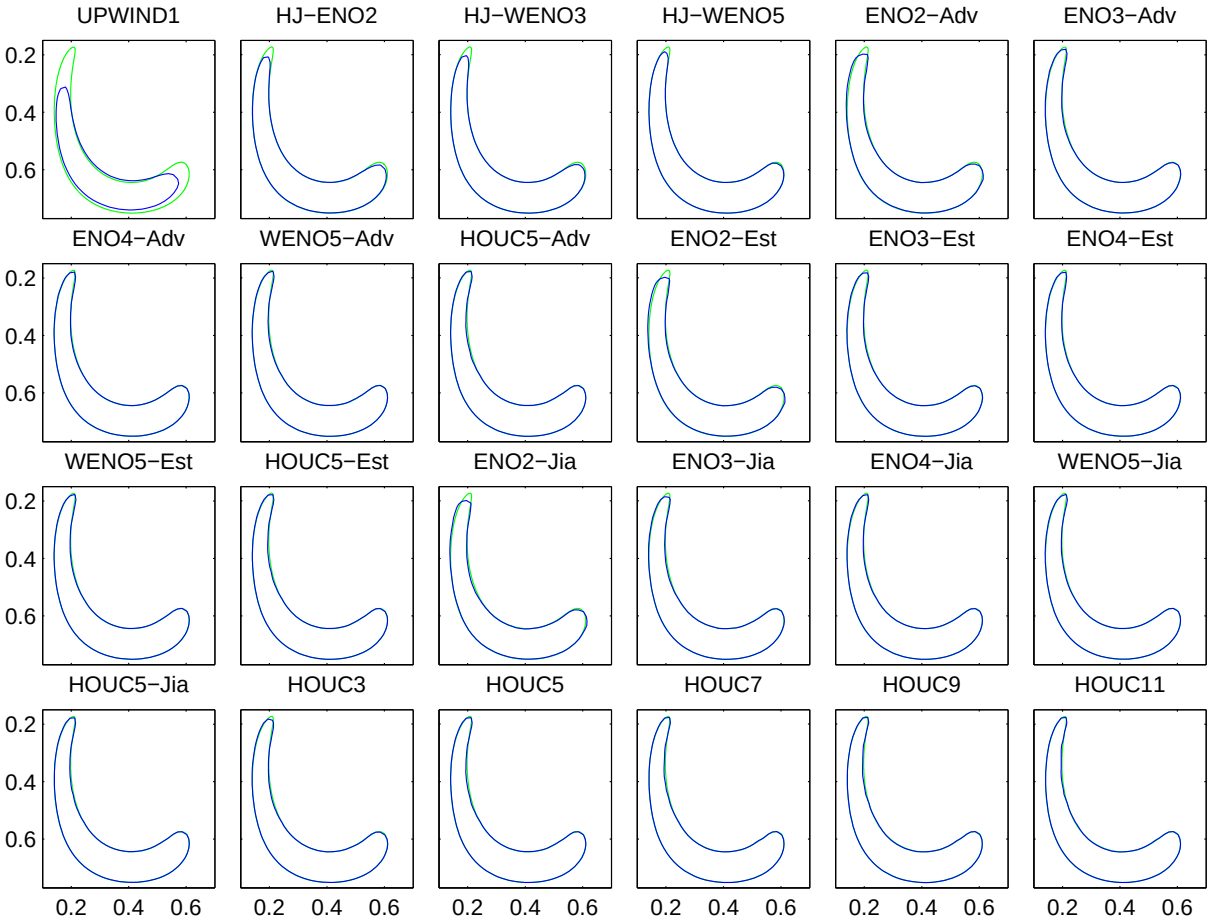

Figure 1.14: Time-reversed single vortex at $t = T/2 = 1$ on a 64^2 grid with CFL $= 0.5$. Sub-cell fix method with the HJ-ENO2 reinitialization scheme for different advection schemes. $\Delta\tau = \Delta x/2$, $\tau_{\max} = 10\Delta x$ and Nreinit $= 10$. Green: Exact interface ; Blue: Sub-cell fix method.

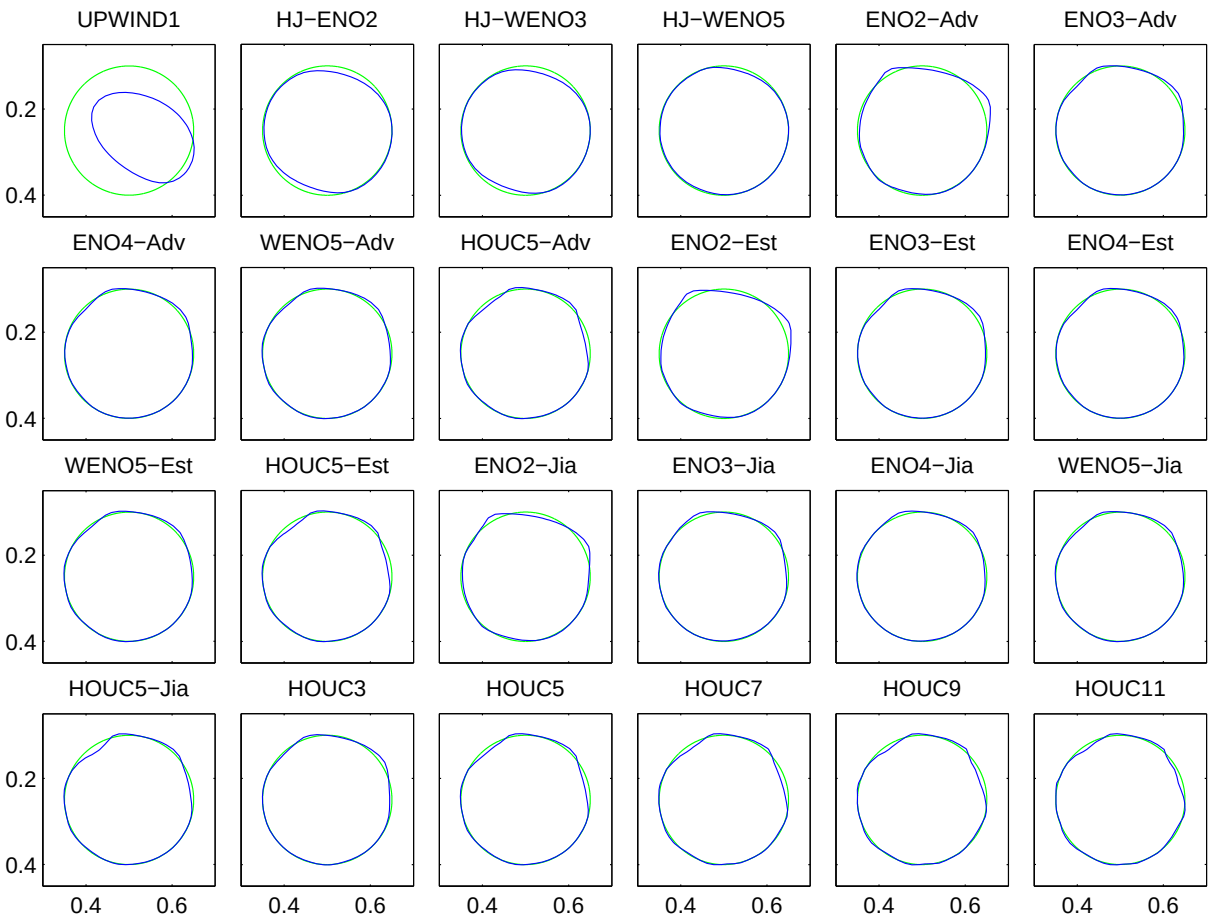

Figure 1.15: Time-reversed single vortex at $t = T = 2$ on a 64^2 grid with CFL $= 0.5$. Sub-cell fix method with the HJ-ENO2 reinitialization scheme for different advection schemes. $\Delta\tau = \Delta x/2$, $\tau_{\max} = 10\Delta x$ and Nreinit $= 10$. Green: Exact interface ; Blue: Sub-cell fix method.

1.6.2.2 Sensibility to the CFL number

The influence of the CFL number on the sub-cell fix method will be investigated for three advection schemes, which are the HJ-WENO5, the WENO5-Jia and the HOUC5 schemes. We use the HJ-ENO2 scheme for the reinitialization. Also, we define a CFL number related to the reinitialization, that we name CFLreinit, defined as:

$$\text{CFLreinit} = \text{CFL} \times \text{Nreinit} \tag{1.65}$$

where Nreinit is the number of advection steps before each reinitialization of the level-set. The CFLreinit should not be confused with the CFL number for the reinitialization equation, which is kept constant equal to 0.5 since $\Delta\tau = \Delta x/2$. The CFLreinit represents a link between the reinitialization and the advection steps. A different approach was investigated in Gómez et al. (2005) and later used by Herrmann (2008), which consist of reinitializing the level-set if its deviation from the signed-distance function $E_{|\nabla\phi|}$ is greater than a certain value. This latter is not investigated in the current work. Our approach is interesting in the sense where in practical simulations, the deviation of the level-set from the signed-distance is not computed after each advection step, contrary to the approach of Gómez et al. (2005).

Fig. 1.16 presents in solid lines (for Nreinit = 10) the positioning error E_1, the mass error E_M and the mean absolute deviation of the level-set from a signed-distance function $E_{|\nabla\phi|}$ as a function of the CFL number for the three test cases. For the three advection schemes, the errors are strongly dependent on the CFL number. Near CFL = 1.0, the errors seem to stabilize and are in general very small. However, when decreasing the CFL number these errors increase to unacceptable levels. This is principally due to the fact that for small CFL numbers, the reinitialization frequency during the simulation is very high and each reinitialization deteriorates the solution. The changes in the level-set field for small time steps are also very limited and does not justify a reinitialization every 10 time steps. In conclusion, for all cases and advection schemes, the errors are minimal only for CFL numbers in the range $[0.1, 1]$ when the Nreinit is kept equal to 10.

When on the other hand the CFLreinit is kept constant, the interface errors remain constant and the simulations do not diverge even for very small CFL values (dash-dot lines with cross markers on Fig. 1.16). The errors are plotted for the CFLreinit leading to the smallest errors for all the simulations for each advection scheme. Our choice is CFLreinit = 2 for the HJ-WENO5 scheme, 8 for the WENO5-Jia scheme and 9 for the HOUC5 scheme. We remark also that the signed-distance property is not altered when keeping CFLreinit constant, since $E_{|\nabla\phi|}$ remains constant.

Fig. 1.17 shows the dependency of the interface errors on the CFLreinit value in the range $[10^{-3}, 10]$. The plotted errors correspond to two different CFL numbers: 10^{-3} (symbols) and 10^{-2} (solid lines). In general, a CFLreinit in the range $[1, 10]$ seems to be a good choice.

1.6.2.3 Influence of the reinitialization scheme

Now, we test the different reinitialization schemes coupled with the sub-fix method. The advection schemes that we employ here are the HJ-WENO5 and WENO5-Jia schemes. The HJ-WENO5 scheme is the one giving good results for almost all the simulations. Table 1.6 summarizes the errors E_M and E_1 for the Zalesak disk and the shearing vortex simulations on a 100^2 grid and for the time-reversed single vortex on a 64^2 grid. The CFL number and CFLreinit value are 0.5 and 5.0, respectively.

The interface errors remain in the same range when changing the reinitialization scheme. This is due to the first-order accuracy of the sub-cell fix method at the interface. From these results,

we may conclude that the HJ-ENO2 scheme seems to be the most suited for the HJ-WENO5 advection, while the HJ-ENO4 scheme seems to be more suited for the WENO5-Jia advection.

1.6.2.4 Convergence rate

The results presented previously have been computed on a 100^2 grid. We have shown the interface errors depend not only on the CFL number itself, but also on the frequency of reinitialization. This latter has led to the introduction of the CFLreinit. Now, we study the influence of the grid size and the CFL number on the sub-cell fix method. Only the Zalesak disk after a full revolution and the shearing vortex problems are treated. The results for the time-reversed single vortex problem are similar to the shearing vortex. The reinitialization equation is discretized in space using the HJ-ENO2 scheme.

Fig. 1.18 shows the positioning error, E_1, after a complete rotation for different CFLreinit numbers. We present also the curve for CFLreinit $= \infty$, which corresponds to a unique reinitialization of the level-set function. The reinitialization in this case is performed after the first advection step. For all high-order schemes, a nearly second-order behavior is observed except

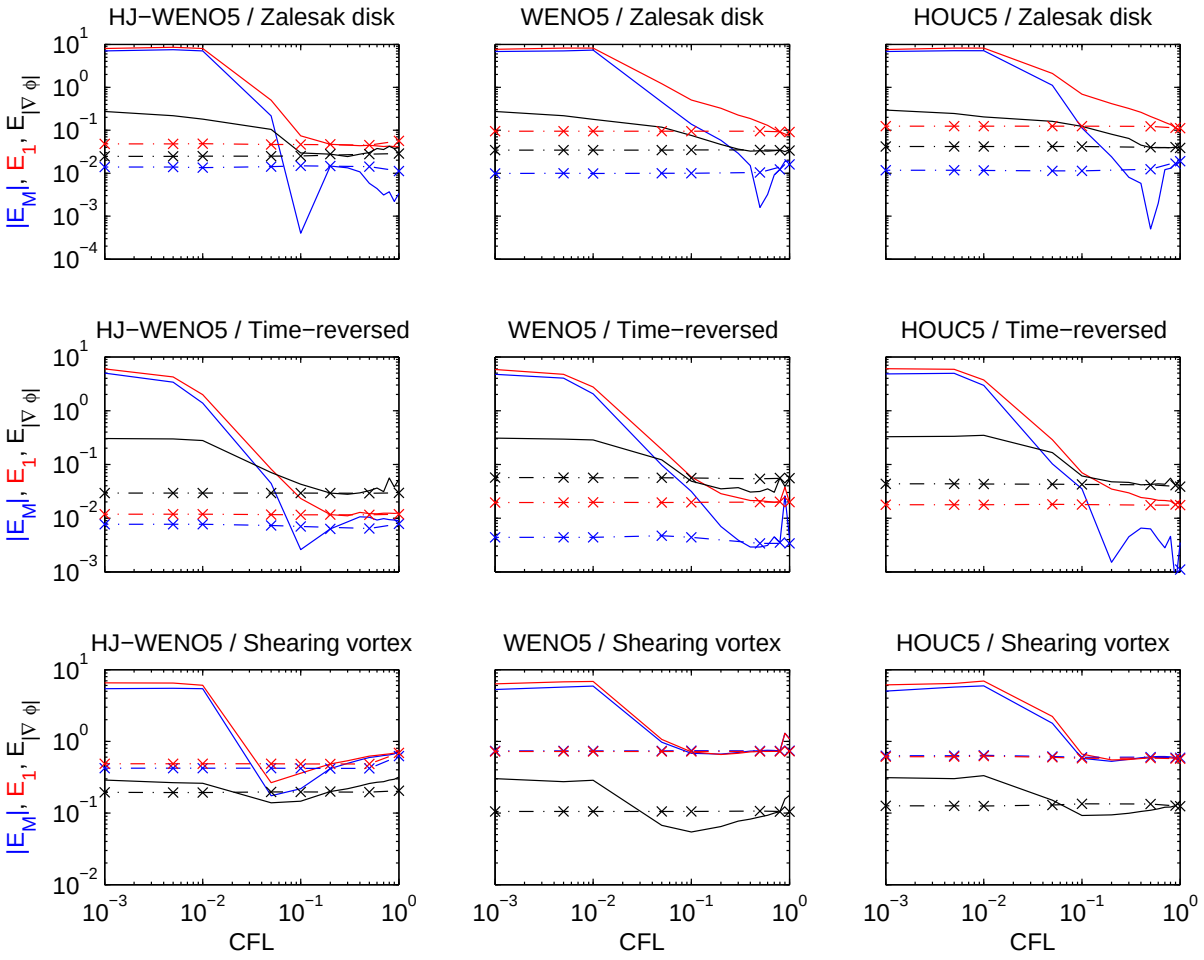

Figure 1.16: Sub-cell fix: interface errors E_M (Blue), E_1 (Red) and $E_{|\nabla\phi|}$ (Black) as a function of the CFL number. Grid : 100^2 for the Zalesak disk and the shearing vortex and 64^2 for the time-reversed single vortex. HJ-ENO2 scheme is used for the reinitialization. Solid lines : Nreinit $= 10$; dashed-dotted lines with cross markers: constant CFLreinit ($= 2$ for HJ-WENO5; $= 8$ for WENO5 and $= 9$ for HOUC5).

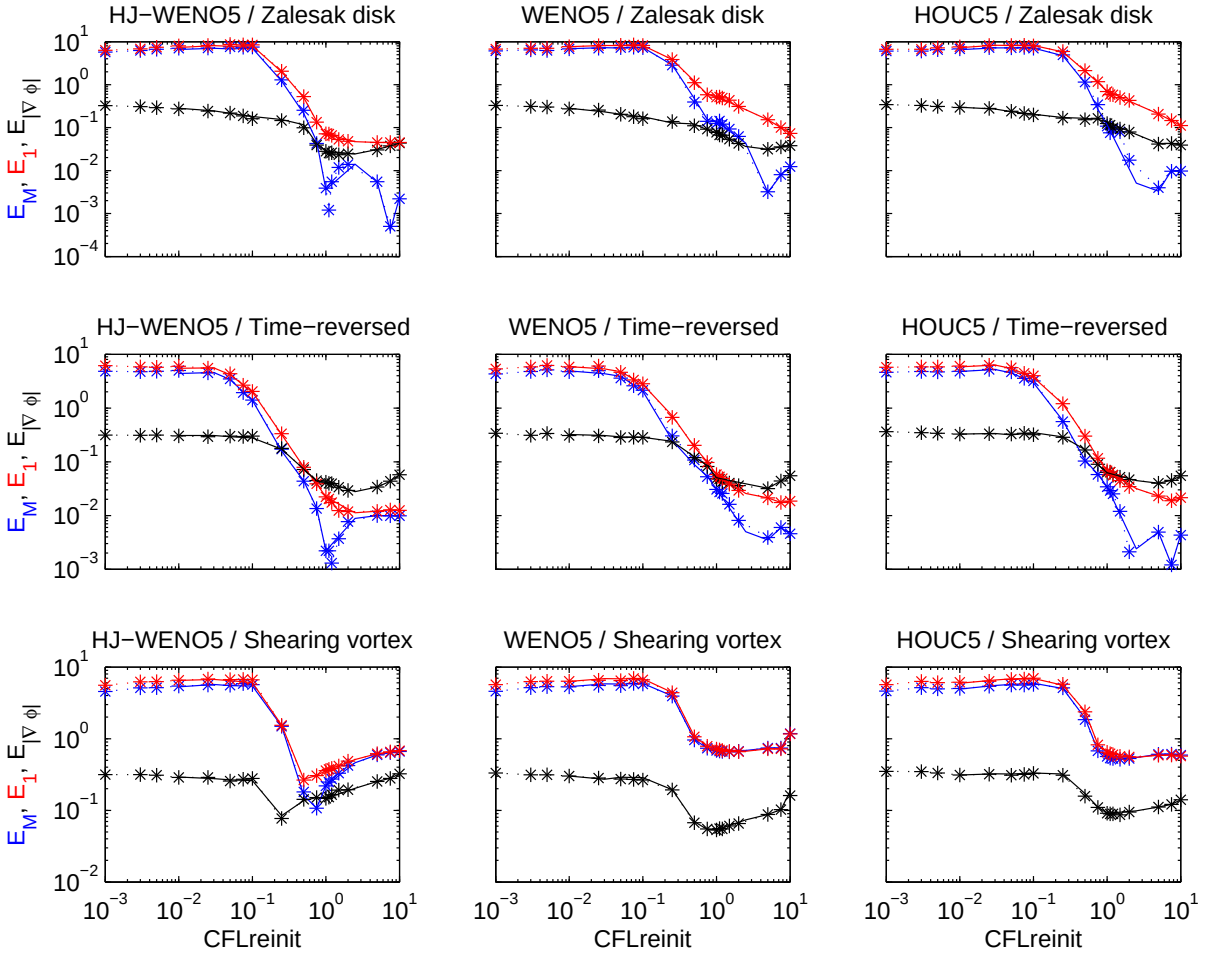

Figure 1.17: Sub-cell fix: interface errors E_M (Blue), E_1 (Red) and $E_{|\nabla\phi|}$ (Black) as a function of the CFLreinit number. Grid: 100^2 for the Zalesak disk and the shearing vortex and 64^2 for the time-reversed single vortex. HJ-ENO2 scheme for the reinitialization. Solid lines: CFL $= 10^{-2}$; symbols: CFL $= 10^{-3}$.

Table 1.6: Sub-cell fix: effect of the reinitialization scheme on the HJ-WENO5 and the WENO5 advection schemes. Errors are expressed in %.

Simulation	**Zalesak disk** $t = 6.28$, Grid 100^2				**Shearing vortex** $t = T/2 = 3$, Grid 100^2				**Time-reversed single vortex** $t = T/2 = 1$, Grid 64^2			
Adv. sch.	**HJ-WENO5**		**WENO5**		**HJ-WENO5**		**WENO5**		**HJ-WENO5**		**WENO5**	
Reinit. sch.	E_M	E_1	E_M	E_1	E_M	E_1	E_M	E_1	E_M	E_1	E_M	E_1
UPWIND1	−0.99	5.19	1.07	15.09	−59.54	62.36	73.79	72.34	−1.23	1.43	0.48	2.24
HJ-ENO2	−0.58	4.47	0.16	15.27	−59.43	62.05	73.81	72.23	−1.04	1.21	0.29	2.10
HJ-ENO3	−1.01	5.34	0.35	15.18	−60.30	62.91	74.05	72.36	−1.05	1.24	0.37	2.05
HJ-ENO4	−2.44	6.39	0.59	14.86	−61.38	63.95	73.97	72.28	−1.27	1.53	0.45	2.01
HJ-WENO3	−0.55	4.66	0.09	15.25	−59.51	62.13	73.80	72.22	−1.06	1.23	0.27	2.10
HJ-WENO5	−0.61	5.30	0.29	15.15	−60.11	62.69	74.24	72.51	−0.96	1.16	0.31	2.15

for very fine grids and for small values of CFLreinit. The smallest errors and a genuine second-order accuracy is obtained for the HOUC schemes without reinitialization, but the same schemes show very poor results when the sub-cell fix reinitialization is used. This is also illustrated in Fig. 1.20. The frequency of reinitialization has again an important influence on the accuracy of the interface method. For fine grids, higher values of CFLreinit seem to give better results.

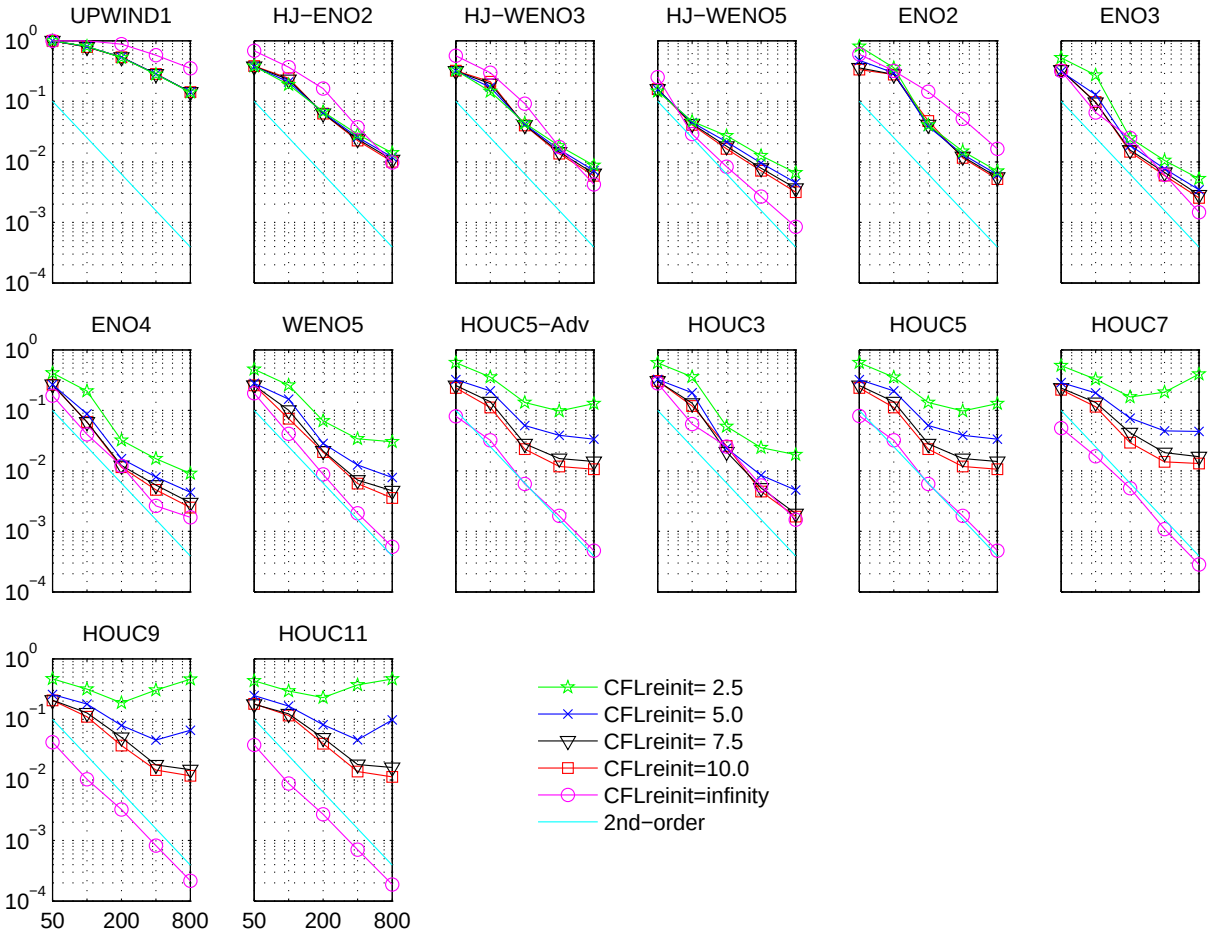

Figure 1.18: Sub-cell fix: Zalesak disk positioning error, E_1, after a full rotation for different CFLreinit numbers as a function of the grid size. the HJ-ENO2 scheme is used for the reinitialization.

The positioning error E_1 of the shearing vortex simulation at time $t = T/2 = 3$ is shown on Fig. 1.19. Again, we see that for the HJ schemes reinitialization enhance the interface capturing. However, for the other schemes of order greater than 2, the reinitialization moves the interface and causes the increase in the positioning error, especially for the HOUC schemes where the positioning error E_1 is of the range of the mass error of the reference solution. When reinitializing often the sub-cell fix method with HOUC scheme, we obtain an interface similar to what is plotted in Fig. 1.20. Since the vortex does not contain sharp angles, the positioning error is nearly second-order for most methods, except at fine grids and small values of CFLreinit. The UPWIND1 scheme is at best only first-order accurate.

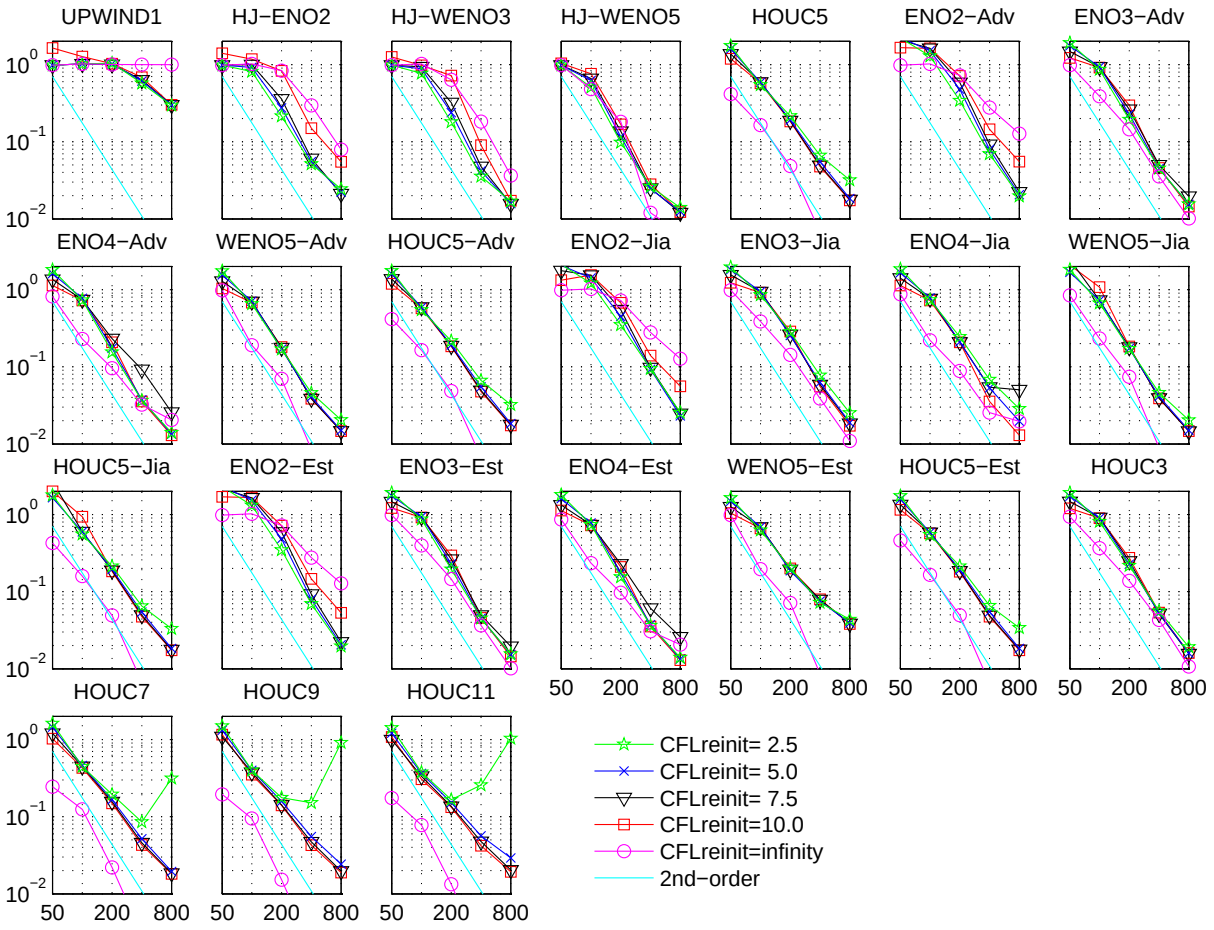

Figure 1.19: Sub-cell fix: shearing vortex positioning error, E_1, at $t = T/2 = 3$ for different CFLreinit numbers as a function of the grid size. The HJ-ENO2 scheme is used for the reinitialization.

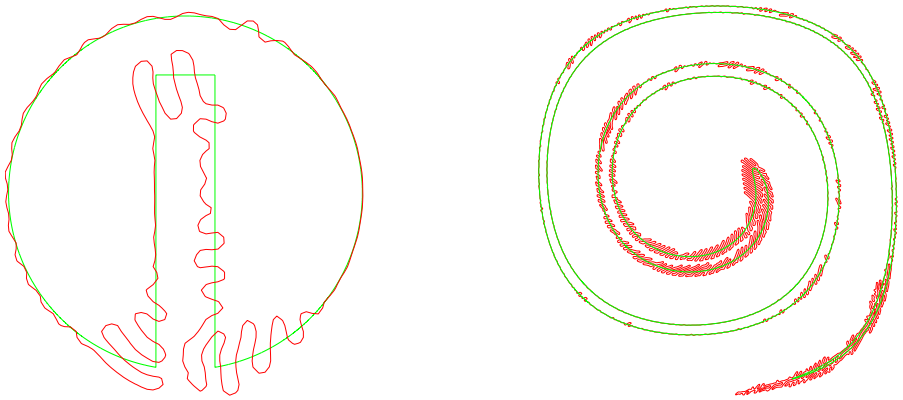

Figure 1.20: Sub-cell fix: the 7th-order HOUC scheme for the advection with CFL = 0.25 and Nreinit=10. left: Zalesak disk after a full revolution on a 200^2 grid and right: shearing vortex at $t = T/2 = 3$ on a 800^2 grid. The HJ-ENO2 scheme is used for the reinitialization. Green: Exact interface; red: Sub-cell fix advection.

1.6.3 Volume constraint

In the previous sections, we have investigated the influence of the different parameters such as the choice of the advection and reinitialization schemes and the CFL number on the sub-cell fix method. In the current section, we will study their influence on the volume constraint method, which was described in §1.3.2.

1.6.3.1 Effect of the advection scheme

The reinitialization equation is solved here after each advection step (Nreinit= 1) during a fictitious time of $\tau_{\max} = 2\Delta x$ with a step size of $\Delta\tau = 0.5\Delta x$. The HJ-ENO2 scheme is used for the spatial discretization of the reinitialization equation. The CFL number for advection is kept constant equal to 0.5. We present the resulting interfaces obtained with different advection scheme for the following cases:

- Zalesak disk after a full rotation, in Fig. 1.21, on a 100^2 grid.
- Shearing vortex at $t = T/2 = 3$ and $t = T = 6$, in Figs. 1.22 and 1.23, on a 100^2 grid.
- Time-reversed single vortex at $t = T/2 = 1$ and $t = T = 2$, in Figs. 1.24 and 1.25, on a 64^2 grid.

Table 1.7 summarizes the positioning and the mass errors.

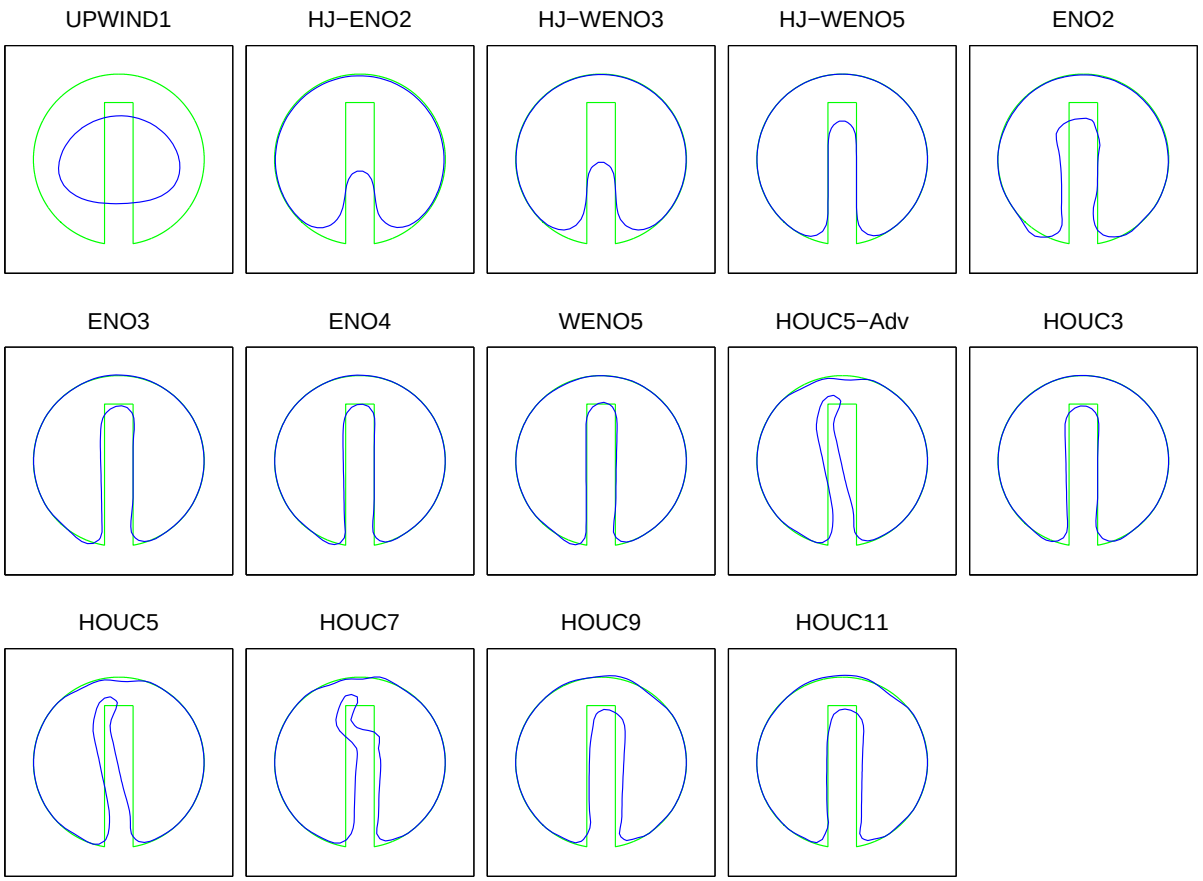

Figure 1.21: Volume constraint: interface of the Zalesak disk after a complete rotation with the different advection schemes; HJ-ENO2 scheme for the reinitialization; CFL = 0.5; Grid 100^2 Green: Exact interface ; Blue: Volume constraint.

We observe for the Zalesak disk in Fig. 1.21 a great improvement for most schemes when compared to the results obtained with reinitialization but without the volume constraint in Fig. 1.6 and those obtained with the sub-cell fix in Fig. 1.11. Only the results obtained without reinitialization in Fig. 1.6 provide better results, but this is mainly due to the solid rotation of the level-set field which in principle does not require reinitialization.

The HOUC advection schemes with fifth- or higher order are the only schemes that show a deterioration of the interface, as was the case with the sub-cell fix. The high-order Hamilton-Jacobi schemes do a very good job on the other hand. The HJ-WENO5 scheme gives the best interface transport for the sub-cell fix method, while the slab's height becomes smaller than what it is initially with the volume constraint method. For the conservative schemes, we remark that the ENO2 scheme causes the thickening of the slab. For higher order schemes, the disk fits almost the exact interface and we can hardly distinguish the ENO3, ENO4 and WENO5 results.

Concerning the shearing vortex problem (Figs. 1.22 and 1.23), the interface vanishes with the UPWIND1 advection before the time $T/2$ is reached. The filament is also much shorter with the other Hamilton-Jacobi schemes. On the other hand, the filament is much longer and becomes thicker than the exact interface with the ENO and WENO schemes. The HOUC schemes yield a thinner filament (closer to the exact interface) than the ENO and WENO schemes at $t = T/2$ and the interface has a triangular shape at $t = T$.

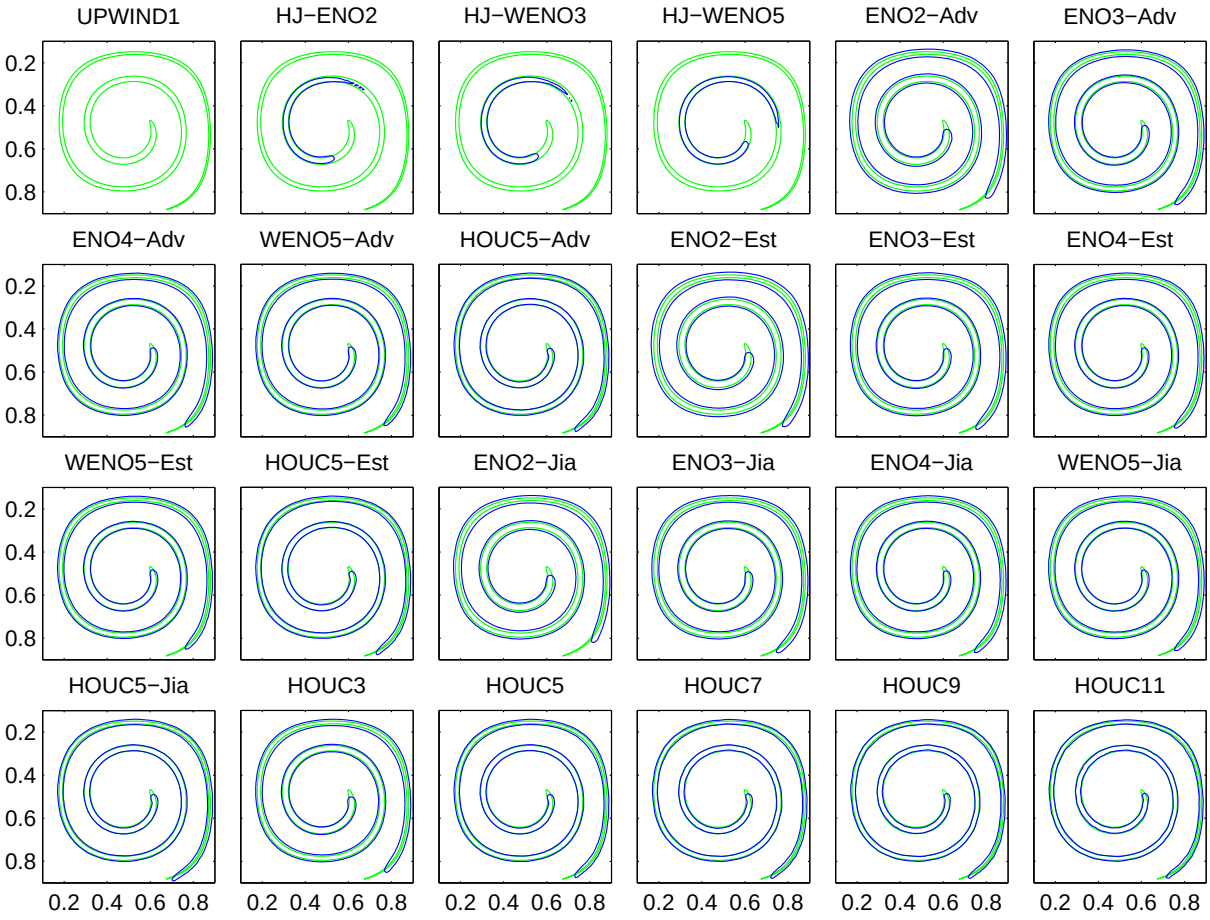

Figure 1.22: Volume constraint: shearing vortex interface at time $t = T/2 = 3$ with the different advection schemes on a 100^2 grid with CFL $= 0.5$. the HJ-ENO2 scheme for the reinitialization. $\Delta\tau = \Delta x/2$, $\tau_{\max} = 2\Delta x$ and Nreinit $= 1$. Green: Exact interface ; Blue: Volume constraint.

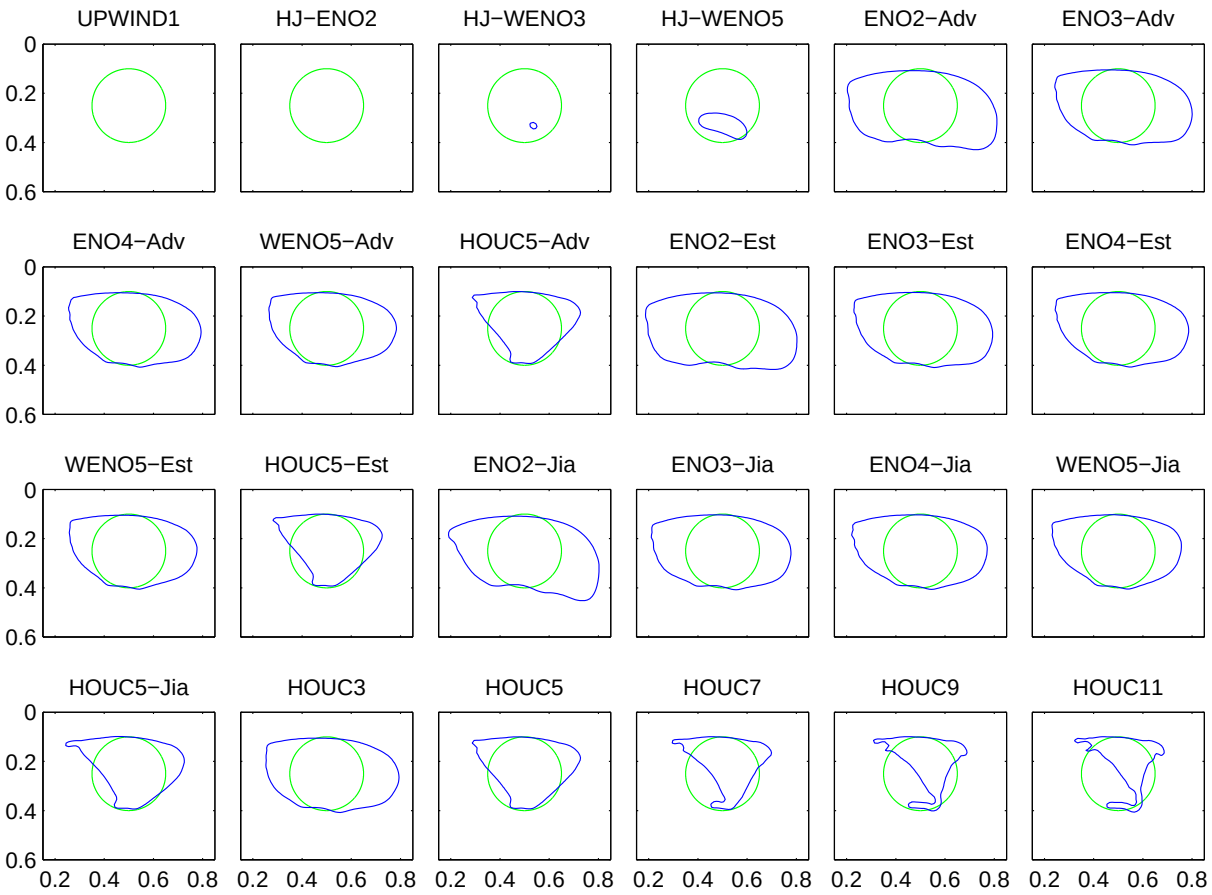

Figure 1.23: Volume constraint: shearing vortex interface at time $t = T = 6$ with the different advection schemes on a 100^2 grid with CFL $= 0.5$. the HJ-ENO2 scheme for reinitialization. $\Delta\tau = \Delta x/2$, $\tau_{\max} = 2\Delta x$ and Nreinit $= 1$. Green: Exact interface ; Blue: Volume constraint.

The results for the time-reversed single vortex (Figs. 1.24 and 1.25) are, as in the previous cases, very similar to those obtained for the shearing vortex, with the difference that the interface undergoes less deformation. The best results are obtained with the ENO3, ENO4 and WENO5 schemes at $T/2$ as well as at T, while the interface vanishes at time T with the UPWIND1 scheme.

Now, if we compare the results in Table 1.7 with those obtained with the sub-cell fix method in Table 1.5, we remark that the volume constraint generally performs better than the sub-cell fix for the Zalesak disk and the shearing vortex for all advection schemes. However, the sub-cell fix method seems to be more accurate for the time-reversed single vortex for all schemes except the ENO4 scheme in its three versions. The WENO5 scheme in its three versions (Jia, Adv and Est) is also more accurate than the HJ-WENO5 scheme for all test problems.

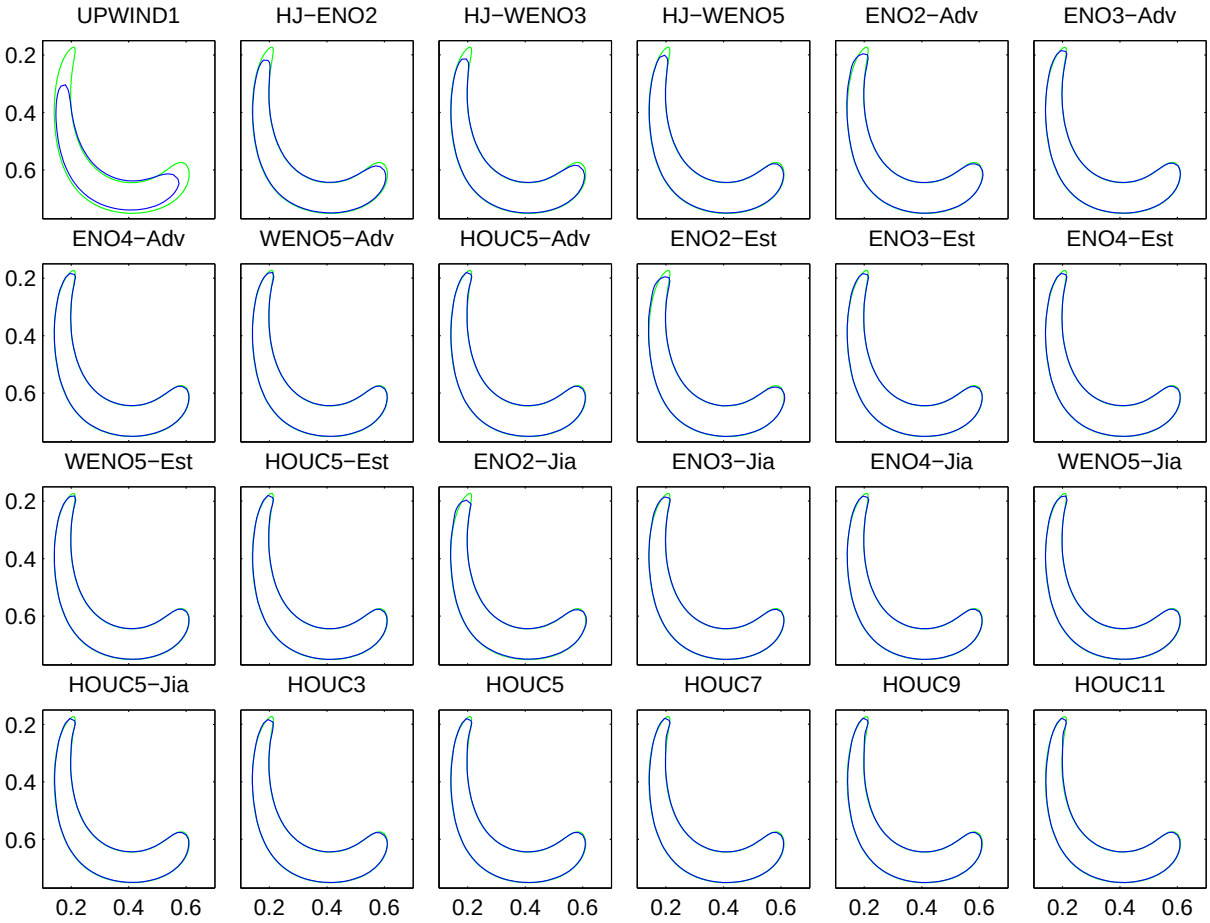

Figure 1.24: Volume constraint: time-reversed single vortex interface at time $t = T/2 = 1$ for the different advection schemes on a 64^2 grid with CFL $= 0.5$. the HJ-ENO2 scheme for the reinitialization . $\Delta\tau = \Delta x/2$, $\tau_{\max} = 2\Delta x$ and Nreinit $= 1$. Green: Exact interface ; Blue: Volume constraint.

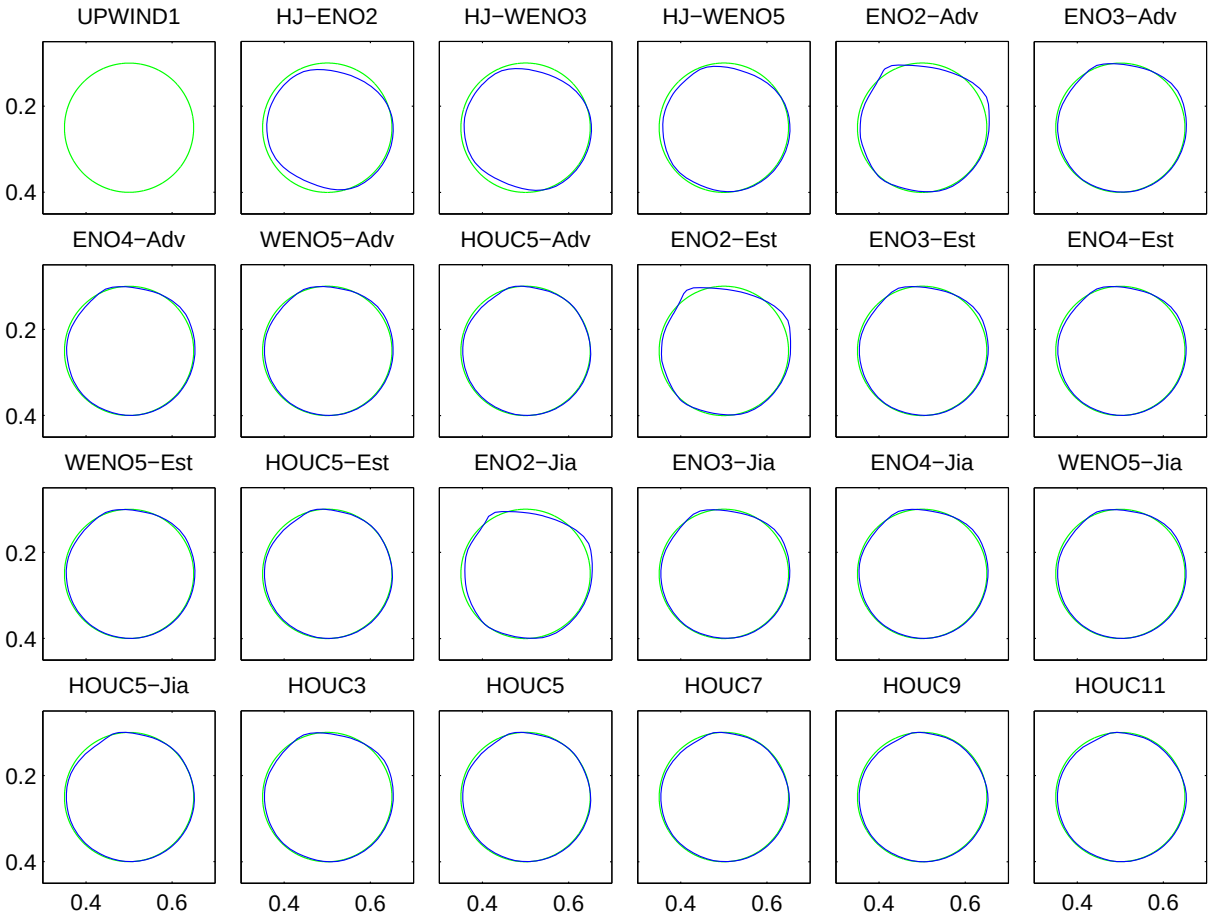

Figure 1.25: Volume constraint: time-reversed single vortex interface at time $t = T = 2$ for the different advection schemes on a 64^2 grid with CFL $= 0.5$. the HJ-ENO2 scheme is used for the reinitialization. $\Delta\tau = \Delta x/2$, $\tau_{\max} = 2\Delta x$ and Nreinit $= 1$. Green: Exact interface ; Blue: Volume constraint.

Table 1.7: Volume constraint: Zalesak disk, shearing vortex and time-reversed single vortex interface errors.

Adv. scheme	Zalesak disk Grid 100^2		Shearing vortex Grid 100^2				Time-reversed single vortex Grid 64^2			
	t = 6.28		t = T/2 = 3		t = T = 6		t = T/2 = 1		t = T = 2	
	$E_M(\%)$	$E_1(\%)$	$E_M(\%)$	$E_1(\%)$	$E_M(\%)$	$E_1(\%)$	$E_M(\%)$	$E_1(\%)$	$E_M(\%)$	$E_1(\%)$
Error of the reference field										
	0.01	–	1.88	–	5.10^{-4}	–	0.007	–	5.10^{-4}	–
UPWIND1	−55.03	81.32	−100.00	100.00	−100.00	100.00	−22.53	28.15	−47.46	51.92
HJ-ENO2	2.16	19.73	−71.65	79.08	−100.00	100.00	−4.69	6.51	−9.56	11.67
HJ-WENO3	3.98	15.41	−68.82	75.90	−99.14	99.14	−3.90	5.56	−7.83	9.70
HJ-WENO5	1.23	5.69	−60.00	63.92	−81.07	81.79	−2.50	4.14	−4.98	6.97
ENO2-Est	−3.61	11.84	108.33	108.79	124.12	126.92	0.50	3.74	0.69	6.26
ENO2-Adv	−3.61	11.84	105.94	106.83	121.50	124.92	0.23	3.72	0.29	6.50
ENO2-Jia	−3.61	11.84	97.19	98.39	113.75	117.52	−0.55	3.94	−0.83	6.66
ENO3-Est	−1.51	4.48	79.07	78.30	95.06	96.57	0.04	2.11	−0.09	3.96
ENO3-Adv	−1.51	4.48	78.45	77.75	94.40	96.10	−0.04	2.15	−0.15	4.08
ENO3-Jia	−1.51	4.48	80.09	79.30	95.95	97.31	−0.07	2.57	−0.02	3.79
ENO4-Est	−0.83	2.86	68.62	67.67	83.13	84.60	−0.21	1.93	−0.59	3.87
ENO4-Adv	−0.83	2.86	68.03	67.14	82.52	84.19	−0.28	2.01	−0.62	3.99
ENO4-Jia	−0.83	2.86	69.56	68.51	84.61	85.80	−0.13	2.23	0.01	3.42
WENO5-Est	−0.96	2.55	57.35	57.18	70.98	72.37	−0.21	2.08	−0.33	3.41
WENO5-Adv	−0.96	2.55	60.27	59.19	72.25	73.66	−0.25	2.14	−0.34	3.46
WENO5-Jia	−0.96	2.55	62.49	61.34	73.58	74.55	−0.13	2.21	−0.29	3.50
HOUC5-Est	4.28	15.39	38.50	43.59	16.27	40.72	−0.87	2.46	−1.83	3.06
HOUC5-Adv	4.28	15.39	38.27	43.54	16.70	40.12	−0.89	2.53	−1.83	3.04
HOUC5-Jia	4.28	15.39	40.44	44.68	17.62	45.01	−0.86	2.58	−1.82	3.31
HOUC3	−1.93	4.19	69.86	69.76	85.00	87.34	−0.17	2.58	−0.40	3.98
HOUC5	4.28	15.39	38.27	43.54	16.69	40.12	−0.89	2.53	−1.83	3.04
HOUC7	1.46	13.92	27.02	36.93	−18.79	50.82	−0.86	2.67	−1.88	3.20
HOUC9	−2.69	9.98	21.38	34.36	−34.63	60.29	−0.79	2.76	−1.86	3.27
HOUC11	−3.91	7.69	18.18	32.81	−42.91	66.57	−0.73	2.84	−1.84	3.28

1.6.3.2 Effect of the CFL number

We now investigate the CFL effect on the interface positioning and mass errors for the level-set method with volume constraint. The reinitialization equation is again discretized in space using the HJ-ENO2 scheme. The reinitialization equation is solved at each physical time step for a fictitious time of $\tau_{max} = 2\Delta x$ with a time step of $\Delta x/2$. We will use the WENO5 advection scheme, which gave relatively good results (see Table 1.7) and we add to this study the HJ-WENO5 and HOUC5 advection schemes.

Fig. 1.26 shows the evolution of the errors when varying the CFL number for the three test cases. As for the sub-cell fix method, the best results are obtained when the CFLreinit is kept fixed (thereby increasing Nreinit as CFL decreases). When Nreinit is kept constant, the interface error is larger when the time step is small, since the high number of reinitializations tends to deteriorate the interface position.

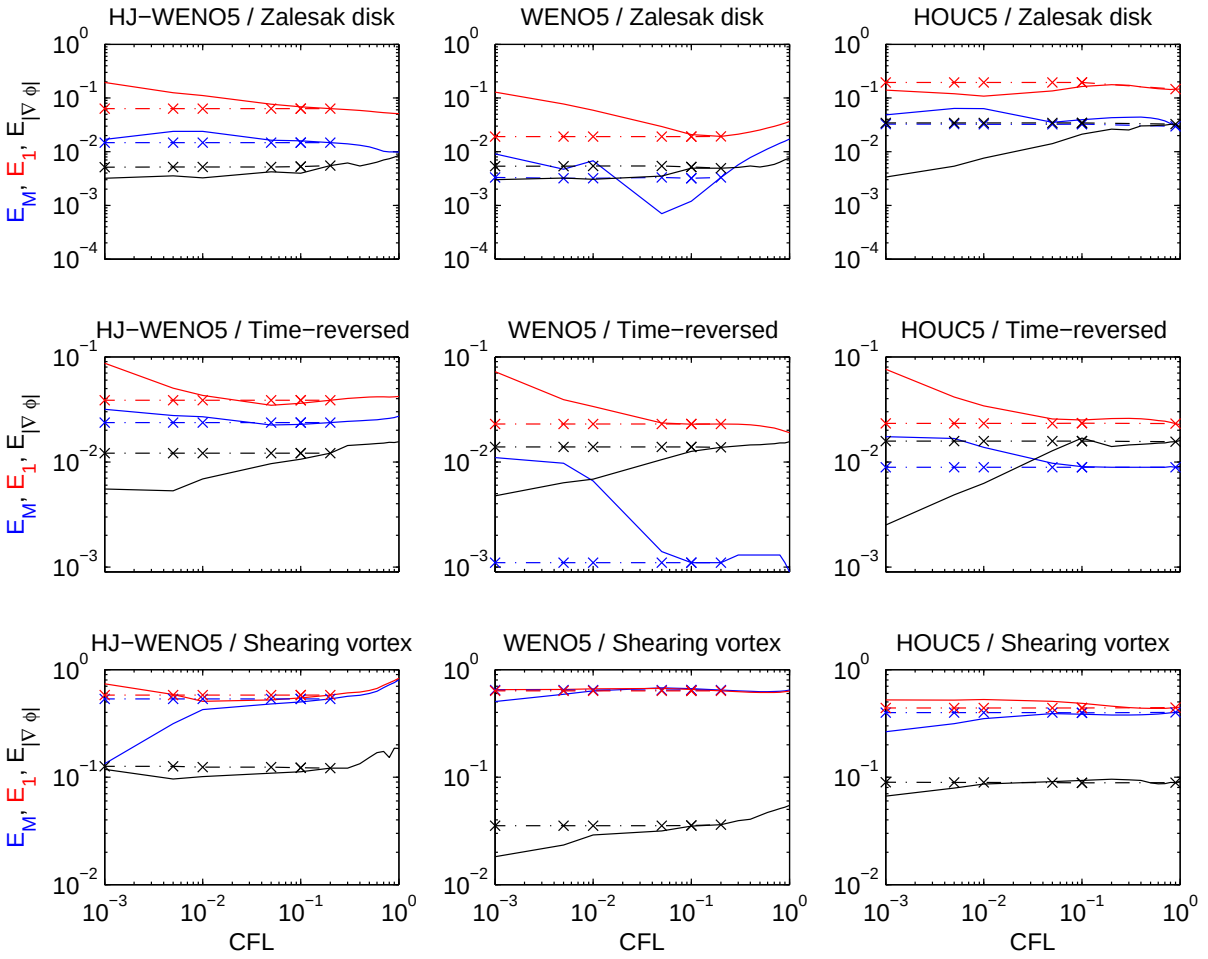

Figure 1.26: Volume constraint: interface errors E_M (Blue), E_1 (Red) and $E_{|\nabla\phi|}$ (Black) for different CFL numbers. Grid : 100^2 for the Zalesak disk and the shearing vortex and 64^2 for the time-reversed single vortex. HJ-ENO2 scheme for the reinitialization. Solid lines : Nreinit = 1 ; dash-dot lines with cross markers: constant CFLreinit (= 0.2 for HJ-WENO5 ; = 0.2 for WENO5 and = 0.9 for HOUC5).

To have an idea of the best values for CFLreinit, Fig. 1.27 plots the interface errors as a function of the CFLreinit for two CFL numbers, 10^{-2} (solid lines) and 10^{-3} (symbols). It is seen that the CFL number has no influence on the accuracy when the CFLreinit is kept constant. In most cases, a large range of suitable values for CFLreinit can be chosen (roughly between 0.01 and 1) without degrading the solution. We remark also that the WENO5 scheme performs again slightly better than the two other presented schemes.

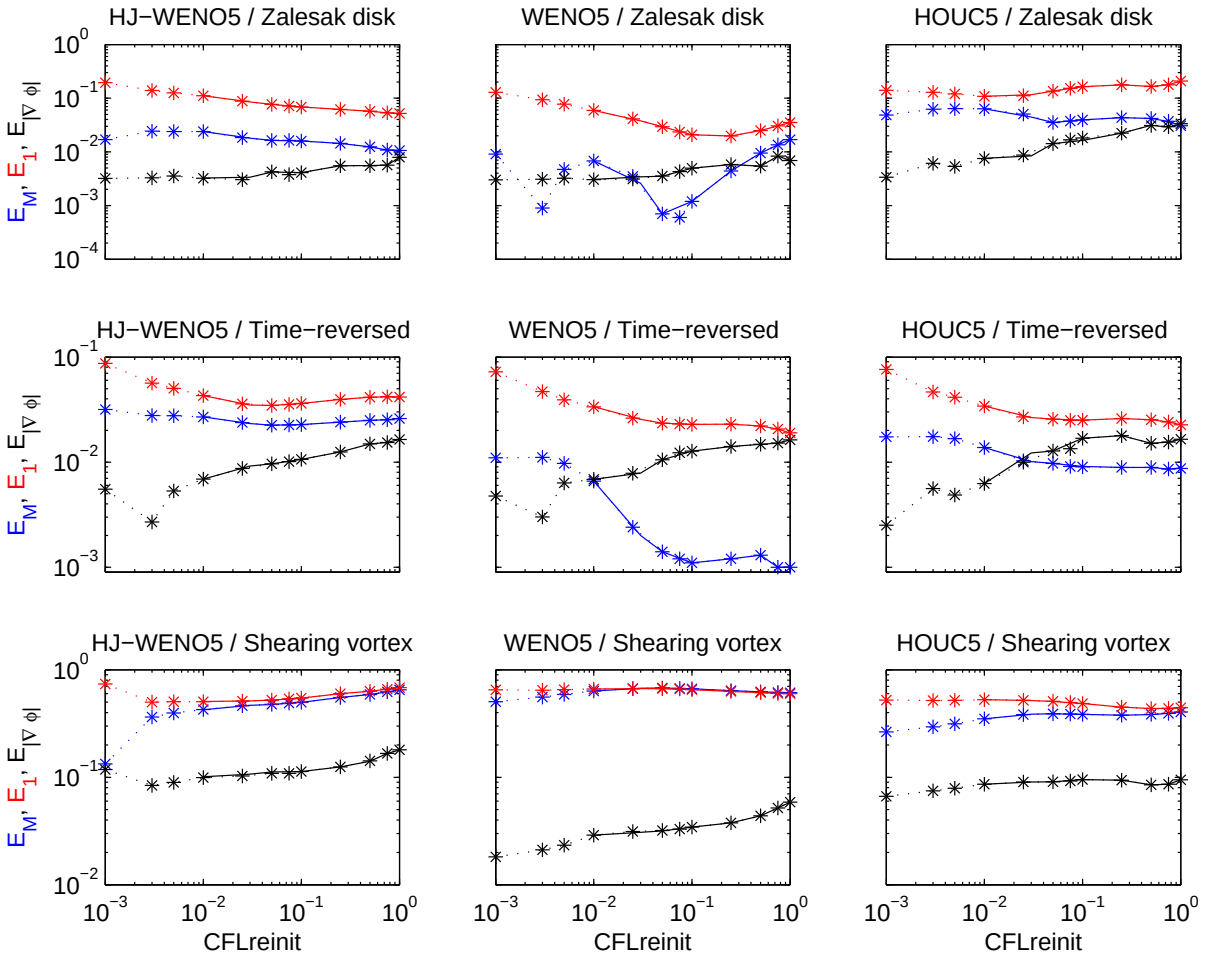

Figure 1.27: Volume constraint: errors E_M (Blue), E_1 (Red) and $E_{|\nabla\phi|}$ (Black) for different CFL numbers.Grid : 100^2 for the Zalesak disk and the shearing vortex and 64^2 for the time-reversed single vortex. HJ-ENO2 scheme for the reinitialization. Solid lines: CFL $= 10^{-2}$; symbols: CFL $= 10^{-3}$.

1.6.3.3 Effect of the reinitialization scheme

We now study the influence of the reinitialization scheme on the level-set method with volume constraint for CFL = CFLreinit = 0.5. The fictitious time for the reinitialization is again $\tau_{\max} = 2\Delta x$ and the fictitious time step $\Delta\tau = \Delta x/2$. We investigate the influence of the reinitialization schemes on the volume constraint method combined with two advection schemes: WENO5 and HJ-WENO5.

Table 1.8 summaries the mass and the positioning errors for the Zalesak disk after a full rotation, the shearing vortex and the time-reversed single vortex at the half period. We remark that the UPWIND1 reinitialization leads to the highest errors, while the remaining schemes give

relatively similar results for the shearing and time-reversed single vortex problems. The HJ-ENO2 and the HJ-WENO3 schemes seem to give the best results and at the same time they are faster than the higher-order schemes (HJ-ENO4, HJ-WENO5).

Table 1.8: Volume constraint: effect of the reinitialization scheme on the HJ-WENO5 and the WENO5 advection. Errors are expressed in %.

Simulation	**Zalesak disk at** $t = 6.28$				**Shearing vortex at** $t = 3$				**Time-reversed at** $t = 1$			
Adv. scheme	**HJ-WENO5**		**WENO5**		**HJ-WENO5**		**WENO5**		**HJ-WENO5**		**WENO5**	
Reinit. scheme	E_M	E_1	E_M	E_1	E_M	E_1	E_M	E_1	E_M	E_1	E_M	E_1
UPWIND1	−10.90	29.48	−6.50	13.32	−64.40	74.44	54.91	59.09	−6.55	8.43	−2.31	4.23
HJ-ENO2	1.21	5.70	−0.84	2.54	−60.00	63.92	62.49	61.34	−2.50	4.14	−0.13	2.21
HJ-ENO3	0.86	5.18	0.08	3.53	−60.29	63.89	63.40	61.59	−2.26	3.96	0.03	2.15
HJ-ENO4	−0.52	5.14	−3.42	5.66	−61.02	64.46	63.42	61.60	−2.40	4.04	−0.03	2.17
HJ-WENO3	1.48	5.48	−0.68	2.42	−59.75	63.70	62.80	61.52	−2.40	4.07	−0.04	2.16
HJ-WENO5	1.20	4.51	0.19	4.08	−59.89	63.41	64.20	62.39	−2.09	3.78	0.16	2.14

1.6.3.4 Spatial convergence rate

We use the HJ-ENO2 for reinitialization and investigate the spatial order of each advection scheme in combination with the volume constraint method by varying the grid size and the CFLreinit number. The positioning errors, E_1, are presented on Fig. 1.28. In contrast to the sub-cell fix method, the volume constraint is less sensitive to the CFLreinit number in the range $[0.1, 1]$. The volume constraint method is almost second-order for all advection schemes for which the order of accuracy is greater than 1. It is also more accurate than the sub-cell fix method.

Fig.1.29 shows the positioning error of the shearing vortex interface for different CFLreinit values. Again, the volume constraint method is less sensitive to the CFLreinit variations. Nevertheless, the errors are of the same order as for the sub-cell fix method. The HOUC schemes also show nearly second-order behavior, although the error levels for all grid sizes are higher than for ENO3, ENO4 and WENO5.

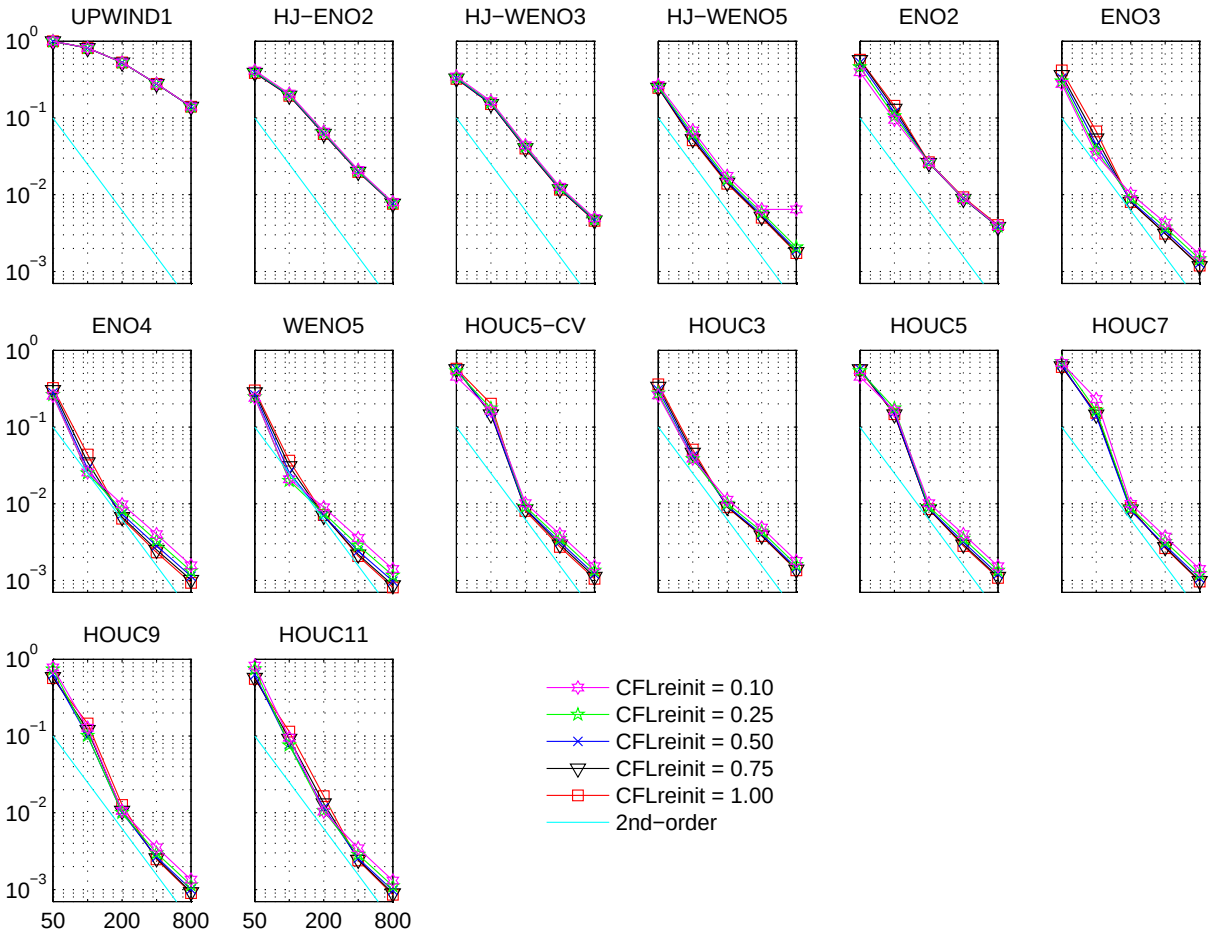

Figure 1.28: Volume constraint: interface positioning error, E_1, of the Zalesak disk after a full rotation for different CFL numbers as a function of the grid size. Nreinit = 1 and HJ-ENO2 scheme for the reinitialization.

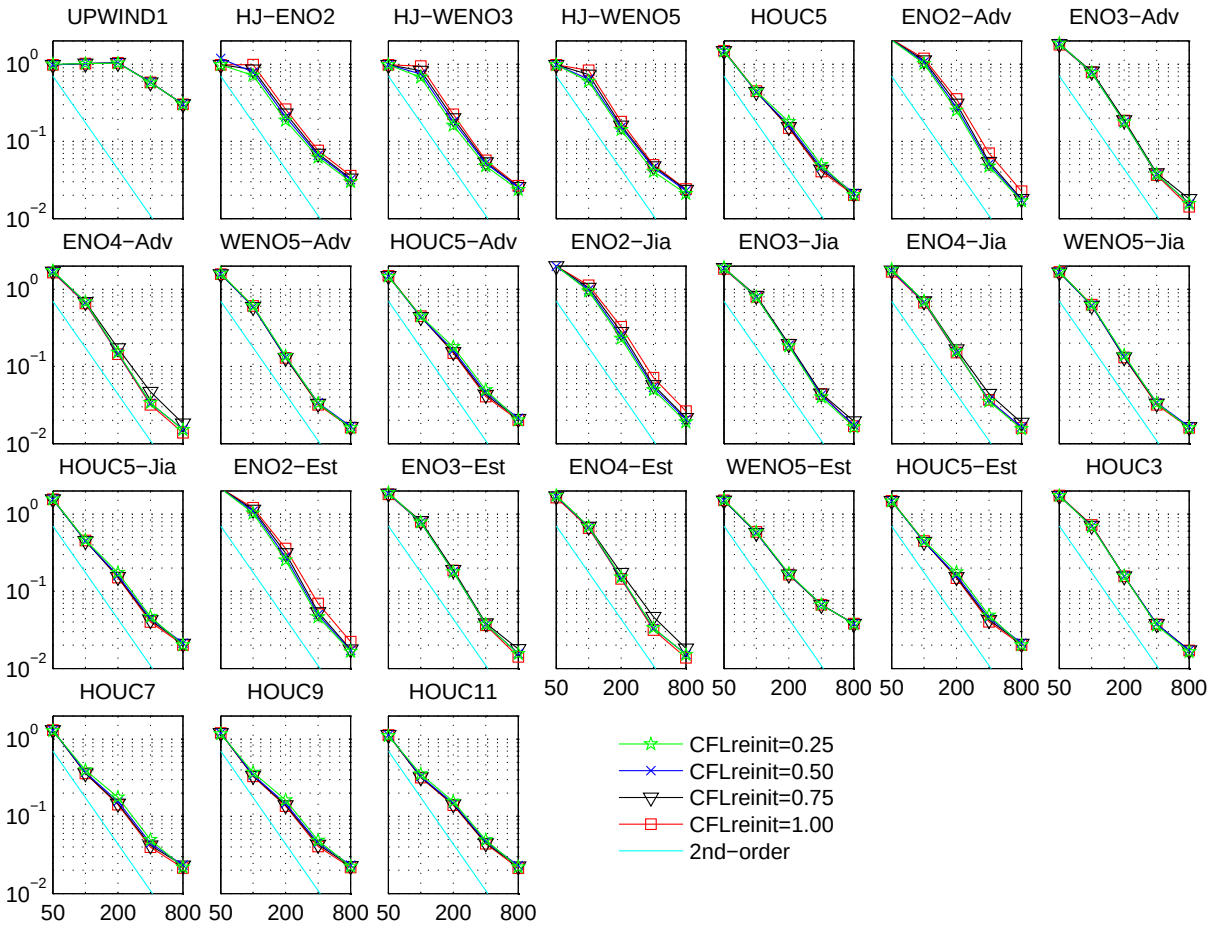

Figure 1.29: Volume constraint: shearing vortex interface positioning error, E_1, at $t = T/2 = 3$ for different CFL numbers as a function of the grid size. Nreinit $= 1$ and HJ-ENO2 scheme for the reinitialization.

1.7 Conclusions

In this chapter, we have investigated the implementation of the level-set method. We started first by a validation of the method with the results provided by Enright et al. (2002); our implementation is slightly more accurate. Then, we compared the level-set with and without reinitialization. The reinitialization disturbs the zero level-set, while it conserves the signed-distance property. When we do not reinitialize the level-set, the interface transport is more accurate for high-order advection schemes, in particular the HOUC schemes. However, without reinitialization the level-set will deviate progressively from the signed-distance function, and eventually become highly inaccurate.

Many attempts to enhance the reinitialization procedure of the level-set method were proposed in the literature. Two popular methods were tested in the current work, namely the sub-cell fix and the volume constraint. The level-set with sub-cell fix seem to be less robust and sensitive to the choice of parameters and schemes. For each given flow problem, we should find the appropriate advection scheme for optimal results. For the Zalesak disk, the fifth-order HJ-WENO scheme is a good choice, but this was not the case for the shearing vortex problem.

The reinitialization with volume constraint is generally more accurate and robust. We have seen that it gives good results in combination with the conservative schemes, in particular the WENO5 scheme, and it is less influenced by variations of the CFL number. Our tests have also shown that the reinitialization scheme has not much influence on the results for both methods. The 2nd-order HJ-ENO scheme combines small interface errors and reduced computational effort with respect to the higher-order reinitialization schemes.

In realistic simulations, the CFL number is probably limited by the physical phenomena taken into account (surface tension, viscous diffusion, etc.). The CFL number required for the stability of the computation may therefore be very small. The performed tests shows that the interface errors may increase drastically for very small CFL numbers, since in this case the interface is advected less and reinitialized more often. The current results show a relevant relation between the frequency of the reinitialization and the interface errors. This led to the introduction of the CFLreinit number, which is the product of the CFL number and the number of advection steps to perform before each reinitialization of the level-set. Keeping the CFLreinit constant leads almost to constant errors for all CFL numbers (provided CFL$\leqslant 1$). Results have shown that the sub-cell fix method is nonetheless more sensitive to the CFLreinit number variations; it diverges when CFLreinit is smaller. The volume constraint method is less influenced by CFLreinit variations for values in the range $[0.1 - 1.0]$.

The CPU-time has not been taken into account in this study, which has been focused mainly on the identification of the level-set method which combines both accuracy and robustness. On the one hand, in realistic simulations the level-set method will run in parallel with the flow solver, which will certainly be more CPU-time consuming. On the other hand, a local resolution of the level-set has been implemented in our code (see §1.4) and can be applied in order to reduce the computational time. It has not been used in the results shown in this chapter, but when used it roughly gives the same results.

In the light of the presented results on this chapter, the level-set with volume constraint seems to be a good choice. In addition, we will use in the reminder of this work:

- the 5th-order WENO-Adv scheme for advection
- reinitialization after each advection step (Nreinit = 1) using the 2nd-order HJ-ENO scheme.

This choice offers the advantage of being robust (works fine on all the performed simulations) and less sensitive to the CFL number.

Chapter 2

Volume-Of-Fluid methods

2.1 Introduction

The Volume-Of-Fluid method (VOF) is an Eulerian method for capturing the interfaces separating immiscible fluids. Let f be the characteristic function with value 1 in the reference fluid and zero elsewhere. The volume-of-fluid (VOF) function C_i is then defined as the volume fraction of the reference fluid in the grid cell Ω_i:

$$C_i = \frac{1}{V_i} \int_{\Omega_i} f dv \tag{2.1}$$

where $V_i = \text{vol}(\Omega_i)$ is the volume of the grid cell. In a flow without phase change, the phase indicator function f is constant on the trajectory of a fluid particle (in a Lagrangian reference system) so that we can write

$$\frac{Df}{Dt} = \frac{\partial f}{\partial t} + \vec{u} \cdot \nabla f = 0$$

In incompressible flows, we have also $\nabla \cdot \vec{u} = 0$ so that we can write

$$\frac{\partial f}{\partial t} + \vec{u} \cdot \nabla f + f \nabla \cdot \vec{u} = \frac{\partial f}{\partial t} + \nabla \cdot (f\vec{u}) = 0 \tag{2.2}$$

Combining Eqs.(2.1) and (2.2) and making use of the divergence theorem, we can write an evolution equation for C:

$$\frac{\partial C_i}{\partial t} + \frac{1}{V_i} \int_{\Omega_i} \nabla \cdot (f\vec{u})\, dv = \frac{\partial C_i}{\partial t} + \frac{1}{V_i} \int_{\partial\Omega_i} f\vec{u} \cdot \vec{n}\, dS = 0 \tag{2.3}$$

where $\partial\Omega_i$ is the boundary of the grid cell Ω_i and $\vec{n}$ is the unit normal vector on $\partial\Omega_i$ pointing outwards of Ω_i. The volume-of-fluid is therefore a conserved quantity in an incompressible fluid. In the literature, Eq. (2.3) is often expressed simply as an advection equation:

$$\frac{\partial C_i}{\partial t} + \nabla \cdot (\vec{u}C) = 0 \tag{2.4}$$

Two different approaches are proposed in the literature in order to compute the evolution of $C_i(t)$. The first one consists of numerically solving the advection equation (2.4) using the so-called compressive scheme (Weller, 1993; Ubbink, 1997; Ubbink and Issa, 1999; Weller, 2008; Denner et al., 2014b), which are based on the donor-acceptor scheme by Hirt and Nichols (1981) and does not necessitate interface reconstruction in the advection process. The second one uses

a geometrical evaluation of the fluxes in Eq. (2.3) through the cell faces (Youngs, 1982; Pilliod, 1992; Rider, 1995; Rider and Kothe, 1998; Scardovelli and Zaleski, 2003), which indeed needs the knowledge of the interface form. The use of the first or the second approach depends strongly on the considered grid type. If the grid is cartesian and uniform, the geometrical resolution is the most suited since it is relatively more accurate. On the counterpart, the compressive schemes could be used for arbitrary mesh (Ubbink and Issa, 1999) and is more easily implemented in existing codes with unstructured grids.

In the current work, the geometrical approach is adopted. Therefore, in order to compute the evolution of $C_i(t)$ we need to reconstruct the scalar field of f in the neighborhood of the grid cell which serves to compute the numerical flux $f\vec{u}\cdot\vec{n}$ on the boundary. Volume-of-fluid methods based on the geometrical approach are therefore always build as a two-step procedure. In the so-called reconstruction step, a subset of $C_{i'}$ in the neighborhood of Ω_i is used to estimate the phase indicator f in this grid cell. From this representation of the interface, Eq. (2.3) is solved by evaluating the fluxes in a geometrical way.

2.2 Interface reconstruction

The volume-of-fluid (VOF) method was originally developed in the 1970s. Rider and Kothe (1998) have presented a historical overview of the earlier papers on VOF. It was soon discovered that in order to preserve a sharp interface, standard finite-volume upwind methods are not well suited since they tend to introduce smearing of the interface. Instead, a geometrical reconstruction of the interface followed by an advection step which preserves the discontinuous fluid properties seems to be a much better approach. The most simple geometrical reconstruction method is the simple line interface calculation (SLIC) method (Noh and Woodward, 1976), which describes the interface as a horizontal or vertical line cutting the grid cell in two parts. Although very simple to implement, the method is not very accurate and the interface reconstruction is not smooth. For a more realistic description of the interface, the piece-wise linear interface calculation (PLIC) method was proposed (Youngs, 1982). The interface is still described by a straight line (or plane surface in 3D), but the normal of the interface is not necessarily aligned with the vertical or horizontal axis but can take on any direction. In PLIC methods, the interface described by a plane surface is most easily defined in the Hessian normal form

$$\vec{x}\cdot\vec{n}+d=0$$

where $\vec{n}$ is the unit normal vector. The parameter d represents the distance of the interface to the origin which is chosen here arbitrary as the center of the grid cell under consideration. Once the normal vector $\vec{n}$ is known, the distance d follows from the consideration that the volume fraction defined by the interface reconstruction equals the target value (see §A.3.2), for example in 2D:

$$C_{i,j}=\frac{1}{V_{i,j}}\int_{\Omega_{i,j}} H(\vec{x}\cdot\vec{n}_{i,j}+d_{i,j})\,dv$$

PLIC methods differ in the way the normal vector is defined (see for example Scardovelli and Zaleski (2003) and Pilliod and Puckett (2004)). In the least-squares volume-of-fluid interface reconstruction algorithm (LVIRA) by Puckett (1991), the linear interface reconstruction in a grid cell is extended to the neighboring grid cells (as shown by the dashed line on Fig. 2.1). The difference between the reconstructed and actual volume fraction is then minimized in the least-squares sense by adjusting the normal to the interface. In 2D, the least-squares error is

defined as:

$$E_{i,j} = \sum_{k=i-1}^{i+1} \sum_{l=j-1}^{j+1} \left[\widehat{C}_{k,l}(\vec{n}_{i,j}) - C_{i,j} \right]^2 \tag{2.5}$$

where $\widehat{C}_{k,l}$ is the reconstructed volume fraction, which depends on the orientation of the interface defined through $\vec{n}$:

$$\widehat{C}_{k,l} = \frac{1}{V_{k,l}} \int_{\Omega_{k,l}} f(\vec{x} \cdot \vec{n}_{i,j} + d_{i,j}) \, dv$$

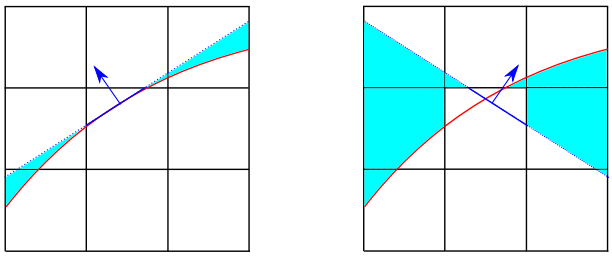

Figure 2.1: Illustration of the least-squares error (which is the sum of the squares of the coloured area in each cell) in a 3×3 block around a central cell in the LVIRA and the ELVIRA techniques. The red line represents the desired (or exact) interface and the blue solid line the interface inside the central cell. the blue dashed line represents the extension of the normal in the 3×3 block of cells.

Although this method has shown to be second-order accurate for smooth surfaces, its main drawback is that the optimization algorithm tends to be time-consuming, especially in 3D. The efficient least-squares volume-of-fluid interface reconstruction algorithm (ELVIRA) by Pilliod (1992) reduces the costs by limiting the optimization procedure to a set of 6 candidate normals in 2D (9 in 3D) constructed from simple finite-differencing. The first three candidates are constructed from backward, central and forward finite-difference approximations in the x-direction:

$$\left.\begin{aligned} \widehat{n}_x^b &= \sum_{l=j-1}^{j+1} (C_{i,l} - C_{i-1,l}) \\ \widehat{n}_x^c &= \frac{1}{2} \sum_{l=j-1}^{j+1} (C_{i+1,l} - C_{i-1,l}) \\ \widehat{n}_x^f &= \sum_{l=j-1}^{j+1} (C_{i+1,l} - C_{i,l}) \end{aligned}\right\} \text{with } \widehat{n}_y = 1.$$

Another 3 candidates are obtained by differentiation in the y-direction:

$$\left.\begin{aligned} \widehat{n}_y^b &= \sum_{l=j-1}^{j+1} (C_{k,j} - C_{k,j-1}) \\ \widehat{n}_y^c &= \frac{1}{2} \sum_{l=j-1}^{j+1} (C_{k,j+1} - C_{k,j-1}) \\ \widehat{n}_y^f &= \sum_{l=j-1}^{j+1} (C_{k,j+1} - C_{k,j}) \end{aligned}\right\} \text{with } \widehat{n}_x = 1.$$

The normal vector is then selected among these 6 candidates as the one that gives the smallest value for the least-squares error, $E_{i,j}$. We use the ELVRA reconstruction except otherwise

specified.

Many other reconstruction techniques were proposed in literature, that we do not implement in the current work, among which we cite

- The parabolic reconstruction of the interface developed by Renardy and Renardy (2002)
- The linear and quadratic least-squares fits of the interface in 2D (Scardovelli and Zaleski, 2003) which are extended to three-dimensions by Aulisa et al. (2007). This fitting is slightly more accurate than the ELVIRA, however it needs more computational resources.
- Later, Vignesh and Bakshi (2013) suggested to reconstruct a continuous interface from the knowledge of one endpoint of the interface which is determined from the already reconstructed interface on the neighboring cell.

A comparison of the earlier reconstruction methods can be found in Rider and Kothe (1998), Scardovelli and Zaleski (2003) and Pilliod and Puckett (2004).

2.3 Interface advection

Although the transport equation (2.3) is conservative by construction, a numerical scheme that conserves the total volume of the reference fluid is not so easily obtained. The main problem resides in the fact that the volume fraction C is limited between 0 and 1. Many standard finite-volume schemes will cause small over- or undershoots of C, and subsequent clipping of the volume fractions will inevitably lead to loss of the conservation properties. This section describes several geometrical methods developed in the literature that are both consistent ($0 \leq C \leq 1$) and conservative.

2.3.1 Conservative operator splitting

We will start by describing a splitting method that allows us to solve the transport equation independently for each spatial direction. A conservative split method (Puckett et al., 1997) is given on a 2D Cartesian grid by:

$$C^*_{i,j} = \frac{C^n_{i,j} + (\delta t/\Delta x)\left(\widehat{F}_{i+1/2,j} - \widehat{F}_{i-1/2,j}\right)}{1 - (\delta t/\Delta x)\left(u_{i+1/2,j} - u_{i-1/2,j}\right)} \tag{2.6}$$

$$C^{n+1}_{i,j} = C^*_{i,j}\left[1 + \frac{\delta t}{\Delta y}\left(v_{i,j+1/2} - v_{i,j-1/2}\right)\right] + \frac{\delta t}{\Delta y}\left(\widehat{G}_{i,j+1/2} - \widehat{G}_{i,j-1/2}\right) \tag{2.7}$$

where $\widehat{F}$ and $\widehat{G}$ are the mass fluxes across the cell faces in the x- and y-direction. Note that this is a numerical approximation of the following advection equation:

$$\frac{\partial C_i}{\partial t} + \nabla \cdot (\vec{u}C) = C(\nabla \cdot \vec{u})$$

rather than the conservation equation (2.4). This scheme will nonetheless conserve the mass if the flow field is incompressible, i.e.

$$(u_{i+1/2,j} - u_{i-1/2,j})/\Delta x + (v_{i,j+1/2} - v_{i,j-1/2})/\Delta y = 0$$

which can easily be shown by inserting Eq. (2.6) into Eq. (2.7).

However, despite the conservation of mass, it is still possible to obtain non-physical values for $C^{n+1}_{i,j}$, below zero or above one. In order to maintain $C^{n+1}_{i,j}$ within the interval $[0, 1]$, it is necessary

that the computation of the numerical fluxes $\widehat{F}_{i+1/2,j}$ and $\widehat{G}_{i,j+1/2}$ are based on a geometrical approach, as explained in detail in §2.3.2 and §2.3.3. This operator splitting scheme is made second-order in time by using Strang-splitting. The simplest implementation is to alternate the order of spatial directions on subsequent time steps.

Sussman and Puckett (2000) have extended this scheme to 2D cylindrical coordinates and 3D cartesian coordinate systems. For cylindrical coordinates, we may use:

$$C^*_{i,j} = \frac{C^n_{i,j} + (\delta t/r_i \Delta r)\left(r_{i+1/2}\widehat{F}_{i+1/2,j} - r_{i-1/2}\widehat{F}_{i-1/2,j}\right)}{1 - (\delta t/r_i \Delta r)\left(r_{i+1/2}u_{i+1/2,j} - r_{i-1/2}u_{i-1/2,j}\right)} \tag{2.8}$$

$$C^{n+1}_{i,j} = C^*_{i,j}\left[1 + \frac{\delta t}{\Delta y}\left(v_{i,j+1/2} - v_{i,j-1/2}\right)\right] + \frac{\delta t}{\Delta y}\left(\widehat{G}_{i,j+1/2} - \widehat{G}_{i,j-1/2}\right) \tag{2.9}$$

and finally for 3D equations:

$$C^*_{i,j,k} = \frac{C^n_{i,j,k} + (\delta t/\Delta x)\left(\widehat{F}_{i+1/2,j,k} - \widehat{F}_{i-1/2,j,k}\right)}{1 - (\delta t/\Delta x)\left(u_{i+1/2,j,k} - u_{i-1/2,j,k}\right)} \tag{2.10}$$

$$C^{**}_{i,j,k} = \frac{C^*_{i,j,k} + (\delta t/\Delta y)\left(\widehat{G}_{i,j+1/2,k} - \widehat{G}_{i,j-1/2,k}\right)}{1 - (\delta t/\Delta y)\left(v_{i,j+1/2,k} - v_{i,j-1/2,k}\right)} \tag{2.11}$$

$$C^{***}_{i,j,k} = \frac{C^{**}_{i,j,k} + (\delta t/\Delta z)\left(\widehat{H}_{i,j,k+1/2} - \widehat{H}_{i,j,k-1/2}\right)}{1 - (\delta t/\Delta z)\left(w_{i,j,k+1/2} - w_{i,j,k-1/2}\right)} \tag{2.12}$$

$$\begin{aligned} C^{n+1}_{i,j,k} =& C^{***}_{i,j,k} - \delta t\left[\frac{C^*_{i,j,k}}{\Delta x}\left(u_{i+1/2,j,k} - u_{i-1/2,j,k}\right)\right. \\ &\left. + \frac{C^{**}_{i,j,k}}{\Delta y}\left(v_{i,j+1/2,k} - v_{i,j-1/2,k}\right) + \frac{C^{***}_{i,j,k}}{\Delta z}\left(w_{i,j,k+1/2} - w_{i,j,k-1/2}\right)\right] \end{aligned} \tag{2.13}$$

To show that this scheme preserves the total mass, one needs to substitute the expression of $C^*_{i,j,k}$, $C^{**}_{i,j,k}$ and $C^{***}_{i,j,k}$ given by the Eqs. (2.10-2.12) in Eq. (2.13), then to sum over all the computational cells. This does not, however, guarantee that $C^{n+1}_{i,j,k}$ is within the interval $[0, 1]$.

2.3.2 One-dimensional explicit Lagrangian scheme

This section describes how the numerical flux is computed in the second step Eq. (2.7) of the 2D directional split scheme. We will first focus on the one-dimensional Eulerian cell $\mathcal{E}_i$ defined by the interval $[x_{i-1/2}, x_{i+1/2}]$, as proposed by Tryggvason et al. (2011). Consider the explicit Lagrangian advection formula for a point x given by

$$x^{n+1} = x^n + u(x^n)\Delta t$$

The velocity is now described by the linear interpolation formula:

$$u(x) = u_{i-1/2} + (u_{i+1/2} - u_{i-1/2})(x - x_{i-1/2})/\Delta x \tag{2.14}$$

The advection equation describes how each fluid particle in the Eulerian cell $\mathcal{E}_i$ is moved from t^n to t^{n+1}. The set of particles positions x^{n+1} defines a new interval called the Lagrangian cell $\mathcal{L}_i$. From the above formulas, we can easily calculate the locations of the boundaries of the

Lagrangian cell, noted $x_{i+1/2}^{\mathcal{L},n+1}$:

$$x_{i+1/2}^{\mathcal{L},n+1} = x_{i+1/2} + u_{i+1/2}\Delta t$$

The compression (or extension) coefficient of the cell, defined as the size ratio of the Lagrangian and Eulerian cells, is now given by

$$a_i = \frac{x_{i+1/2}^{\mathcal{L},n+1} - x_{i-1/2}^{\mathcal{L},n+1}}{\Delta x} = 1 + \frac{(u_{i+1/2} - u_{i-1/2})\Delta t}{\Delta x} \tag{2.15}$$

where $\Delta x = x_{i+1/2} - x_{i-1/2}$. It is also possible to trace backward the initial locations of the fluid particles $x_{i+1/2}^{\mathcal{E},n}$ which end up at the Eulerian boundary $x_{i+1/2}$ at t^{n+1}:

$$x_{i+1/2} = x_{i+1/2}^{\mathcal{E},n} + u\left(x_{i+1/2}^{\mathcal{E},n}\right)\Delta t$$

Supposing that $u_{i+1/2}$ is positive as shown in Fig. 2.2, we obtain

$$x_{i+1/2}^{\mathcal{E},n} = \left(x_{i+1/2} - u_{i+1/2}\Delta t\right)/a_i$$

where a_i is the compression coefficients for the donor cell $\mathcal{E}_i$. If the sign of $u_{i+1/2}$ is negative, the indices of a should be changed so that it always points to the donor cell i.e. $\mathcal{E}_{i+1}$. It is therefore convenient to define

$$a_{i+1/2} = \begin{cases} a_i & \text{if} \quad u_{i+1/2} \geq 0 \\ a_{i+1} & \text{if} \quad u_{i+1/2} \leq 0 \end{cases} \tag{2.16}$$

In Fig. 2.2, the dashed zones indicate the volume that is fluxed through the cell boundaries. The numerical flux $F_{i+1/2}$ is therefore defined as

$$F_{i+1/2} = \frac{S\left(u_{i+1/2}\right)}{\Delta x \Delta y} \int_{y_{j-1/2}}^{y_{j+1/2}} \int_{x_{i+1/2}-\max\left(0,u_{i+1/2}\right)\Delta t/a_i}^{x_{i+1/2}-\min\left(0,u_{i+1/2}\right)\Delta t/a_{i+1}} f\left(x,y\right)\ dx\ dy \tag{2.17}$$

where $f(x,y)$ is the phase indicator function obtained from the reconstructed interface indicated by the red line in Fig. 2.2 and $S()$ is the sign function. The integration limits are written in a way they work for both positive and negative values of $u_{i+1/2}$.

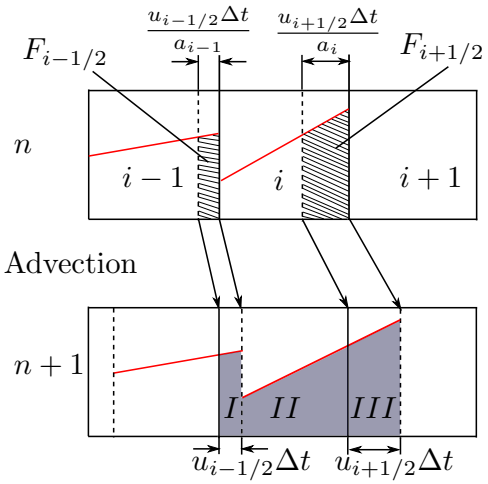

Figure 2.2: Illustration of the one-dimensional explicit Lagrangian time scheme.

The volume fraction C_i^{n+1} is defined as the fraction of the reference fluid in the Eulerian cell $\mathcal{E}_i$ at t^{n+1} and is obtained in a geometrical interpretation by projecting the Lagrangian cells onto the Eulerian grid. In Fig. 2.2, for example, C_i^{n+1} is defined as the area of the two grayed surfaces (I) and (II), whereas the area (III) contributes to the fraction C_{i+1}^{n+1} of the neighboring cell $\mathcal{E}_{i+1}$.

The one-dimensional explicit Lagrangian scheme can also be written as an algebraic equation as:

$$C_i^{n+1} = a_i C_i^n - a_{i+1/2} F_{i+1/2} + a_{i-1/2} F_{i-1/2} \tag{2.18}$$

where the first term on the right-hand side corresponds to the sum of the areas (II) and (III), the second term subtracts the area (III) and the third term is equal to the area (I). If $u_{i+1/2}$ or $u_{i-1/2}$ changes sign, the gray areas in Fig. 2.2 should be adapted accordingly, but Eq. (2.18) remains valid. Note that the multiplication factor $a_{i+1/2}$ is indexed according to the sign of $u_{i+1/2}$ and not to the cell under consideration (it is attached to the cell face, like the numerical flux $F_{i+1/2}$), it is given by Eq. (2.16). We can therefore write:

$$\sum_{i=1}^{N} C_i^{n+1} = \sum_{i=1}^{N} a_i C_i^n - a_N F_{N+1/2} + a_0 F_{1/2}$$

On its own, this scheme is not conservative, due to the coefficients a_i in the first term, but we see that the fluxes at the inner cell boundaries cancel out, in a manner similar to classical finite volume scheme. We will see in §2.3.4 that the combination with the implicit Lagrangian scheme described in §2.3.3 will make the splitting method conservative. It is important to note here that the present scheme is consistent due to the geometrical reconstruction of the reference fluid in the grid cell at the next time step.

Finally, Eq. (2.18) can also be written as

$$\frac{C_i^{n+1} - C_i^n}{\Delta t} + \frac{u_{i+1/2}\widehat{C}_{i+1/2} - u_{i-1/2}\widehat{C}_{i-1/2}}{\Delta x} = C_i^n \frac{u_{i+1/2} - u_{i-1/2}}{\Delta x} \tag{2.19}$$

if we define $\widehat{C}_{i+1/2} = a_{i+1/2} F_{i+1/2} \Delta x / (u_{i+1/2} \Delta t)$. This equation can be seen as a discrete approximation of the advection equation $\partial C/\partial t + \partial(uC)/\partial x = C\partial u/\partial x$ with the right-hand side term in the explicit formulation (with respect to C).

2.3.3 One-dimensional implicit Lagrangian scheme

This section describes how the numerical flux is computed in the first step, Eq. (2.6), of the 2D directional split scheme. We will again focus on the Eulerian cell $\mathcal{E}_i$. Consider now the implicit Lagrangian advection formula given by

$$x^{n+1} = x^n + u(x^{n+1})\Delta t \tag{2.20}$$

The velocity is again described by the linear interpolation, Eq. (2.14). The advection equation defines the Lagrangian cell $\mathcal{L}_i$ with boundaries $x_{i+1/2}^{\mathcal{L},n+1}$ given by:

$$x_{i+1/2}^{\mathcal{L},n+1} = x_{i+1/2} + u\left(x_{i+1/2}^{\mathcal{L},n+1}\right)\Delta t$$

Supposing again that $u_{i+1/2}$ is positive, the combination of Eqs. (2.14) and (2.20) leads to the following expressions:

$$x_{i+1/2}^{\mathcal{L},n+1} = x_{i+1/2} + b_{i+1} u_{i+1/2} \Delta t$$

where b_i is the compression (or extension) ratio of the cell given by

$$b_i = \frac{1}{1 - \left(u_{i+1/2} - u_{i-1/2}\right) \Delta t / \Delta x} \tag{2.21}$$

The index of b_i refers to the receiving cell and should be modified when $u_{i+1/2}$ changes sign. Tracing backwards the location of the cell boundary is relatively simple and gives:

$$x_{i+1/2} = x_{i+1/2}^{\mathcal{E},n+1} + u_{i+1/2}\Delta t$$

Fig. 2.3 illustrates the advection of cell boundaries in the case both $u_{i+1/2}$ and $u_{i-1/2}$ are positive. The dashed zones indicate the volume that is fluxed at the cell boundaries, $F_{i-1/2}$ and $F_{i+1/2}$, still defined by

$$F_{i+1/2,j} = \frac{S\left(u_{i+1/2}\right)}{\Delta x \Delta y} \int_{y_{j-1/2}}^{y_{j+1/2}} \int_{x_{i+1/2}-\min\left(0,u_{i+1/2}\right)\Delta t}^{x_{i+1/2}-\max\left(0,u_{i+1/2}\right)\Delta t} f\left(x,y\right)\ dx\ dy \tag{2.22}$$

The volume fraction C^{n+1} is defined as the area of the two grayed surfaces (I) and (II), whereas the area (III) contributes to the fraction C_{i+1}^{n+1} of the neighboring cell $\mathcal{E}_{i+1}$.

The equivalent algebraic equation for the one-dimensional implicit Lagrangian scheme is:

$$C_i^{n+1} = b_i \left(C_i^n - F_{i+1/2} + F_{i-1/2}\right) \tag{2.23}$$

where the first term on the right-hand side corresponds to the area (II) and the second term is equal to the area (I). Note that the multiplication factors b_i for the numerical fluxes $F_{i-1/2}$ and $F_{i+1/2}$ are indexed according to the cell under consideration. Therefore, the numerical fluxes do not cancel out if we sum up the values of C_i^{n+1} for $i = 1, \cdots, N$.

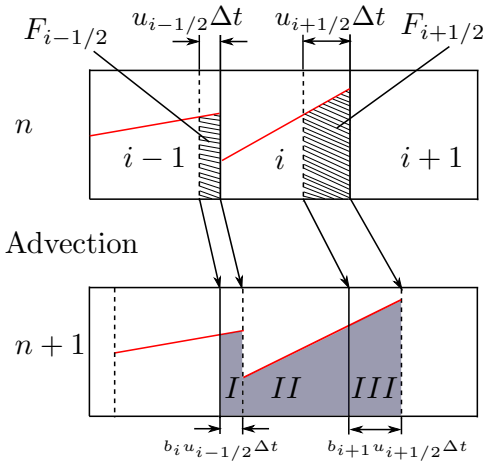

Figure 2.3: Illustration of the one-dimensional implicit Lagrangian time scheme.

Finally, Eq. (2.23) can also be written as

$$\frac{C_i^{n+1} - C_i^n}{\Delta t} + \frac{u_{i+1/2}\widehat{C}_{i+1/2} - u_{i-1/2}\widehat{C}_{i-1/2}}{\Delta x} = C_i^{n+1}\frac{u_{i+1/2} - u_{i-1/2}}{\Delta x} \tag{2.24}$$

if we define $\widehat{C}_{i+1/2} = F_{i+1/2}\Delta x/(u_{i+1/2}\Delta t)$. This equation can be seen as a discrete approximation of the advection equation $\partial C/\partial t + \partial(uC)/\partial x = C\partial u/\partial x$ with the right-hand side term in

implicit form with respect to C.

2.3.4 Conservative and consistent flux splitting

It is clear that the geometrical construction of the volume fraction always leads to consistent values of C: the cell is either empty, $C = 0$, completely filled, $C = 1$, or partially filled, $0 < C < 1$. This can only be obtained by a geometrical construction in which the set of Lagrangian cells do not overlap (with areas counted twice) and do not present gaps, as is the case for the explicit and implicit schemes described above. Both schemes are not conservative, but for a two-dimensional incompressible flow, the application of the implicit scheme in one direction, followed by the explicit scheme in the other direction leads to the desired conservation properties. Let us take the x-direction for the implicit scheme, so that we can write the following equation for the provisional volume fraction $C^*_{i,j}$:

$$C^*_{i,j} = b_{i,j}\left(C^n_{i,j} - F_{i+1/2,j} + F_{i-1/2,j}\right) \tag{2.25}$$

where $F_{i+1/2,j}$ is defined by Eq. (2.22). The explicit scheme is used for the y-direction:

$$C^{n+1}_{i,j} = a_{i,j}C^*_{i,j} - a_{i,j+1/2}F^*_{i,j+1/2} + a_{i,j-1/2}F^*_{i,j-1/2} \tag{2.26}$$

where $F_{i,j+1/2}$ is defined by Eq. (2.17), with f computed from the interface reconstruction based on $C^*_{i,j}$. The coefficient $a_{i,j+1/2}$ is defined by $a_{i,j}$ or $a_{i,j+1}$ depending on whether $v_{i,j+1/2}$ is positive or negative.

The coefficients $a_{i,j}$ and $b_{i,j}$ are defined by

$$\begin{aligned} a_{i,j} &= 1 + \left(v_{i,j+1/2} - v_{i,j-1/2}\right)\Delta t/\Delta y \\ b_{i,j} &= \left[1 - \left(u_{i+1/2,j} - u_{i-1/2,j}\right)\Delta t/\Delta x\right]^{-1} \end{aligned}$$

and due to the incompressibility condition,

$$\frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta y} = 0$$

the product $a_{i,j}b_{i,j}$ is equal to unity,

$$a_{i,j}b_{i,j} = \frac{1 + \left(v_{i,j+1/2} - v_{i,j-1/2}\right)\Delta t/\Delta y}{1 - \left(u_{i+1/2,j} - u_{i-1/2,j}\right)\Delta t/\Delta x} = 1$$

The combined implicit/explicit flux splitting scheme finally leads to the following global equation:

$$\begin{aligned} C^{n+1}_{i,j} &= a_{i,j}b_{i,j}\left(C^n_{i,j} - F_{i+1/2,j} + F_{i-1/2,j}\right) - a_{i,j+1/2}F_{i,j+1/2} + a_{i,j-1/2}F_{i,j-1/2} \\ &= C^n_{i,j} - F_{i+1/2,j} + F_{i-1/2,j} - a_{i,j+1/2}F_{i,j+1/2} + a_{i,j-1/2}F_{i,j-1/2} \end{aligned}$$

It is already shown that the fluxes of the explicit scheme cancel out on inner cell boundaries. Due to the incompressibility condition, the fluxes in the x-direction now also cancel out and we can write:

$$\sum_{i=1}^{N}\sum_{j=1}^{M} C^{n+1}_{i,j} = \sum_{i=1}^{N}\sum_{j=1}^{M} C^n_{i,j} + \sum_{j=1}^{M}\left(F_{1/2,j} - F_{N+1/2,j}\right) + \sum_{i=1}^{N}\left(a_{i,1/2}F_{i,1/2} + a_{i,M+1/2}F_{i,M+1/2}\right)$$

This flux-splitting scheme is therefore both conservative and consistent. To obtain second-order

accuracy in time, Strang-splitting should be used which consists of alternating the x- and y-directions for successive time steps. This also avoids the development of flow structures in privileged directions due to the use of different schemes in each direction.

Finally, the conservation properties can also be evaluated from the following equivalent discrete approximations of the advection equations:

$$\frac{C^*_{i,j} - C^n_{i,j}}{\Delta t} + \frac{u_{i+1/2,j}\widehat{C}_{i+1/2,j} - u_{i-1/2,j}\widehat{C}_{i-1/2,j}}{\Delta x} = C^*_{i,j}\frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} \tag{2.27}$$

$$\frac{C^{n+1}_{i,j} - C^*_{i,j}}{\Delta t} + \frac{v_{i,j+1/2}\widehat{C}^*_{i,j+1/2} - v_{i,j-1/2}\widehat{C}^*_{i,j-1/2}}{\Delta y} = C^*_{i,j}\frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta y} \tag{2.28}$$

It is now easily seen that due to the combination of the implicit and explicit treatment of the compression terms on the right-hand side, they cancel out in an incompressible flow. The global equation reads:

$$\frac{C^{n+1}_{i,j} - C^n_{i,j}}{\Delta t} + \frac{u_{i+1/2,j}\widehat{C}_{i+1/2,j} - u_{i-1/2,j}\widehat{C}_{i-1/2,j}}{\Delta x} + \frac{v_{i,j+1/2}\widehat{C}^*_{i,j+1/2} - v_{i,j-1/2}\widehat{C}^*_{i,j-1/2}}{\Delta y} = 0$$

and is in conservative form.

2.3.5 An alternative simple conservative and consistent flux splitting

A different simpler conservative and consistent flux splitting, working in both two and three dimensions, is recently proposed by Weymouth and Yue (2010).

$$\frac{C^*_{i,j} - C^n_{i,j}}{\Delta t} + \frac{u_{i+1/2,j}\widehat{C}_{i+1/2,j} - u_{i-1/2,j}\widehat{C}_{i-1/2,j}}{\Delta x} = f_{i,j}\frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} \tag{2.29}$$

$$\frac{C^{n+1}_{i,j} - C^*_{i,j}}{\Delta t} + \frac{v_{i,j+1/2}\widehat{C}^*_{i,j+1/2} - v_{i,j-1/2}\widehat{C}^*_{i,j-1/2}}{\Delta y} = f_{i,j}\frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta y} \tag{2.30}$$

where $f_{i,j}$ is the characteristic function evaluated at the cell center, given by:

$$f_{i,j} = \begin{cases} 1 & \text{if} \quad C^n_{i,j} > 1/2 \\ 0 & \text{Otherwise.} \end{cases}$$

which is treated fully explicitly. Note that if there is no cell compression/expansion, Eqs. (2.29-2.30) and Eqs. (2.27-2.28) are identical. The update of the volume fractions is now performed as follows:

$$C^*_{i,j} = C^n_{i,j} - F_{i+1/2,j} + F_{i-1/2,j} + f_{i,j}(u_{i+1/2,j} - u_{i-1/2,j})\Delta t/\Delta x \tag{2.31}$$

$$C^{n+1}_{i,j} = C^*_{i,j} - G_{i,j+1/2} + G_{i,j-1/2} + f_{i,j}(v_{i,j+1/2} - v_{i,j-1/2})\Delta t/\Delta y \tag{2.32}$$

where the fluxes F and G on the cell faces are computed using the implicit Lagrangian mapping in Eq. (2.22). It is easily seen that this scheme is conservative. Weymouth and Yue (2010) presented a proof of the consistency, which for brevity sake is omitted here. This alternative conservative and consistent flux splitting is also implemented in our VOF solver, combined with the ELVIRA reconstruction of the interface, and will be referred to as VOF-WY.

2.4 Moment-Of-Fluid (MOF) method

The Moment-of-Fluid method is an improvement of the VOF method developed initially by Dyadechko and Shashkov (2005, 2006, 2008). The general idea is to store not only the volume fraction, but also the mass centroid of the reference fluid in each grid cell. A more accurate estimation of the interface normal can be obtained in this manner. An additional advantage is that information of the neighboring cells is not necessary, making it possible to capture small scale interface structures.

The method is based on moments from which we can deduce the volume fraction and centroid of the reference phase. Given a grid cell Ω and the phase indicator function f, the first two moments of the volume occupied by the reference fluid are defined as:

$$M_0 \equiv \int_\Omega f dv \tag{2.33}$$

$$\vec{M}_1 \equiv \int_\Omega \vec{x} f dv \tag{2.34}$$

If $V \equiv \int_\Omega dv$ is the volume of the grid cell, the volume fraction and centroid are subsequently defined as:

$$C \equiv \frac{M_0}{V} \tag{2.35}$$

$$\vec{x}_c \equiv \frac{\vec{M}_1}{M_0} \tag{2.36}$$

In the MOF method, a piecewise linear interface (PLIC) reconstruction is used to describe the interface as in most second-order VOF methods. Each interface is therefore defined by its normal $\vec{n}$ and distance from the cell center d. It is not possible in the general case to find a linear interface which exactly matches both the volume fraction and centroid. The priority is given to the volume fraction, which is matched exactly, whereas a best approximation is sought for the centroid. For a given normal vector $\vec{n}$, the distance d is therefore defined by the condition that the volume occupied by the reference fluid is exactly equal to $M_0 = C \cdot V$, which in turn also defines the centroid $\vec{x}_c^*$ of the reconstructed cell. The interface reconstruction procedure therefore consists in finding the normal vector $\vec{n}$ that gives the best match for $\vec{x}_c^*$ by minimizing the function:

$$E_{MOF} = ||\vec{x}_c^*(\vec{n}) - \vec{x}_c||_2 \tag{2.37}$$

The original MOF scheme is implemented using a multi-dimensional Lagrangian remapping advection scheme. However, this scheme does not exactly conserve the cell volume and a local correction is added to avoid inconsistency. Since the idea of the MOF method can be applied to any VOF advection scheme, we will use here the conservative and consistent directional-split advection scheme described in §2.3.4.

2.4.1 Interface reconstruction

The procedure for the interface reconstruction in 2D is explained in detail by Dyadechko and Shashkov (2005). We write the linear interface representation in the Hessian normal form $\vec{x} \cdot \vec{n} + d = 0$ where the origin for $\vec{x}$ is the center of the grid cell under consideration and $\vec{n}$ is the unit normal vector. In a two-dimensional setting, the interface is entirely defined by the polar angle ϕ, since the normal vector can be expressed as $\vec{n}(\phi) = (\cos(\phi), \sin(\phi))$ and the distance d is

determined from the target volume fraction C. This also defines the centroid of the reconstructed reference phase as a function of ϕ, i.e. $\vec{x}_c^*(\phi)$. A first estimation of the normal vector is given by $\vec{n}^0 = -\vec{x}_c^{ref}$, from which we can deduce the initial estimate of the angle ϕ^0. We will use the Gauss-Newton algorithm to minimize the centroid error E_{MOF}. A different method is used by Dyadechko and Shashkov (2005) for general polyhedral grid cells.

The derivative of Eq. (2.37) with respect to ϕ gives

$$E'_{MOF} = 2\left(\vec{x}_c(\phi) - \vec{x}_c^{ref}\right) \cdot \vec{x}'_c \tag{2.38}$$

where $\vec{x}'_c$ is given by

$$\vec{x}'_c = -\frac{|\Gamma(\phi)|^3}{12C|\Omega|}\vec{t} \tag{2.39}$$

where $|\Gamma(\phi)|$ is the length of the interface, $|\Omega|$ is the cell volume and $\vec{t} = (-\sin(\phi), \cos(\phi))$ is the tangent unit vector. A minimum error E_{MOF} is obtained when $(\vec{x}_c(\phi) - \vec{x}_c^{ref}) \cdot \vec{t} = 0$, which implies that $\Delta\vec{x}_c = \vec{x}_c(\phi) - \vec{x}_c^{ref}$ should be aligned with $\vec{n}$. Eq. (2.39) suggest that a good new estimate for ϕ is then

$$\phi^{k+1} = \phi^k + \frac{\Delta\vec{x}_c \cdot \vec{t}}{\frac{|\Gamma(\phi)|^3}{12C|\Omega|}}$$

which corresponds to the Gauss-Newton method. Iterations are repeated until $\Delta\phi$ falls below a small value, say 10^{-6}. A slightly more robust estimate of ϕ^{k+1} is obtained if we consider the function $g(\phi) = (\vec{x}_c(\phi) - \vec{x}_c^{ref}) \cdot \vec{t}$. The derivative of g is then

$$g'(\phi) = \vec{x}'_c \cdot \vec{t} + \Delta\vec{x}_c \cdot \frac{\partial\vec{t}}{\partial\phi} = -\frac{12|\Gamma(\phi)|^3}{C|\Omega|} - \Delta\vec{x}_c \cdot \vec{n}$$

The Newton method to obtain $g(\phi) = 0$ (condition for a minimum of E_{MOF}) gives

$$\phi^{k+1} = \phi^k - \frac{g(\phi^k)}{g'(\phi^k)} = \frac{\Delta\vec{x}'_c \cdot \vec{t}}{\frac{|\Gamma(\phi)|^3}{12C|\Omega|} + \Delta\vec{x}_c \cdot \vec{n}}$$

This method converges in most cases except when the grid cell is almost empty $C < 0.01$ or entirely filled $C > 0.99$. For these cases, we use the ELVIRA method to estimate the interface normal. After having determined the normal to the interface, the distance d follows from enforcing the volume fraction.

2.4.2 The one-dimensional explicit Lagrangian scheme revisited

First-order moments of the reference phase fluxed through the left and right cell boundaries are also computed:

$$\vec{M}_{1,i+1/2,j} = \frac{S(u_{i+1/2,j})}{\Delta x \Delta y}\int_{y_{j-1/2}}^{y_{j+1/2}}\int_{x_{i+1/2}-\max(0,u_{i+1/2})\Delta t/a_i}^{x_{i+1/2}-\min(0,u_{i+1/2})\Delta t/a_{i+1}} \vec{x} f(x,y)dxdy \tag{2.40}$$

from which we can easily deduce the centroid of the fluxed fluid $\vec{x}_{c,i+1/2,j} = \vec{M}_{1,i+1/2,j}/F_{i+1/2,j}$.

From the definitions in Eqs. (2.33) and (2.34), it is clear that when the cell is divided in parts, the moments can be obtained by addition of the partial moments, which is exactly what

is done for the volume fraction update:

$$C_i^{n+1} = a_i C_i^n - a_{i+1/2} F_{i+1/2} + a_{i-1/2} F_{i-1/2}$$

The update of the centroid can also be written as the sum of partial first-order moments:

$$\vec{x}_{c,i}^{n+1} C_i^{n+1} = \vec{x}_{c,II} a_i C_i^n - \vec{x}_{c,III} a_{i+1/2} F_{i+1/2} + \vec{x}_{c,I} a_{i-1/2} F_{i-1/2} \tag{2.41}$$

where $\vec{x}_{c,I}$, $\vec{x}_{c,II}$ and $\vec{x}_{c,III}$ denote the centroids of the cell parts in Fig. 2.2.

Since the velocity is described by linear interpolation Eq. (2.14), for a given volume-of-fluid, the location of the centroid does not change relative to the (moving) boundaries. It is therefore convenient to compute first the non-dimensional centroid, $0 < \widehat{x_c} < 1$, which remains constant when moving from t^n and t^{n+1}. The centroids of the fluxed fluid (explicit or implicit scheme) can then be computed easily from the new locations of boundaries.

2.4.3 The one-dimensional implicit Lagrangian scheme revisited

In addition to the fluxed fluid fraction $F_{i+1/2,j}$ defined by Eq. (2.22), the first moments of the reference phase fluxed through the left and right cell boundaries are also computed:

$$\vec{M}_{1,i+1/2,j} = \frac{S(u_{i+1/2,j})}{\Delta x \Delta y} \int_{x_{i+1/2}-\max(0,u_{i+1/2})\Delta t}^{x_{i+1/2}-\min(0,u_{i+1/2})\Delta t} \vec{x} f(x,y) dx dy \tag{2.42}$$

from which we can deduce the centroids $\vec{x}_{c,i+1/2} = \vec{M}_{1,i+1/2,j}/F_{i+1/2,j}$. The update of the volume fraction and the centroid is therefore according to

$$\begin{aligned} C_i^{n+1} &= b_i \left(C_i^n - F_{i+1/2} + F_{i-1/2} \right) \\ \vec{x}_{c,i}^{n+1} C_i^{n+1} &= b_i \left(\vec{x}_{c,II} C_i^n - \vec{x}_{c,III} F_{i+1/2} + \vec{x}_{c,I} F_{i-1/2} \right) \end{aligned} \tag{2.43}$$

2.5 Overview of tested methods

The volume-of-fluid methods conserve the total mass. The relative mass error, E_M, given by Eq. (1.59) is therefore extremely small and mainly due to the initialization of the fractions. In the absence of initialization errors, E_M is of the order of 10^{-14}.

The positioning error, E_1, of the interface for the volume-of-fluid methods, however, is not negligible. It is defined in the same sense as for the level-set method, as:

$$E_1 = \frac{\int_\Omega |f^R - f^C| dv}{\text{Exact volume}} \tag{2.44}$$

where the superscripts R and C refer to the reference and computed solutions and f is the phase indicator function.

For the Zalesak disk (§1.5.2), the exact signed-distance to the interface is known. We note this signed-distance ϕ^E, which is defined on the cell vertices. The positioning error, E_1, in Eq. (2.44) is computed by:

$$E_1 = \frac{\sum_{(i,j)\in\Omega} \int_{\Omega_{i,j}} |H(\phi_{i,j}^E) - f_{i,j}^C| d\Omega}{\text{Exact volume}} \tag{2.45}$$

Using the same procedure as for the level-set on §A.1, we compute the positioning error, E_1, by subdividing the cell in 1000^2 sub-cells, and we determine the fraction provided by the reference solution inside each sub-cell using bicubic interpolation if the cell is mixed. For the computed

solution, we use the linear interpolation since the reconstruction algorithm provides a linear interface inside mixed cells, or the fraction for one fluid cells.

In the case of the shearing vortex (§1.5.3) and the time-reversed single vortex (§1.5.4), we use the Lagrangian markers as a reference solution as in §1.5.1. The same treatment as described for the Zalesak disk is adopted for the error computation at the end of the simulation. On a given (coarse) grid at an intermediate time, we use a slightly different formulation than that in Eq (2.45). The procedure is:

1. Compute the level-set on a 2000×2000 grid for the shearing vortex (or 1024×1024 grid for the time-reversed single vortex) (see §A.2)

2. Determine the volume fraction using the Heaviside integral on each of the 2000^2 (1024^2, respectively) cells.

3. Determine $C^R_{i,j}$ on each cell of the actual coarse grid by summing up the fractions on $(2000/N)^2$ corresponding sub-cells ($(1024/N)^2$ sub-cells for the time-reversed single vortex).

Then, the positioning error, E_1, is computed from:

$$E_1 = \frac{\sum_{(i,j)\in\Omega} |C^R_{i,j} - C^C_{i,j}| \Delta x \Delta y}{\text{Exact volume}} \tag{2.46}$$

The volume-of-fluid methods we have implemented are purely geometric as described before. We have tested the three methods :

VOF

The VOF method that we implemented in this work is given by Algorithm 2.1. Note that the direction x and y are alternated after each time step.

Algorithm 2.1 VOF method

Input: C^n, $\vec{u}^{n+1/2}$, Δt
  for cells with $0 < C^n < 1$ **do**
    n^n, $d^n \leftarrow$ Compute interface normal by minimizing Eq. (2.5) (C^n)
  end for
  $C^* \leftarrow$ Advance and remap in x- (or y-) direction using Eq. (2.25) (C^n, $\vec{n}^n$, d^n, $u^{n+1/2}$)
  for cells with $0 < C^* < 1$ **do**
    n^*, $d^* \leftarrow$ Compute interface normal by minimizing Eq. (2.5) (C^*)
  end for
  $C^{n+1} \leftarrow$ Advance and remap in y- (or x-) direction using Eq. (2.26) (C^*, $\vec{n}^*$, d^*, $v^{n+1/2}$)
  $\vec{u}^{n+3/2} \leftarrow$ Advance flow equations (C^{n+1}, $\vec{u}^{n+1/2}$, Δt)
  return C^{n+1}, $\vec{u}^{n+3/2}$

VOF-WY

The same algorithm as VOF (given on Algorithm 2.1), except that the volume fraction is advected as recommended by Weymouth and Yue (2010). We use Eqs. (2.31) and (2.32) instead of Eqs. (2.25) and (2.26).

MOF

It is provided by the algorithm below

Algorithm 2.2 MOF method

Input: C^n, $\vec{x}_c^n$, $\vec{u}^{n+1/2}$, Δt
  for cells with $0 < C^n < 1$ **do**
    n^n, $d^n \leftarrow$ Compute interface normal by minimizing Eq. (2.37) (C^n, $\vec{x}_c^n$)
  end for
  C^*, $\vec{x}_c^* \leftarrow$ Advance and remap in x- (or y-) direction using using Eqs. (2.25) and (2.43) (C^n, $\vec{n}^n$, d^n, $u^{n+1/2}$)
  for cells with $0 < C^* < 1$ **do**
    n^*, $d^* \leftarrow$ Compute interface normal by minimizing Eq. (2.37) (C^*, $\vec{x}_c^*$)
  end for
  C^{n+1}, $\vec{x}_c^{n+1} \leftarrow$ Advance and remap in y- (or x-) direction using Eqs. (2.26) and (2.41) (C^*, $\vec{n}^*$, d^*, $v^{n+1/2}$)
  $\vec{u}^{n+3/2} \leftarrow$ Advance flow equations (C^{n+1}, $\vec{u}^{n+1/2}$, Δt)
  return C^{n+1}, $\vec{x}_c^{n+1}$, $\vec{u}^{n+3/2}$

2.6 Results and discussion

We start first by validating our implementation of the VOF method by comparing it with the results provided by Scardovelli and Zaleski (2003) for the time-reversed single vortex of §1.5.4 at the final time $t = T = 2$. The CFL number based on the maximum velocity is kept constant for all the simulations and equal to unity. The only difference between our simulations and that of Scardovelli and Zaleski (2003) is the manner in which the positioning errors are computed. In our computation, the error E_1 is computed using the interpolation inside mixed cells, while the average fraction over the cell is considered in Scardovelli and Zaleski (2003).

Table 2.1 below shows the interface positioning errors, E_1. The method named "ELVIRA/EI-LE" is identical to our VOF implementation and the error levels compare well. For the VOF-WY, the positioning error is slightly higher compared to the VOF method, which is perhaps due to the large CFL number. The MOF method is more accurate than the volume-of-fluid methods.

Table 2.1: Time-reversed single vortex: comparison of our VOF implementation with the results of Scardovelli and Zaleski (2003).

	Scardovelli and Zaleski (2003)		Our implementation					
	ELVIRA/EI–LE		VOF		VOF-WY		MOF	
Grid	$\boldsymbol{E_1}(\%)$	$\mathcal{O}$	$\boldsymbol{E_1}(\%)$	$\mathcal{O}$	$\boldsymbol{E_1}(\%)$	$\mathcal{O}$	$\boldsymbol{E_1}(\%)$	$\mathcal{O}$
32^2	3.565	–	3.573	–	4.42	–	1.42	–
64^2	0.914	1.95	0.921	1.95	1.13	1.97	0.35	2.02
128^2	0.205	2.15	0.207	2.15	0.29	1.96	0.08	2.13

In the next section, the influence of the CFL number on the accuracy of the interface transport will be investigated. The spatial convergence rate is presented on §3.6.3.

2.6.1 Zalesak disk

The Zalesak disk interface is plotted after a full revolution when its interface is advected using the VOF and MOF method on a 100^2 grid on Fig. 2.4 for three different CFL numbers. The VOF-WY gives exactly the same interface as the VOF method since the term $(u_{i+1/2,j} - u_{i-1/2,j})/\Delta x = (v_{i,j+1/2} - v_{i,j-1/2})/\Delta y = 0$ which means that there is no compression/expansion of the cells (see §2.3.5). The figure shows that the MOF method is less sensitive to the CFL number than the VOF method.

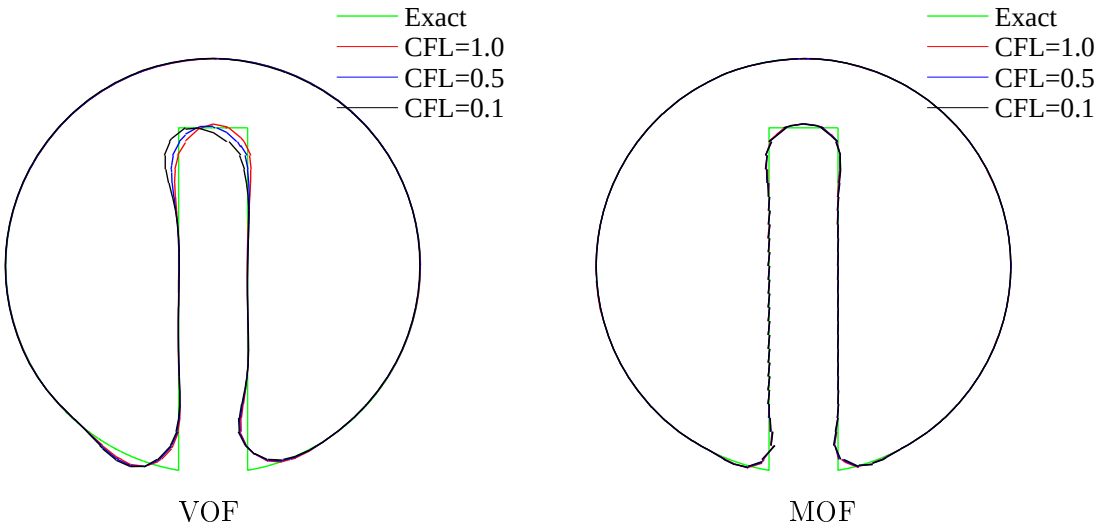

Figure 2.4: Zalesak disk interface after a full revolution for different CFL numbers using the VOF (left) and MOF (right) methods. The grid size is 100^2.

The positioning error, E_1, as a function of the CFL number is plotted for both the VOF and MOF methods on Fig. 2.5. We observe that by decreasing the CFL value, the positioning error E_1 of the VOF method increases until it reaches a maximum value for which this error does not vary when reducing the CFL number. For the MOF method, only a small increase of the error

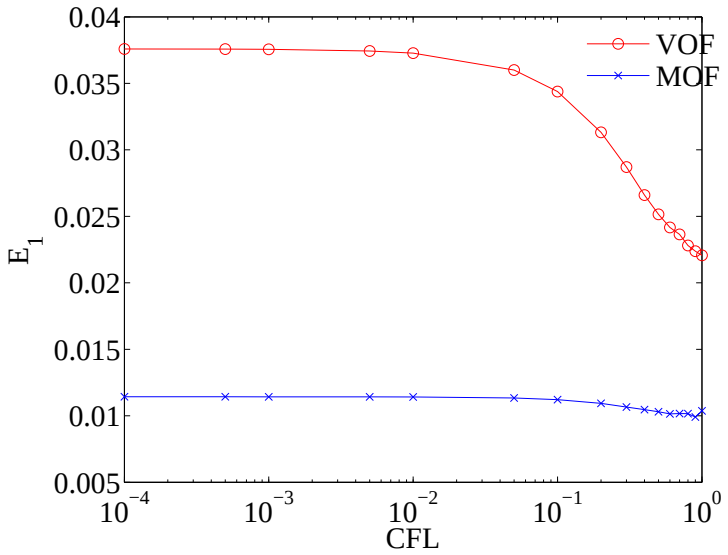

Figure 2.5: Positioning error of the Zalesak disk interface after a full rotation as a function of the CFL number. The grid size is 100^2.

is observed when the CFL number is decreased. This is due to the use of the information of the centroids that makes the interface reconstruction and advection more accurate than in the standard VOF methods.

In comparison with the level-set method with volume constraint (Figs. 1.21 and 1.26 in §1.6.3), it seems that the level-set method advected with WENO5 scheme is better (if we do not take into consideration the mass loss/gain) than the VOF method, and is comparable to the MOF method.

2.6.2 Time-reversed single vortex

In the time-reversed single vortex, the interface is only moderately stretched at t=T/2 since the velocity magnitude is decaying in time and the period is relatively small (T=2). The obtained interface at maximum stretching when advected with the three tested methods is represented on Fig. 2.6 for three different CFL numbers. The figure shows that the CFL number has almost no influence on the quality of the obtained interface.

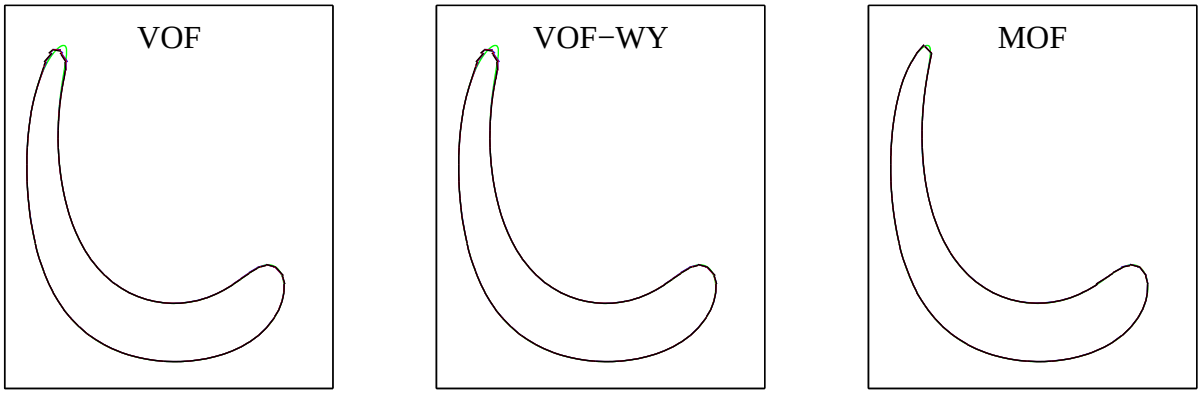

Figure 2.6: Positioning error of the vortex interface at t=T/2=1 as a function of the CFL number. The grid size is 64^2. Green: reference solution, Black: CFL=0.1; Red: CFL=0.5 and Blue: CFL=1.0.

Fig. 2.7 plots the positioning error of the interface at t=T/2 and t=T for the three methods. Again, we see that the MOF method is more precise than the VOF and VOF-WY methods. The positioning error at $T/2$ is less than that at T, which is due to the accumulation of errors. This error does not vary for the CFL number in the range $[10^{-4}, 10^{-1}]$. In this range, the VOF and the VOF-WY methods provide exactly the same interface error. However, for CFL in $[0.1, 1]$, the VOF-WY method seems to be slightly more accurate than the VOF method. In this range, the positioning error decreases than increases. The MOF method remains more accurate than the VOF and VOF-WY methods. The volume-of-fluid and MOF methods predict better this test case than the level-set method with volume constraint (Figs. 1.24 and 1.26 on §1.6.3).

2.6.3 Shearing Vortex

In the shearing vortex simulation, the interface undergoes more stretching and deformation than for the previous test case. Fig. 2.8 plots the interface of the shearing vortex at time $t = T/2 = 3$ for the two VOF methods and MOF method. We choose again the three CFL values: 1.0, 0.5 and 0.1. We see that when refining the time step, the VOF and VOF-WY methods loose in precision, i.e. the vortex is more fragmented and the non-fragmented part of the filament becomes shorter. These two methods also have a larger error in the prediction of the starting point of the filament. The MOF method seems to predict it more accurately and generates a less fragmented filament.

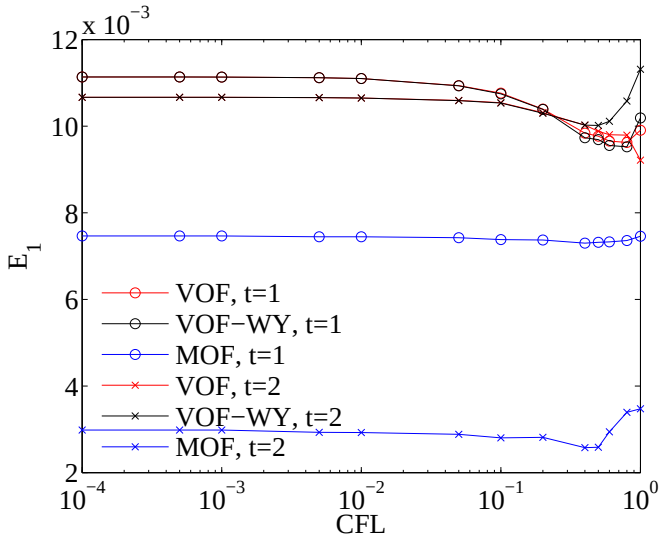

Figure 2.7: Positioning error of the time-reversed single vortex interface at t=T/2=1 and t=T=2 as a function of the CFL number. The grid size is 64^2.

Its particularity when refining the time step is to make the filament tail slightly displaced from the exact solution.

The fragmentation of thin filaments seems to be a result and counterpart of the mass conservation property of VOF methods. As shown in the previous chapter, the level-set methods tend to either thicken or shorten the filament, gaining or losing mass in the process, the VOF methods break up the filaments in small parts of thickness roughly the size of several grid cells. The overall mass is conserved, but the fragmentation introduces non-physical droplets in the simulation.

Fig. 2.9 shows the results when the velocity is inverted at $t = T/2$ and the interface should return to the initial position (circle) at $t = T$. Again, we remark that the interface position prediction is better for high CFL numbers. When the CFL number is too small, a part of the filament is not advected back into the circle. The MOF method seems to be more affected by this phenomena as shown also on Fig. 2.10. This is perhaps due to the very thin filament obtained at $t = T/2$, which is not advected accurately backward until $t = T$.

In comparison to the level-set method with volume constraint (Figs. 1.22, 1.23 and 1.26 in §1.6.3), the volume-of-fluid methods tend to break the filament (at $t = T/2 = 3$) while the level-set method makes it thicker. The MOF method is clearly better than the level-set method.

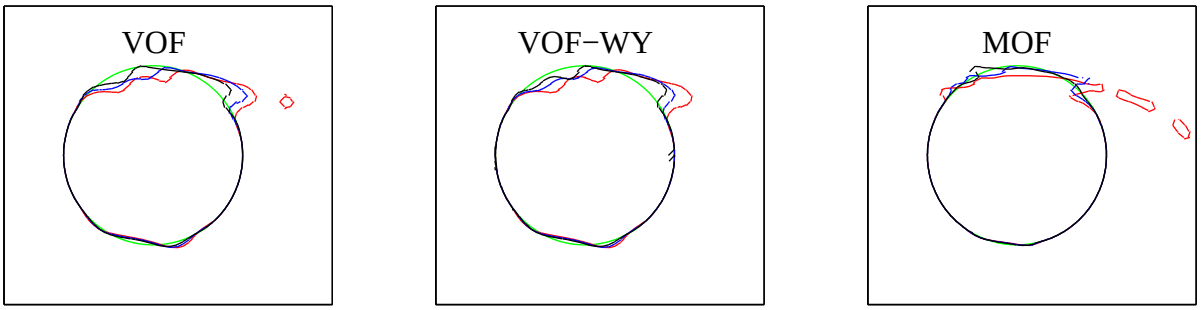

Figure 2.9: Positioning error of the shearing vortex interface at $t = T = 6$ as a function of the CFL number. The grid size is 100^2. Green: exact solution, Black: CFL= 1.0 ; Blue: CFL= 0.5 and Red: CFL= 0.1

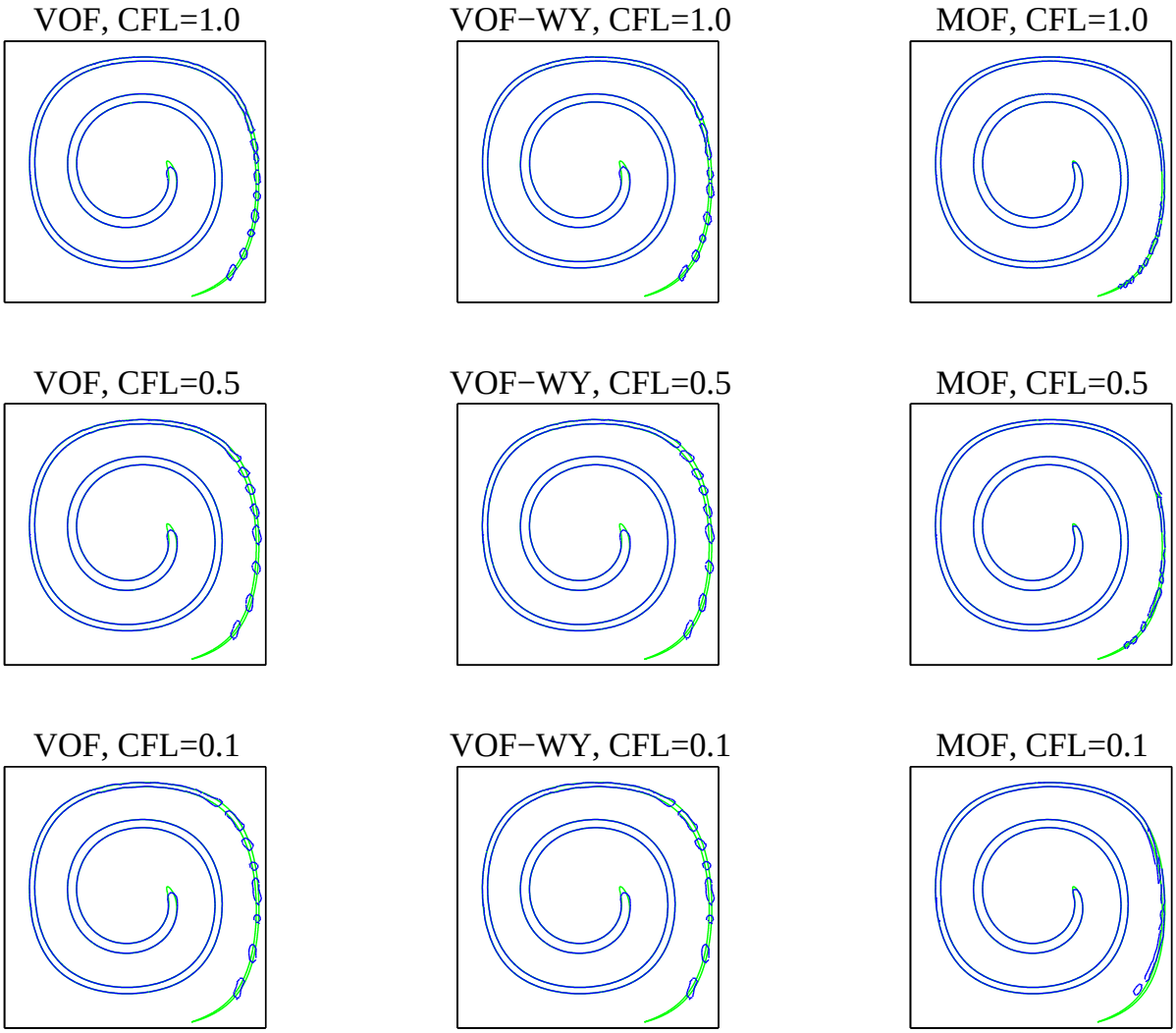

Figure 2.8: Positioning error of the shearing vortex interface at t=T/2=3 as a function of the CFL number. The grid size is 100^2. Green: reference solution, Blue: Computed solution.

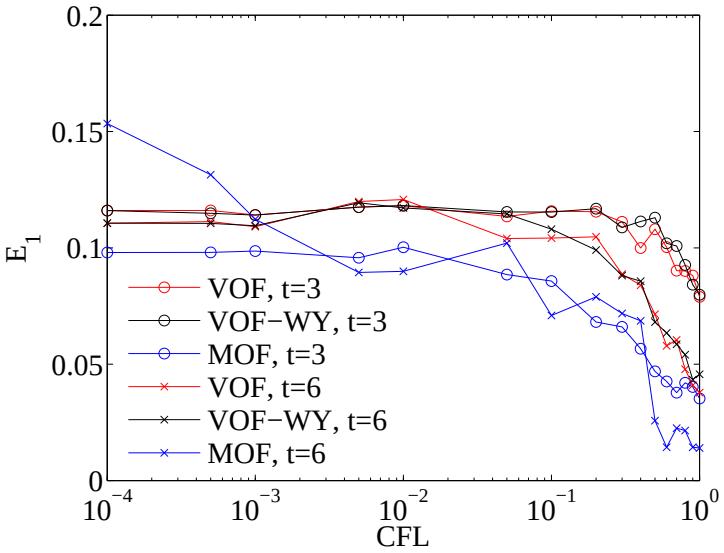

Figure 2.10: Positioning error of the shearing vortex interface at $t = T/2 = 3$ and $t = T = 6$ as a function of the CFL number. The grid size is 100^2.

2.7 Conclusions

In this chapter, we have investigated the geometrical resolution of the volume-of-fluid advection equation in two-dimensional uniform grids. The selected directional-splitting methods for temporal integration are consistent (keeps C within the limits of 0 and 1) and conservative (the total mass of the reference fluid is conserved). We have implemented the ELVIRA and moment-of-fluid method to reconstruct the piecewise linear interface. We investigated the accuracy of these methods for solving two-dimensional interface transport problems using the well know test cases of the Zalesak disk, the shearing vortex and the time-reversed single vortex.

In general, both the volume-of-fluid and moment-of-fluid methods are shown to be more accurate and robust than the level-set methods. In particular, the moment-of-fluid method is more accurate and less sensitive to the time step variations than the ELVIRA method. On the contrary, the moment-of-fluid method is much more difficult to implement, in particular when cylindrical or unstructured grids need to be used. The ELVIRA volume-of-fluid methods are relatively simple to implement, in particular on Cartesian grids and using the basic operations defined in Appendix A. While not explicitly shown, the volume-of-fluid methods require less computational time than the level-set methods due the localized approach (the reconstruction algorithm is applied only on grid cells containing the interface). However, the volume-of-fluid and moment-of-fluid methods tend to break up thin filaments into artificial small "droplets" of the size of a few grid cells.

Chapter 3

Coupled Volume-Of-Fluid and Level-Set methods

3.1 Introduction

A coupled method combines two different interface methods, often Eulerian, although some techniques couple an Eulerian and a Lagrangian interface methods. The coupling between Eulerian methods was first conceived by Bourlioux (1995), who proposed to combine the level-set and the volume-of-fluid methods. The idea is to gather the advantages of each one in the same method: the well-defined interface description of the level-set method and the mass conservation of the volume-of-fluid methods. This first coupling technique became well known by the work of Sussman and Puckett (2000), who proposed a detailed description of the algorithm and improved the coupling technique by reinitializing the level-set method by means of the level-set and the volume-of-fluid method. They also described the implementation of the method for axisymmetric and three-dimensional coordinate systems and named it Coupled Level-Set and Volume-Of-Fluid (CLSVOF) method. Later, Son and Hur (2002) have presented a slightly different implementation of the CLSVOF method in two dimensions and in axisymmetric coordinate systems, then in three dimensions in a subsequent paper (Son, 2003). The Hybrid Particle-Level-Set (PLS) method which couples the level-set with a Marker method (Lagrangian) has been implemented by Enright et al. (2002). The markers in this case ensure the mass conservation of the level-set method and enhance the interface description in the under-resolved regions. However, these markers need to be reseeded when the interface undergoes topological variations.

Other coupled methods have been proposed in the literature, such as the Mass Conserving Level-Set (MCLS) method by van der Pijl et al. (2005), in which a slightly different coupling algorithm was proposed for two-dimensional configurations, and three dimensions in van der Pijl et al. (2008). The MCLS method is mainly based on the level-set method. It is not as widely used as the CLSVOF methods. A comparison between the volume-of-fluid and the MCLS methods, coupled with the Navier-Stokes solver, can be found in Denner et al. (2014b). Their results shows that the MCLS method is slightly less accurate than the VOF methods, which is most probably due to the implementation of the surface tension term. Sun and Tao (2010) have proposed another coupled method called the coupled Volume-Of-fluid and level-SET (VOSET) method in which only the volume-of-fluid is advanced in time. The level-set is determined as a signed-distance function through a geometric iterative method. Later, Albadawi et al. (2013) proposed a simplified coupling between the volume-of-fluid and the level-set methods named S-CLSVOF that the authors implemented into the free CFD software OpenFOAM. In reality,

this last one is similar to the VOSET technique since only the volume-of-fluid is advected in time and the level-set is first deduced from the volume fractions and then reinitialized using the standard reinitialization equation. The new volume-of-fluid method by Wang et al. (2012b) is somehow another VOSET alternative, but its authors have not categorized it among the coupled methods. Finally, Jemison et al. (2013) proposed to couple the Moment-Of-Fluid method, a few years after its apparition, with the level-set method. The method is called Coupled Level-Set and Moment-Of-Fluid (CLSMOF) method. A literature review on the interface capturing methods (level-set, VOF, CLSVOF, VOSET, phase field methods, surface tension models) may be found in Wörner (2012).

In the current work, we focus on the study of the coupling techniques issued from the Eulerian methods, mainly the level-set and the volume-of-fluids. We recall that the reason of coupling the level-set and the volume-of-fluid methods is to gather the advantages of both methods in the same coupled method. The volume-of-fluid method conserves mass but the interface representation depends on the reconstruction algorithm, and is often discontinuous on the junction between two neighboring cells. On the contrary, the level-set method gives a good interface representation despite of not conserving the mass. We can distinguish two families of the coupled methods:

1. The level-set and the volume-of-fluid are both advected in time. In this category, we have for example the CLSVOF method (Sussman et al., 1999), the CLSMOF method (Jemison et al., 2013) and MCLS method (van der Pijl et al., 2005). The coupled methods differ in how each method interacts with the other.

2. Only VOF is advected in time, and the level-set field is deduced from the volume fractions. In this category, the normal to the interface is either computed by means of the PLIC algorithm (VOSET method by Sun and Tao (2010), reconstructed distance function by Cummins et al. (2005)), or from the deduced level-set (new VOF method by Wang et al. (2012b)).

In order to verify the benefits of coupled methods with respect to the basic VOF and LS methods, we have selected, among the most popular coupled methods, the VOSET, MCLS, CLSVOF and CLSMOF approaches. In the next subsections, we will detail the main steps constituting each one of these coupled methods. Then, we test the ability of each method in transporting simple interfaces.

3.2 Coupled Volume-Of-fluid and level-SET method (VOSET)

The coupled volume-of-fluid and level-set (VOSET) method was first introduced by Sun and Tao (2010). It consists only of advecting the volume fractions in time, and then computing by geometric iterations the signed-distance function to calculate the surface tension term. It is relatively easy to implement in the sense that when the volume-of-fluid method is already used, it is easy to integrate the computation of the signed-distance function.

Few works are related to the VOSET method in the literature since it is a recent technique. We can for example mention the work of Guo et al. (2011) and Ling et al. (2014), where it is used to study phase change phenomena. Recently, it has been extended to adaptive grid systems in Wang et al. (2013, 2014), then applied to study the bubble dynamics in the presence of electric fields (Wang et al., 2015, 2016). In another study, Shi et al. (2014) has applied the VOSET method to simulate a falling ferrofluid droplet in a uniform magnetic field. The VOSET method is also presented for 3D cartesian grids in Ling et al. (2015).

The main steps constituting the VOSET method are summarized in 2D in Algorithm 3.1. Note that for a pure interface advection problem (without the Navier-Stokes equations), the

VOSET is identical to the VOF method.

Algorithm 3.1 VOSET method

Input: C^n, $\vec{u}^{n+1/2}$, Δt

$\vec{n}$, $d \leftarrow$ Reconstruct the interface using a PLIC Algorithm (C^n)

$C^* \leftarrow$ Advect the volume-of-fluid in the x-(or y-) direction (C^n, $u^{n+1/2}$, $\vec{n}$, d)

$\vec{n}$, $d \leftarrow$ Reconstruct the interface using a PLIC Algorithm (C^*)

$C^{n+1} \leftarrow$ Advect the volume-of-fluid in the y-(or x-) direction (C^*, $v^{n+1/2}$, $\vec{n}$, d)

$\vec{n}$, $d \leftarrow$ Reconstruct the interface using a PLIC Algorithm (C^{n+1})

ϕ^{n+1}, $\vec{n}$, $d \leftarrow$ Determine the signed-distance function iteratively (C^{n+1}, $\vec{n}$, d)

$\vec{u}^{n+3/2} \leftarrow$ Advance flow equations (ϕ^{n+1}, $\vec{u}^{n+1/2}$)

return ϕ^{n+1}, $\vec{u}^{n+3/2}$, C^{n+1}

In this work, the VOSET is implemented as follows. The volume-of-fluid is first advected using the conservative-consistent EI/LE operator splitting of §2.3.4. In the advection step, the interface is reconstructed using the ELVIRA algorithm, whereas Sun and Tao (2010) used the Youngs-PLIC method for this purpose. The reconstruction algorithm gives in the case of the VOSET method also a first estimation of the interface for the signed-distance reconstruction. Once the VOF function is advected and the interface is reconstructed, an iterative procedure is used to determine the signed-distance function, with the following steps:

Step 1. Set the initial field for the signed-distance function ϕ in the whole computational domain such as:

$$\phi^0_{i,j} = \begin{cases} -\beta & \text{if } C_{i,j} \geq 0.5 \\ \beta & \text{if } C_{i,j} < 0.5 \end{cases} \tag{3.1}$$

where β is a distance of few grid sizes. It's value will be given in the next step.

Step 2. Flag cells near the interface. The region of flagging is of width of $3\Delta x$ on each side of the interface. This region is large enough to smooth the discontinuities and to determine accurately the curvature. With the flagging region equal to $3\Delta x$, β should be taken as $4\Delta x$.

Step 3. Compute the signed-distance function ϕ in the flagged region around the interface. This is done by computing the minimal distance d separating the cell center (i, j) from the interface on the region of $3\Delta x$ around the interface. Then, set the sign of the distance according to the volume fraction inside the cell (i, j) as follows:

$$\phi_{i,j} = \begin{cases} -d & \text{if } C_{i,j} \geq 0.5 \\ 0 & \text{if } C_{i,j} = 0.5 \\ d & \text{if } C_{i,j} < 0.5 \end{cases} \tag{3.2}$$

Step 4. In each cell containing the interface ($0 < C < 1$), reconstruct the piecewise-linear interface by first determining the normal from the level-set function as $\vec{n} = \nabla\phi/|\nabla\phi|$ using second-order central differencing. The distance of this interface to the cell center is then the only free parameter that we should enforce to satisfy the volume fraction.

Step 5. Once the reconstruction in each interface cell is completed, compute a new signed-distance function by repeating steps 1 to 4 if the maximum number of iterations is not reached.

Sun and Tao (2010) recommended to repeat steps 1 to 4 one time if the grid is coarse, and three times for fine grids. In our computations, we have used 1, 3, 5 and 10 iterations with a fixed grid size.

3.3 Mass Conserving Level-Set method (MCLS)

The mass conserving level-set (MCLS) was first introduced in two-dimensional cartesian grids by van der Pijl et al. (2005). It has been extended to three-dimensional configurations in a subsequent paper (van der Pijl et al., 2008), then to non-uniform grids in Raees (2016). Cubos-Ramírez et al. (2016) has used the MCLS to compute the breaking dam problem.

This coupling method consists of advecting and reinitializing the level-set with at best second-order spatial and temporal discretization schemes in order to guarantee the smoothness of the solution. Mass conservation is then ensured by correcting the level-set by means of the volume fractions.

The algorithm of this method is summarized (in 2D) in Algorithm 3.2 below (van der Pijl et al., 2005). Two mathematical tools are needed: a tool to determine the volume fractions from the level-set, and a tool to compute the level-set knowing the volume fractions. The advantage of this coupling technique is the easy extension from 2D to 3D configurations. In the MCLS method, the level-set advection and reinitialization does not require high-order space and time discretization. First-order schemes with some numerical dissipation are preferred: the goal is to get a sufficiently smooth level-set.

Algorithm 3.2 MCLS method

Input: ϕ^n, $\vec{u}^{n+1/2}$, Δt

$C^n \leftarrow$ Determine the volume fractions using Eq. (3.4) (ϕ^n)

$C^* \leftarrow$ Advect the volume-of-fluid in the x-(or y-) direction using Eq. (2.25) with the flux definition in Eq. (3.6) (C^n, $u^{n+1/2}$, ϕ^n)

$\phi \leftarrow$ Determine the distance function from the volume fractions Eq. (3.5) (C^*)

$C^{n+1} \leftarrow$ Advect the volume-of-fluid in the y-(or x-) direction using Eq. (2.26) with the flux definition in Eq. (3.7) (C^*, $u^{n+1/2}$, ϕ)

$\phi^{n+1,*} \leftarrow$ Advect and reinitialize the level-set (ϕ^n)

$\phi^{n+1} \leftarrow$ Correct the level-set by enforcing total mass conservation using Algorithm 3.3 (C^{n+1}, $\phi^{n+1,*}$)

$\vec{u}^{n+3/2} \leftarrow$ Advance flow equations (ϕ^{n+1}, $\vec{u}^{n+1/2}$)

return ϕ^{n+1}, $\vec{u}^{n+3/2}$

3.3.1 Computing volume fractions from the level-set function

If we approximate the interface by a line segment in each mixed cell, the volume fraction inside the cell will be the area under the line segment as in the PLIC methods. Since the level-set field is known, we can determine the normal using the central-difference scheme:

$$n_x = \frac{\partial \phi}{\partial x} \quad \text{and} \quad n_y = \frac{\partial \phi}{\partial y}$$

The distance of the interface to the cell center is the level-set value itself. The area of the fluid of interest is the integral of the line equation in the cell. As in the PLIC algorithms, the number

of possible cases is significantly reduced by rotating the cell in such a way that the normal to the interface points toward the first quadrant. Four critical distances can be determined, which correspond to the interface passing through the cell vertices. The number of these critical values is reduced to two when considering the absolute value of the level-set. They are:

$$\phi_{max} = \frac{1}{2}\left(\Delta x|n_x| + \Delta y|n_y|\right) \quad \text{and} \quad \phi_{mid} = \frac{1}{2}|\Delta x|n_x| - \Delta y|n_y|| \tag{3.3}$$

The values $\phi = \pm\phi_{max}$ denote the extreme cases where the interface cuts through the corner of the cell, which is either completely full or empty. When $\phi \in [-\phi_{mid}, \phi_{mid}]$ the interface cuts through two opposite edges of the cell, and the fraction varies linearly with the distance. Outside this interval, the interface cuts through two adjacent edges and the fraction is quadratic function of the distance. See Fig. 3.1 below.

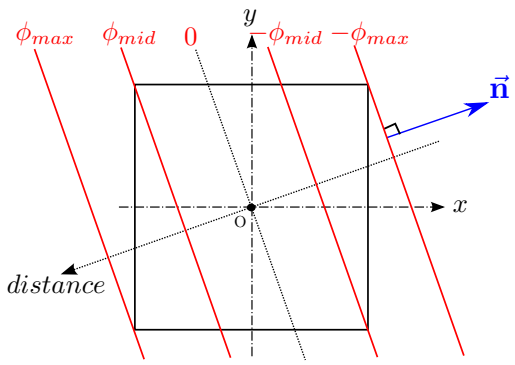

Figure 3.1: Characteristic distances in 2D cartesian coordinates.

In this work, inside the reference fluid the level-set is (arbitrary) defined negative and the volume fraction is equal to unity. In the following, we drop the index (i, j). In 2D, the volume fraction is given by:

$$C = f(\phi, \nabla\phi) = \begin{cases} 1 & \text{if} \quad \phi \leq -\phi_{max} \\ 1 - \dfrac{1}{2}\dfrac{(\phi_{max} + \phi)^2}{\phi_{max}^2 - \phi_{mid}^2} & \text{if} \quad -\phi_{max} < \phi < -\phi_{mid} \\ \dfrac{1}{2} - \dfrac{\phi}{\phi_{max} + \phi_{mid}} & \text{if} \quad -\phi_{mid} \leq \phi \leq \phi_{mid} \\ \dfrac{1}{2}\dfrac{(\phi_{max} - \phi)^2}{\phi_{max}^2 - \phi_{mid}^2} & \text{if} \quad \phi_{mid} < \phi < \phi_{max} \\ 0 & \text{if} \quad \phi \geq \phi_{max} \end{cases} \tag{3.4}$$

Note that the gradient $\nabla\phi$ is included in the function arguments in order to emphasize that the interface normal is computed from the derivatives of ϕ prior to evoking Eq. (3.4). This function computes the exact volume fraction, considering a piecewise-linear interface with normal computed from the gradient and distance from the level-set. It is a different manner of writing Algorithm A.1 on §A.3.1. It is much faster to execute Eq. (3.4) than the Algorithm A.1.

The total mass of the reference fluid defined by the level-set field can now be determined by first using Eq. (3.4) to compute the volume fraction inside each computational cell, then sum over all cells. The mass error of the initial interface of the Zalesak disk and a circle are plotted on Fig. 3.2. This method has a second-order accuracy behaviour in the determination of the total mass of the Zalesak disk and a circle and it is slightly more accurate than the bilinear interpolation (Appendix A).

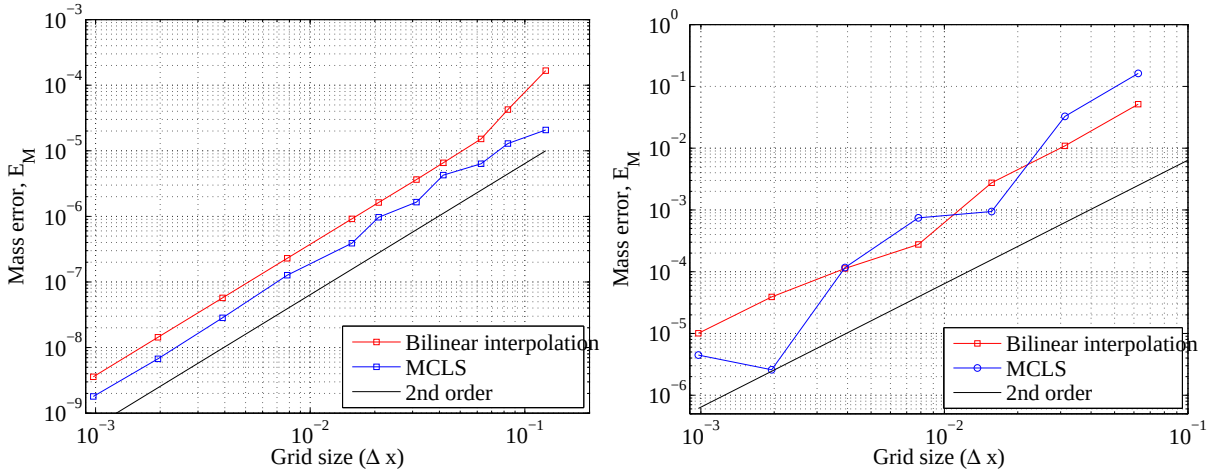

Figure 3.2: Mass error when the mass of a circle (left) or a Zalesak disk (right) is estimated using Eq. (3.4) vs bilinear interpolation with 1000^2 sub-cells.

3.3.2 Level-set correction

Now, in order to make the level-set mass conserving, we should correct it in the region where the volume fraction determined from Eq. (3.4) is not equal to the volume fraction resulting from the advection of the volume fractions. Algorithm 3.3 summarizes the steps of correcting the level-set according to van der Pijl et al. (2005). In our implementation, we set $\epsilon = 10^{-9}$.

Algorithm 3.3 Level-set correction in the MCLS method.

Input: ϕ^{n+1}, C^{n+1}, ϵ

$C \leftarrow$ Determine the volume fractions from the level-set using Eq. (3.4) (ϕ^{n+1})

while $\max\left(\left|C - C^{n+1}\right|\right) > \epsilon$ **do**

$\phi^{n+1} \leftarrow$ Correct the level-set using Eq. (3.5)

$C \leftarrow$ Determine the volume fractions from the level-set using Eq. (3.4) (ϕ^{n+1})

end while

return ϕ^{n+1}

The level-set represents the signed distance to the interface. Correcting this distance consists in moving the reconstructed interface in the normal direction to this interface. So, we need first to compute ϕ_{max} and ϕ_{mid} for each mixed cell, which is given by Eq. (3.3). We define an intermediate fraction C_{mid}:

$$C_{mid} = \frac{\phi_{max} + 3\phi_{mid}}{2(\phi_{max} + \phi_{mid})}$$

The fractions C_{mid} and $1 - C_{mid}$ correspond to the cases where the interface cuts through corners of the cell (the level-set equals ϕ_{mid} and $-\phi_{mid}$ respectively, see Fig. 3.1. When the estimated volume fraction inside the cell (Eq. (3.4)) is not equal to the volume fraction C^{n+1}, we correct the level-set. The new signed-distance function in this cell is then given by:

$$\phi = g(C, \nabla\phi) = \begin{cases} \phi_{max} & \text{if} \quad C \leq 0 \\ \zeta & \text{if} \quad 0 < C < 1 - C_{mid} \\ \gamma & \text{if} \quad 1 - C_{mid} \leq C \leq C_{mid} \\ \beta & \text{if} \quad C_{mid} < C < 1 \\ -\phi_{max} & \text{if} \quad C \geq 1 \end{cases} \tag{3.5}$$

where

$$\beta = \sqrt{2(1-C)\left(\phi_{max}^2 - \phi_{mid}^2\right)} - \phi_{max}$$
$$\gamma = -\left(C - \frac{1}{2}\right)(\phi_{max} + \phi_{mid})$$
$$\zeta = -\sqrt{2C\left(\phi_{max}^2 - \phi_{mid}^2\right)} + \phi_{max}$$

This function computes the distance from a given volume fraction and normal $\vec{n}$. Note that a similar function (Algorithm A.2) with a slightly different implementation is also used in the VOF method to compute the distance d.

3.3.3 Volume-of-fluid advection and flux computation

The advection of the volume fractions is realized through the conservative and consistent directional splitting as in §2.3.4, with some modifications in the manner of determining the fluxes through the cell faces. The flux at the cell-face $(i+1/2, j)$, for example, is given by

$$F_{i+1/2,j} = \begin{cases} u_{i+1/2,j}\Delta t \; f\left(\phi_{i+1/2,j}, \left(\dfrac{-u_{i+1/2,j}\Delta t}{\Delta x} n_{x,i+1,j}, n_{y,i+1,j}\right)^t\right) & \text{if} \quad u_{i+1/2,j} < 0 \\ u_{i+1/2,j}\Delta t \; f\left(\phi_{i+1/2,j}, \left(\dfrac{u_{i+1/2,j}\Delta t}{\Delta x} n_{x,i,j}, n_{y,i,j}\right)^t\right) & \text{elsewhere.} \end{cases} \tag{3.6}$$

in the implicit step Eq. (2.27), and

$$F_{i+1/2,j} = \begin{cases} u_{i+1/2,j}\Delta t \; f\left(\phi_{i+1/2,j}, \left(\dfrac{-u_{i+1/2,j}\Delta t}{a_{i+1,j}\Delta x} n_{x,i+1,j}, n_{y,i+1,j}\right)^t\right) & \text{if} \quad u_{i+1/2,j} < 0 \\ u_{i+1/2,j}\Delta t \; f\left(\phi_{i+1/2,j}, \left(\dfrac{u_{i+1/2,j}\Delta t}{a_{i,j}\Delta x} n_{x,i,j}, n_{y,i,j}\right)^t\right) & \text{elsewhere.} \end{cases} \tag{3.7}$$

in the explicit step in Eq. (2.28) where $f()$ permits to determine the volume fraction and is given by the Eq. (3.4). $\phi_{i+1/2,j}$ is the level-set extrapolated in both, space and time, given in the implicit step by:

$$\phi_{i+1/2,j} = \begin{cases} \phi_{i+1,j} - \dfrac{\Delta x}{2}\left(1 + u_{i+1/2,j}\dfrac{\Delta t}{\Delta x}\right) n_{x,i+1,j} & \text{if} \quad u_{i+1/2,j} < 0 \\ \phi_{i,j} + \dfrac{\Delta x}{2}\left(1 - u_{i+1/2,j}\dfrac{\Delta t}{\Delta x}\right) n_{x,i,j} & \text{if} \quad u_{i+1/2,j} > 0 \end{cases}$$

and

$$\phi_{i+1/2,j} = \begin{cases} \phi_{i+1,j} - \dfrac{\Delta x}{2}\left(1 + u_{i+1/2,j}\dfrac{\Delta t}{a_{i+1,j}\Delta x}\right) n_{x,i+1,j} & \text{if} \quad u_{i+1/2,j} < 0 \\ \phi_{i,j} + \dfrac{\Delta x}{2}\left(1 - u_{i+1/2,j}\dfrac{\Delta t}{a_{i,j}\Delta x}\right) n_{x,i,j} & \text{if} \quad u_{i+1/2,j} > 0 \end{cases}$$

in the explicit step.

In the original MCLS method (van der Pijl et al., 2005), the authors used the scheme of Sussman and Puckett (2000) given by Eq. 2.27 and 2.28 for the advection of the volume fractions with the fluxes computed according to Eq. (3.6) in both the implicit and the explicit steps, which leads to a non-consistency of the fractions. A flux redistribution step is therefore proposed in which the over-/undershoots of the volume fraction after advection are redistributed to neigh-

boring cells to conserve mass. This flux redistribution is not needed in our implementation since the volume fractions are advected using the consistent and conservative scheme.

In this chapter, we compare 4 different implementations of the MCLS method where only the level-set advection and reinitialization differ.

3.4 Coupled Level-Set and Volume-of-Fluid method (CLSVOF)

The coupled level-set and volume-of-fluid (CLSVOF) method is the most well-known coupling technique. The pioneering work is from Bourlioux (1995), but it became popular after the work of Sussman and Puckett (2000). Later, Son and Hur (2002) have proposed a slightly different implementation for 2D cartesian and 2D-axisymmetric grid systems, then a fully 3D implementation in a subsequent paper (Son, 2003). A multidirectional implementation of the CLSFVOF on non-uniform grids is proposed by Ningegowda and Premachandran (2014), in which the level-set advection terms are discretized using second-order ENO scheme and the volume-of-fluid is advected using the Edge Matched Flux Polygon Advection (EMFPA) algorithm (López et al., 2004).

The CLSVOF has been the subject of comparison with the level-set method and with volume-of-fluid methods for example in Sussman and Puckett (2000); Wang et al. (2012b); Gerlach et al. (2006) and Wang et al. (2012a). In addition to the classical tests of validation of interface methods, the CLSVOF method has been used for example in the study of bubble formation and/or bubble rising (Ohta et al., 2005; Wang and Tong, 2006; Gerlach et al., 2007; Buwa et al., 2007; Chakraborty et al., 2011, 2013; Albadawi et al., 2013), jet atomization (Menard et al., 2007), boiling (Tomar et al., 2005, 2008), coalescence and breakup processes (Kwakkel et al., 2013), impact of a droplet on a dry surface (Yokoi et al., 2009).

Gerlach et al. (2006) have compared the two surface tension models: Continuous Surface Force (CSF) and Continuous Surface Stress (CSS) on the "spurious currents" test case and concluded that the CLSVOF-CSF behaves slightly better than CLSVOF-CSS. In a subsequent paper, they extended their code to axisymmetric geometries and simulated periodic bubble formation at a submerged orifice (Gerlach et al., 2007).

The CLSVOF algorithm has been detailed in Sussman and Puckett (2000) for axisymmetric and cartesian grid systems. Algorithm 3.4 summarizes the different steps of this method.

3.4.1 Computation of the interface normal

We seek the normal to the interface of the cell (i, j), which can be computed either by solving a minimization problem of the error between the level-set and a linear approximation (Sussman and Puckett, 2000; Menard et al., 2007), or by using the second-order central differences of the level-set (Son and Hur, 2002), or using the local averaging (Vigneaux, 2008).

- In Sussman and Puckett (2000) and Menard et al. (2007), the normal vector is determined by minimizing the error between ϕ and a reconstructed bilinear level-set function $\phi^R_{i,j}$

$$E_{i,j} = \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} \delta(\phi) \left(\phi - \phi^R_{i,j}\right)^2 dxdy$$

  where δ is the Dirac function and $\phi^R_{i,j}$ is defined by

$$\phi^R_{i,j} = n_{x,i,j}(x - x_i) + n_{y,i,j}(y - y_j) + d_{i,j} \tag{3.8}$$

Algorithm 3.4 CLSVOF method

Input: ϕ^n, C^n, $\vec{u}^{n+1/2}$, Δt

$\vec{n} \leftarrow$ Compute the normal to the interface from the level-set Eq. (3.8), or (3.10) (ϕ^n)

$d \leftarrow$ Enforce the distance to satisfy the volume fraction inside the cell (C^n, $\vec{n}$)

$C^* \leftarrow$ Advect the volume-of-fluid in the x- (or y-) direction Eq. (2.27) or (2.28) on §2.3.4 ($\vec{n}$, d, C^n, $u^{n+1/2}$)

$\phi^* \leftarrow$ Advect the level-set in the x- (or y-) direction using Eq. (3.12) or (3.13) (ϕ^n, $u^{n+1/2}$)

$C^* \leftarrow$ Truncate the volume fraction Eq.(3.11) (C^*, ϕ^*)

$\vec{n} \leftarrow$ Compute the normal to the interface from the level-set Eq. (3.8), or (3.10) (ϕ^*)

$d \leftarrow$ Enforce the distance to satisfy the volume fraction inside the cell (C^*, $\vec{n}$)

$C^{n+1} \leftarrow$ Advect the volume-of-fluid in the y- (or x-) direction Eq. (2.28) or (2.27) ($\vec{n}$, d, C^*, $u^{n+1/2}$)

$\phi^{n+1} \leftarrow$ Advect the level-set in the y- (or x-) direction using Eq. (3.13) or (3.12) (ϕ^*, $u^{n+1/2}$)

$C^{n+1} \leftarrow$ Truncate the volume fraction Eq.(3.11) (C^{n+1}, ϕ^{n+1})

$\phi^{n+1} \leftarrow$ Reinitialize the level-set (ϕ^{n+1}, C^{n+1})

$\vec{u}^{n+3/2} \leftarrow$ Advance flow equations (ϕ^{n+1} or C^{n+1} , $\vec{u}^{n+1/2}$)

return ϕ^{n+1}, C^{n+1}, $\vec{u}^{n+3/2}$

where $n_{x,i,j}$, $n_{y,i,j}$ are respectively the normal components following the $x-$ and the $y-$direction and $d_{i,j}$ the distance from the cell center (x_i, y_j) to the interface. The superscript R refers to the reconstructed interface. Using a nine-point stencil, the integral is discretized as (Sussman and Puckett, 2000):

$$E_{i,j} = \sum_{i'=i-1}^{i+1} \sum_{j'=j-1}^{j+1} w_{i'-i,j'-j} \delta_\epsilon(\phi_{i',j'}) \left(\phi_{i',j'} - n_{x,i,j}(x_{i'} - x_i) - n_{y,i,j}(y_{i'} - y_j) - d_{i,j}\right)^2$$

where δ_ϵ is the smoothed Dirac function defined by

$$\delta_\epsilon(x) = \begin{cases} 0, & |x| > \epsilon \\ \dfrac{1}{2\epsilon}\left[1 + \cos\left(\dfrac{\pi x}{\epsilon}\right)\right], & |x| < \epsilon \end{cases}$$

with $\epsilon = \sqrt{2}\Delta x$.

The discrete weights $w_{i'-i,j'-j}$ are chosen to be as $w_{i'-i,j'-j} = 16$ if $i' - i = j' - j = 0$ and $w_{i'-i,j'-j} = 1$ if $i' - i \neq 0$ or $j' - j \neq 0$. An expression of the weights in three dimensions is provided by Jemison et al. (2013).

The minimization of the error reads:

$$\frac{\partial E_{i,j}}{\partial n_{x,i,j}} = \frac{\partial E_{i,j}}{\partial n_{y,i,j}} = \frac{\partial E_{i,j}}{\partial d_{i,j}} = 0$$

and this leads to a resolution of the 3×3 linear system:

$$\begin{bmatrix} \sum\sum whX^2 & \sum\sum whXY & \sum\sum whX \\ \sum\sum whXY & \sum\sum whY^2 & \sum\sum whY \\ \sum\sum whX & \sum\sum whY & \sum\sum wh \end{bmatrix} \begin{bmatrix} n_{x,i,j} \\ n_{y,i,j} \\ d_{i,j} \end{bmatrix} = \begin{bmatrix} \sum\sum wh\phi X \\ \sum\sum wh\phi Y \\ \sum\sum wh\phi \end{bmatrix} \tag{3.9}$$

with the following notations (Menard et al., 2007):

$$\begin{aligned}\sum\sum &\longleftrightarrow \sum_{i'=i-1}^{i+1}\sum_{j'=j-1}^{j+1}\\ wh &\longleftrightarrow w_{i'-i,j'-j}\delta_\epsilon(\phi_{i',j'})\\ X &\longleftrightarrow x_{i'}-x_i\\ Y &\longleftrightarrow y_{i'}-y_j\\ \phi &\longleftrightarrow \phi_{i',j'}\end{aligned}$$

The resolution of this linear system in Eq. (3.9) gives the components of the normal to the interface and the distance. We should then normalize the normal components.

- Son and Hur (2002) have proposed a simple method which consists of computing the normal to the interface directly from the level-set using the second-order central differences scheme.

$$\vec{n} = \left(\frac{\delta\phi}{\delta x}, \frac{\delta\phi}{\delta y}\right) \Big/ \sqrt{\left(\frac{\delta\phi}{\delta x}\right)^2 + \left(\frac{\delta\phi}{\delta y}\right)^2} \tag{3.10}$$

where $(\delta\phi/\delta x)_i = (\phi_{i+1} - \phi_{i-1})/(2\Delta x)$.

Once the normal to the interface is computed, the distance from the interface to the cell center should satisfy the volume fraction inside the cell if this latter is available (in our case, we use Algorithm A.2). If the volume fractions are not available, especially in the first time step where only the level-set field is known, the volume fraction inside the mixed cell (i, j) is computed as:

$$\begin{aligned}C_{i,j} &= \frac{1}{\Delta x\Delta y}\int_{x_{i-1/2}}^{x_{i+1/2}}\int_{y_{j-1/2}}^{y_{j+1/2}} H\left(\phi^R_{i,j}\right)dxdy\\ &= \frac{1}{\Delta x\Delta y}\int_{x_{i-1/2}}^{x_{i+1/2}}\int_{y_{j-1/2}}^{y_{j+1/2}} H\big(n_{x,i,j}(x-x_i) + n_{y,i,j}(y-y_j) + \phi_{i,j}\big)dxdy\end{aligned}$$

where H is the Heaviside function.

Note also that the latter equation, combined with the method to determine the interface normal, is yet another way to compute the volume fractions from the level-set. A similar function, Eq. (3.4), was proposed in the MCLS method and described in §A.3.2. The integral can also be computed with the Algorithm A.1, given the normal from CLSVOF.

3.4.2 Volume-of-fluid advection

For the advection of the volume-of-fluid, we use the conservative and consistent flux splitting detailed on §2.3.4. For the x-direction:

$$\frac{C^*_{i,j} - C^n_{i,j}}{\Delta t} + \frac{u_{i+1/2,j}\widehat{C}_{i+1/2,j} - u_{i-1/2,j}\widehat{C}_{i-1/2,j}}{\Delta x} = C^*_{i,j}\frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} \tag{2.27}$$

and for the y-direction :

$$\frac{C^{n+1}_{i,j} - C^*_{i,j}}{\Delta t} + \frac{v_{i,j+1/2}\widehat{C}^*_{i,j+1/2} - v_{i,j-1/2}\widehat{C}^*_{i,j-1/2}}{\Delta y} = C^*_{i,j}\frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta y} \tag{2.28}$$

This scheme is consistent and conservative, i.e. the fractions are never neither greater than

unity nor negative and the total mass is conserved. However, the volume fraction inside a given cell could be slightly lower than unity or slightly greater than zero due to round-off errors. The remedy to this is to truncate the volume fractions using the level-set as follows:

$$C_{i,j} = \begin{cases} 0 & \text{if } C_{i,j} \leq 0 \text{ or } \phi_{i,j} > \Delta x \\ 1 & \text{if } C_{i,j} \geq 1 \text{ or } \phi_{i,j} < -\Delta x \\ C_{i,j} & \text{otherwise.} \end{cases} \tag{3.11}$$

This truncation may also eliminate the small non-physical fluid particles (floatsam and jetsam).

Of course, the truncation of the volume fractions causes unfortunately a little variation in the total mass (Wang et al., 2009; Li et al., 2012). However, this mass conservation error is considered to be much less than that induced by the level-set method.

3.4.3 Level-set advection

Concerning the level-set advection, we have several options:

- Using directional splitting as in Sussman and Puckett (2000) as follows:

$$\frac{\phi^*_{i,j} - \phi^n_{i,j}}{\Delta t} + \frac{u_{i+1/2,j}\phi^n_{i+1/2,j} - u_{i-1/2,j}\phi^n_{i-1/2,j}}{\Delta x} = \phi^*_{i,j}\frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} \tag{3.12}$$

  For the y-direction:

$$\frac{\phi^{n+1}_{i,j} - \phi^*_{i,j}}{\Delta t} + \frac{v_{i,j+1/2}\phi^*_{i,j+1/2} - v_{i,j-1/2}\phi^*_{i,j-1/2}}{\Delta y} = \phi^*_{i,j}\frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta y} \tag{3.13}$$

  where the level-set at the cell faces is determined by extrapolating the level-set in both space and time (Sussman and Puckett, 2000) . For example, $\phi_{i+1/2,j}$ is determined by upwinding:

$$\phi_{i+1/2,j} = \begin{cases} \phi_{i,j} + \dfrac{\Delta x}{2}\left(1 - u_{i+1/2,j}\dfrac{\Delta t}{\Delta x}\right)\dfrac{\phi_{i+1,j} - \phi_{i-1,j}}{\Delta x} & \text{if } u_{i+1/2,j} > 0 \\ \phi_{i,j} - \dfrac{\Delta x}{2}\left(1 + u_{i+1/2,j}\dfrac{\Delta t}{\Delta x}\right)\dfrac{\phi_{i+2,j} - \phi_{i,j}}{\Delta x} & \text{if } u_{i+1/2,j} < 0 \end{cases} \tag{3.14}$$

- Son and Hur (2002) have used the directional splitting to solve the level-set advection equation as follows:

$$\begin{aligned} \frac{\phi^*_{i,j} - \phi^n_{i,j}}{\Delta t} + u_{i,j}\left.\frac{\partial \phi^n}{\partial x}\right|_{i,j} &= 0 \\ \frac{\phi^{n+1}_{i,j} - \phi^*_{i,j}}{\Delta t} + v_{i,j}\left.\frac{\partial \phi^*}{\partial y}\right|_{i,j} &= 0 \end{aligned} \tag{3.15}$$

  where spatial derivatives are evaluated with 2nd-order ENO schemes using the minmod limiter:

$$\frac{\partial \phi^n}{\partial x} = \frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x} + \text{minmod}\left(\frac{\phi_{i+1,j} + \phi_{i-1,j} - 2\phi_i}{2\Delta x}, \frac{\phi_{i,j} + \phi_{i-2,j} - 2\phi_{i-1,j}}{2\Delta x}\right)$$

if $u_{i+1/2,j} > 0$ or

$$\frac{\partial \phi^n}{\partial x} = \frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x} - \text{minmod}\left(\frac{\phi_{i+1,j} + \phi_{i-1,j} - 2\phi_{i,j}}{2\Delta x}, \frac{\phi_{i,j} + \phi_{i+2,j} - 2\phi_{i+1,j}}{2\Delta x}\right)$$

if $u_{i+1/2,j} < 0$. with

$$\text{minmod}(a,b) = \begin{cases} \text{sign}(a)\min(|a|,|b|) & \text{if } ab > 0 \\ 0 & \text{elsewhere.} \end{cases}$$

The advection of the level-set method in the CLSVOF by Son and Hur (2002) is similar to the level-set that we already tested in the §1.6.1. In principle, any other advection scheme described before could be used.

3.4.4 Reinitialization of the level-set field

After the advection step, the level-set needs to be reinitialized in order to keep the signed-distance property. The reinitialization in the CLSVOF method is performed using the information provided by the level-set itself and the volume-of-fluid (Sussman and Puckett, 2000; Son and Hur, 2002). It is performed on a bandwidth of k cells around the interface.

An interface may pass through a given cell (i,j) if the level-set changes its sign between this given cell and one of its neighbors:

$$\min(\phi_{i,j} \times \phi_{i',j'}) \leq 0 \quad \text{for} \quad i' \in [i-1, i, i+1] \text{ and } j' \in [j-1, j, j+1], (i',j') \neq (i,j) \tag{3.16}$$

The reinitialization procedure is shown on Algorithm 3.5 and explained in detail in Son and Hur (2002). It is based only on the volume fraction field and the linear interface reconstruction in each mixed cell. Examples of the computation of the minimal distance between the cell center and the interface is shown on Fig. 3.3, where the distance is represented by a line segment and the reference fluid by the blue color.

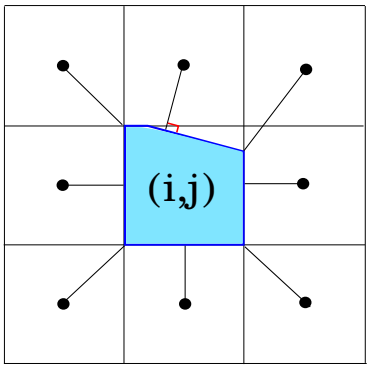

Figure 3.3: Minimal distance between the cell boundaries and a cell center.

Algorithm 3.5 Level-set reinitialization in the CLSVOF method

Input: C, $\vec{n}$, d, k

$\phi \leftarrow \begin{cases} d & \text{if } 0 < C < 1 \\ Sign(\frac{1}{2} - C) \times k\Delta x & \text{Otherwise.} \end{cases}$

for all computational cells (i, j) where Eq. (3.16) is verified **do**

  if ($C_{i,j} = 0$ or $C_{i,j} = 1$) **then** ▷ (i, j)-th cell is single phase in VOF

    for each neighbor cells (i', j') **do** ▷ $i' \in [i-k, i+k]$ and $j' \in [j-k, j+k]$

      $\phi^{temporary}_{i',j'} \leftarrow$ minimal distance from the (i', j') cell center to the nearest point on the (i, j) cell faces

      if $\phi_{i,j}\phi_{i',j'} \leq 0$ **then**

        $\phi_{i',j'} \leftarrow |\phi^{temporary}_{i',j'}| \times Sign(\phi_{i',j'})$

      end if

    end for

  else ▷ (i, j)-th cell is mixed in VOF

    for each neighbor cells (i', j') **do** ▷ $i' \in [i-k, i+k]$ and $j' \in [j-k, j+k]$

      $(x_1, y_1) \in \Omega_{i,j} \leftarrow$ coordinates of the nearest point on cell (i, j) to (i', j') cell center

      $d_1 \leftarrow$ distance from (x_1, y_1) to the linear interface.

      if $\phi_{i',j'} \times d_1 < 0$ **then** ▷ opposite sign of (i', j') and nearest point

        $\phi^{temporary}_{i',j'} \leftarrow$ distance from the (i', j') cell center to the point (x_1, y_1)

      else ▷ same signs for cell (i', j') and nearest point

        $\phi^{temporary}_{i',j'} \leftarrow$ distance from the (i', j') cell center to the linear interface.

      end if

      if $|\phi_{i',j'}| \geq |\phi^{temporary}_{i',j'}|$ **then**

        $\phi_{i',j'} \leftarrow |\phi^{temporary}_{i',j'}| \times Sign(\phi_{i',j'})$

      end if

    end for

  end if

end for

return ϕ

3.5 Coupled Level-set and Moment-Of-Fluid (CLSMOF)

The coupled level-set and moment-of-fluid (CLSMOF) method was first introduced by Jemison et al. (2013). Instead of coupling the level-set with the volume-of-fluid as in the CLSVOF method, the authors coupled the level-set with the moment-of-fluid method with some modifications in the Algorithm 3.4. The main difference consists in the manner the normal to the interface is computed.

The CLSMOF method of Jemison et al. (2013) combines the level-set slope reconstruction from CLSVOF and the MOF approach to reconstruct the interface:

Step 1. An initial estimate of the interface normal, $\vec{n}^{0,1}$, is first determined by minimizing the error, Eq. (3.9), as in the CLSVOF method. The distance d is then adjusted so that the volume fraction of the reconstructed interface corresponds exactly to the target value C^n of the cell under consideration. This is equivalent to the CLSVOF reconstruction.

Step 2. A second normal is computed from the target centroid, $\vec{n}^{0,2} = -\vec{x}_c^{ref}$, and both $\vec{n}^{0,1}$ and $\vec{n}^{0,2}$ are used to initialize the MOF algorithm. The normal which gives the smallest centroid error of Eq. (2.37) is used in the first iteration of the next step.

Step 3. The MOF method is then used to compute a third normal vector, $\vec{n}^{0,3}$, by matching the target centroid $\vec{x}_c^{ref}$ as close as possible to the centroid of the reconstructed cell (see §2.4.1).

Step 4. The final selection between the MOF or CLSVOF reconstruction ($\vec{n}^{0,1}$ and $\vec{n}^{0,3}$) is based on the values of the centroid error E_{MOF}, Eq. (2.37), and the curvature $\kappa(\phi)$ (the latter is computed from the levelset field). When the actual centroid matches closely the target centroid, or in regions of high curvature, the MOF reconstruction is preferred. More specifically, the MOF reconstruction is selected if any of the following equations holds:

$$\kappa(\phi) > \frac{1}{\beta \Delta x} \quad , \quad E_{MOF} < 10^{-8}\Delta x \quad , \quad E_{MOF} > \Delta x/720$$

where $\beta = 6$. Otherwise the CLSVOF reconstruction is used.

3.6 Results and comparison between interface methods

In this section, the coupled level-set/volume-of-fluid methods described above are compared with each other and with the non-coupled level-set and VOF methods using the following test problems: the Zalesak disk and the shearing vortex.

3.6.1 Overview of tested methods

We provide here first the abbreviations used in this section and a brief description of the different methods:

VOF: The volume-of-fluid fractions are advected using the conservative and consistent flux splitting scheme of §2.3.4. The normal to the interface is determined by ELVIRA, then the distance is computed according to the Algorithm A.2. This method is similar to that tested in Chapter 2 where it is given by Algorithm 2.1.

VOF-WY: It is given by Algorithm 2.1, but the volume fractions advection is realized through the fully explicit, conservative and consistent scheme of Weymouth and Yue (2010) detailed on §2.3.5. The interface is reconstructed by computing the normal using ELVIRA, and the distance using the Algorithm A.2. When there is no cell compression/expansion, this method is exactly the same as the **VOF** method.

LS: The level-set method with reinitialization after each advection using the volume constraint reinitialization Eq. (1.50). This latter is discretized in space using the 2nd-order Hamilton-Jacobi ENO scheme while the spatial discretization of the advection equation is performed using the 5th-order WENO-Adv scheme. It is reinitialized after each advection on a band-width of $2\Delta x$ on each side of the interface with a fictitious time step of $\frac{1}{2}\Delta x$.

MCLS/VC: In this coupled method, the level-set equation is solved in the same manner as for the LS method described above which includes the volume constraint (VC) for reinitialization. All other characteristics are detailed in Algorithm 3.2.

MCLS/STD: Similar to the previous one, but the reinitialization of the level-set function is performed without the volume constraint equation.

MCLS/STD/HJ-ENO2: Similar to the previous one, but the advection step for the level-set function is done using the 2nd-order HJ-ENO scheme instead of the 5th-order WENO-Adv scheme. Both the advection and the reinitialization equations are now solved using the faster 2nd-order HJ-ENO scheme.

MCLS/STD/UPWIND1: Similar to the previous one, but both the advection equation and the reinitialization equation are solved using the first-order Upwind scheme.

VOSET−k The one-way coupled method, where the $k = (1, 3, 5, \text{or } 10)$ represents the number of iterations performed to obtain the level-set field. Note that in the interface advection test, the results are strictly identical to those obtained with the VOF method.

CLSVOF-Sussman: The coupled level-set/volume-of-fluid method, in which the interface reconstruction is based on the resolution of the linear system of Eq. (3.9) and the level-set advection equation is solved using Eqs. (3.12-3.14), as recommended by Sussman and Puckett (2000).

CLSVOF-Son: The coupled level-set/volume-of-fluid method where the interface is reconstructed by computing the normal using central differencing of the level-set, while the level-set is advected using the 2nd-order ENO scheme, as recommended by Son and Hur (2002).

MOF: The moment-of-fluid method is advanced in time according to the Algorithm 2.2 given on §2.5.

CLSMOF: The coupled level-set and moment-of-fluid is given by Algorithm 3.4, with the normal computed either by the MOF methods or as in the CLSVOF method as described in §3.5. The level-set is advected in time using Eqs. (3.12-3.14), as recommended by Sussman and Puckett (2000). The moment-of-fluid advection is realized as already described for the **MOF** method.

In the next subsection, the quantitative and qualitative comparison will be shown.

3.6.2 Results and discussion

We have performed simulations for the three test problems already presented in §1.5.2-1.5.4, with the CFL number based on the maximum velocity set to 0.5 for all methods. Table 3.1 summarizes the global error measurements for mass conservation, E_M, and interface location, E_1, defined in §1.5.1. Note that in the following, the figures showing the interface results do not include those obtained with the VOSET method, since they are indiscernible from the VOF results (the interface advection is solved with the same method). However, the global errors for the VOSET methods are included in Table 3.1 since they are computed from the level-set field (using bicubic interpolation with a 1000^2 sub-cells). The error for this method therefore only reflects the accuracy of the algorithm that computes the level-set from the VOF, detailed in §3.2. For comparison, the error induced by this algorithm when applied to the exact solution is also indicated in Table 3.1 (error of the reference field).

Zalesak disk

Fig. 3.4 shows the results for the Zalesak disk after 1, 2, 4 and 8 complete revolutions on a 100×100 grid. All the methods conserve the general form of the Zalesak disk even after 8 revolutions, with smoothed angles. The VOSET, VOF, CLSVOF and CLSMOF methods

produce a slight displacement of the top of the slab in the flow direction, while the level-set method displaces it slightly in the counter-flow direction. The MOF method is less sensitive to the number of the revolutions than the other methods.

The errors, E_M, in Table 3.1 show that the VOF and MOF methods conserve, as expected, global mass within machine precision (the error is based here on the volume fractions). In the case of the VOSET methods, the interface is advanced with the VOF method, but the level-set is computed at each time step from the volume fractions and the global mass error is computed from the level-set field. The global mass error induced by this conversion is, however, relatively small.

The CLSVOF methods have mass errors, computed from the level-set field, comparable to VOSET and location errors comparable to both VOF and VOSET. Since the interfaces of CLSVOF in Fig. 3.4 are very close to that observed with the VOF methods, it seems that the interface evolution with these methods are mainly governed by the volume-of-fluid advection schemes, rather than the level-set schemes.

Shearing vortex

The results for the single vortex problem at maximum stretch at $t = T/2 = 3$ are shown in Fig. 3.5. As already observed in chapters 1 and 2, the level-set method tends to thicken the vortex filament, while the VOF and MOF methods cause the formation of small bubbles due to the mass which needs to be conserved and the fact that the grid is coarse and more than one interface passes through reconstruction blocks of 3×3 cells. The coupled methods also show the breakup of the filament, with different degrees of regularity and length of the non-fractured filament. The MOF and CLSMOF methods produce results with the longest part of the non-fractured vortex. They produce also the smallest location errors. These methods clearly profit from the use of the centroid information, in addition to the volume fraction, for the interface reconstruction.

The results obtained with the CLSVOF-Son method are very close to that of the VOF method, and is slightly better than that given by the CLSVOF-Sussman method. The interface location errors, E_1, are slightly worse than for the VOF methods, while the global mass is comparable to that of the VOSET method (since this error is calculated from the level-set). The global mass error for the VOF and MOF methods are again very small, since computed directly from the volume fractions.

Furthermore, the MCLS methods all produce very thin filaments, close to the exact solution, but with very irregular interfaces. This may be a problem when for the computation of the surface tension term when combined with the Navier-Stokes equations. The global mass and location error are larger than all other methods, except the level-set method.

When inverting the velocity direction and coming back to the initial position, the interface methods should reproduce the initial interface, which is a circle. However, the interface obtained at time $t = T = 6$ deviates slightly from a circle as shown in Fig. 3.6. The worst results are given by the level-set method, mainly due to the important gain of mass, while the best results are obtained with the CLSMOF and MOF methods.

Table 3.1: Mass and location errors for the Zalesak disk, single vortex and time-reversed single vortex for all the interface methods

Method	**Zalesak disk Grid 100^2**			**Shearing vortex Grid 100^2**						**Time-reversed single vortex Grid 64^2**					
	t = 6.28			t = T/2 = 3			t = T = 6			t = T/2 = 1			t = T = 2		
	$E_M(\%)$	$E_1(\%)$	CPUt	$E_M(\%)$	$E_1(\%)$	CPUt	$E_M(\%)$	$E_1(\%)$	CPUt	$E_M(\%)$	$E_1(\%)$	CPUt	$E_M(\%)$	$E_1(\%)$	CPUt
Error of the reference field	0.01	–	–	1.88	–	–	5.10^{-4}	–	–	0.007	–	–	5.10^{-4}	–	–
VOF	1.91×10^{-7}	2.52	0.59	1.05×10^{-7}	10.80		1.05×10^{-7}	7.15	1.99	-1.96×10^{-14}	0.98		-9.81×10^{-14}	0.99	0.12
VOF-WY	1.91×10^{-7}	2.52	0.56	1.05×10^{-7}	11.30		1.05×10^{-7}	6.81	1.91	-7.85×10^{-14}	0.97		-5.89×10^{-14}	1.00	0.12
MOF	1.91×10^{-7}	1.03	0.69	1.05×10^{-7}	4.69		1.05×10^{-7}	2.57	2.19	2.68×10^{-14}	0.73		2.68×10^{-11}	0.26	0.15
VOSET-1	-3.91×10^{-2}	2.50	1.88	2.08	11.05		6.67×10^{-2}	7.35	7.93	-2.98×10^{-2}	1.07		-0.25	1.29	0.43
VOSET-3	-4.00×10^{-2}	2.50	3.16	2.18	11.12		8.96×10^{-2}	7.38	14.04	-2.93×10^{-2}	1.07		-0.28	1.34	0.76
VOSET-5	-4.00×10^{-2}	2.50	4.54	2.17	11.13		8.37×10^{-2}	7.38	20.26	-2.92×10^{-2}	1.07		-0.28	1.34	1.07
VOSET-10	-4.00×10^{-2}	2.50	7.75	2.17	11.14		8.39×10^{-2}	7.38	35.42	-2.92×10^{-2}	1.07		-0.28	1.34	1.87
CLSVOF-Son	-3.45×10^{-2}	2.12	0.78	1.80	12.86		-1.31×10^{-2}	7.91	2.50	-3.84×10^{-2}	0.87		-6.62×10^{-2}	0.53	0.16
CLSVOF-Sussman	-4.23×10^{-2}	2.42	0.95	1.38	18.26		-3.81×10^{-2}	11.38	2.91	-3.87×10^{-2}	1.06		-7.41×10^{-2}	0.92	0.19
CLSMOF	-4.20×10^{-2}	1.65	1.73	1.63	5.77		-7.66×10^{-2}	2.74	5.29	-6.46×10^{-2}	0.75		-6.77×10^{-2}	0.33	0.37
MCLS/VC	0.22	2.42	19.96	-5.08	17.72		-2.48×10^{-2}	10.73	36.90	-2.03	3.19		-1.13×10^{-2}	1.87	2.29
MCLS/STD	0.24	2.48	18.69	-4.20	20.04		-0.24	8.99	35.17	-0.94	2.00		0.14	1.37	2.10
MCLS/STD/HJ-ENO2	0.19	2.34	16.01	-4.21	17.47		-7.27×10^{-2}	6.67	28.58	-0.86	2.00		8.86×10^{-2}	1.34	1.56
MCLS/STD/UPWIND1	0.28	2.85	4.65	-5.62	15.34		-0.11	6.24	10.7	-1.22	2.07		2.33×10^{-2}	1.40	0.82
LS	-0.96	2.55	16.21	60.26	59.19		72.24	73.66	29.11	-0.25	2.14		-0.34	3.46	1.68

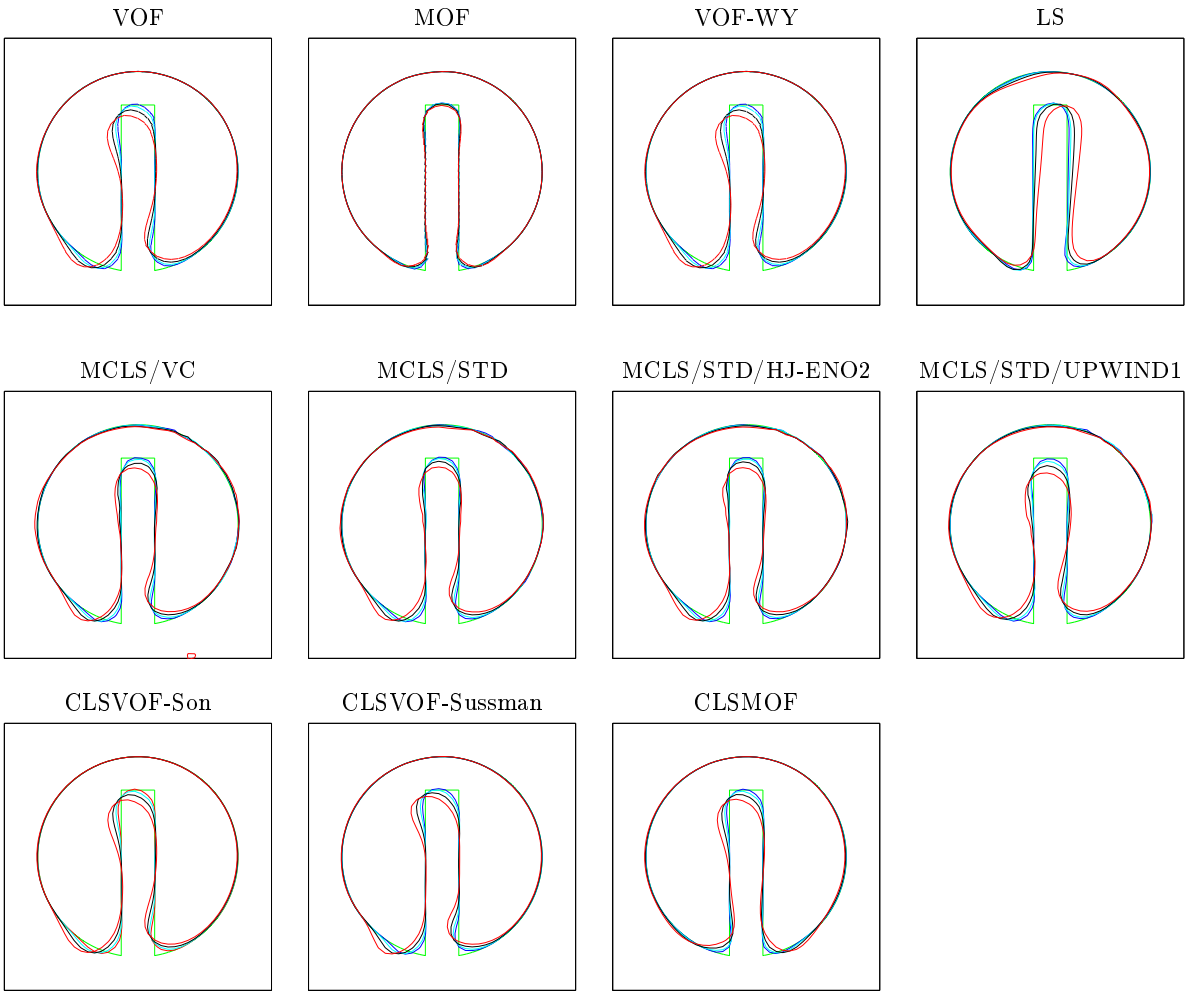

Figure 3.4: Interface reconstruction from the level-set contour for the Zalesak disk calculated on a 100×100 grid with CFL $= 0.5$. Different colors correspond to the number of complete revolutions: 1 (blue), 2 (cyan), 4 (black), 8 (red). The green line indicates the exact solution.

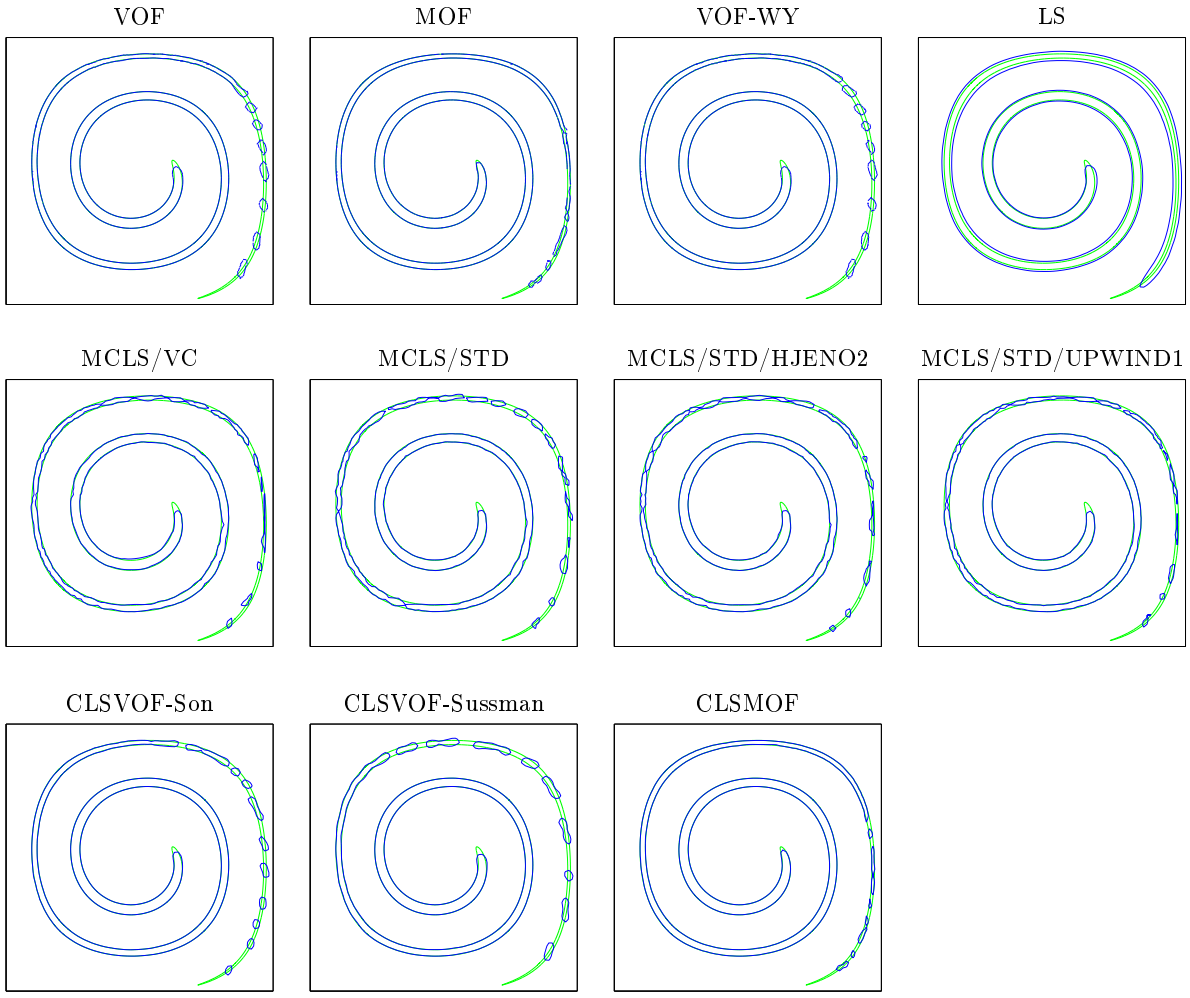

Figure 3.5: Interface reconstruction from the level-set contour for the single Vortex problem calculated at time $t = T/2 = 3$ (maximum stretch) on a 100×100 grid with CFL $= 0.5$. The green line indicates the exact solution.

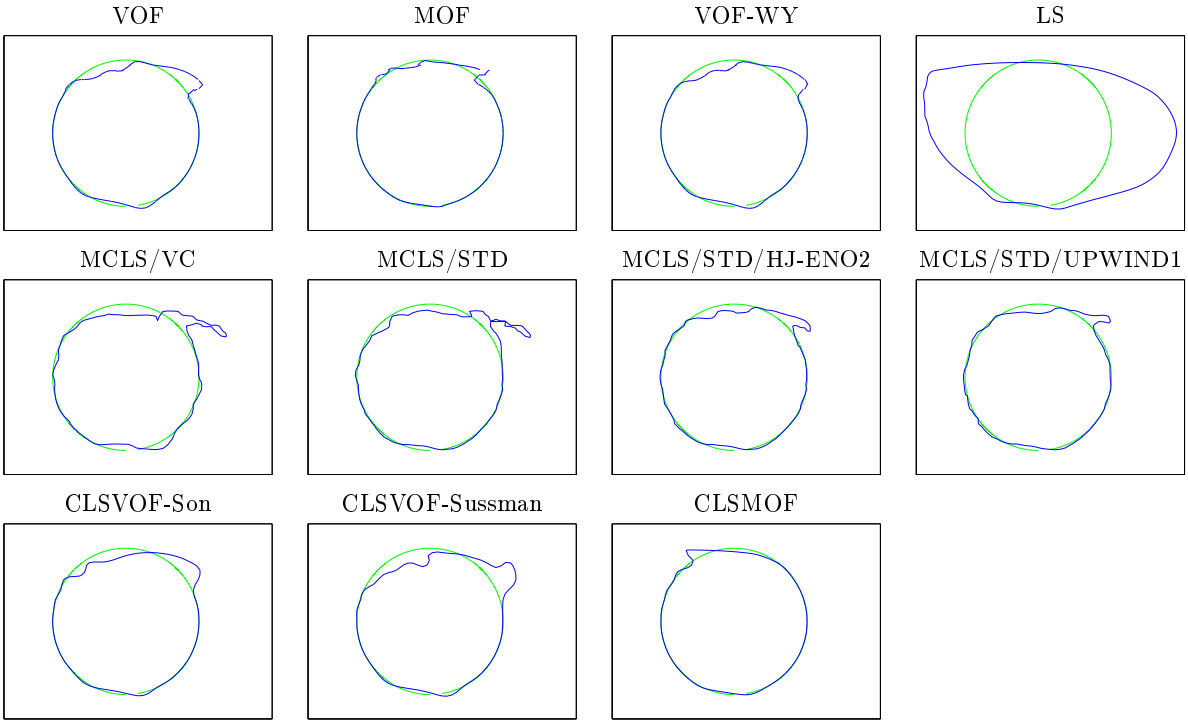

Figure 3.6: Interface reconstruction from the level-set contour for the single Vortex problem calculated at time $t = T = 6$ (end of reversed cycle) on a 100×100 grid with CFL $= 0.5$. The green line indicates the exact solution.

3.6.3 Rate of convergence

To determine the convergence rate, we have chosen only the methods that gave the best results in the previous section, namely:

- LS, VOF, MOF, CLSVOF-Son, CLSVOF-Sussman and CLSMOF
- VOSET with 3 geometric iterations
- MCLS with the standard level-set reinitialization, and the 2nd-order HJ-ENO scheme for the spatial discretization of the advection and the reinitialization equations. The level-set is reinitialized after each advection.

For the Zalesak disk problem, the location error is plotted on Fig. 3.7 for the CFL numbers varying in the range 0.25 to 1.0. The CFL variation has not much effect on the errors, except that for most methods the location error increases when reducing the CFL number. None of the methods show a second-order behavior on fine grids. This is mainly due to the presence of the sharp angles. The MOF method remains the method yielding the least location error in the range of tested grid sizes, closely followed by the CLSMOF method. The worst result is obtained with the LS and MCLS methods.

For the shearing vortex, the location error is plotted at maximum stretch on Fig. 3.8 . A nearly second-order convergence is now observed for nearly all methods and CFL numbers. The best results are again obtained with the MOF and CLSMOF method, while the LS method gives the highest errors. The MCLS method seems to do a fairly good job, but this is mainly due to the mean thickness of the filament, which is close to that of the exact solution. However, as mentioned above, the irregularity of the interface might be a problem when calculating the curvature (not included in the present tests).

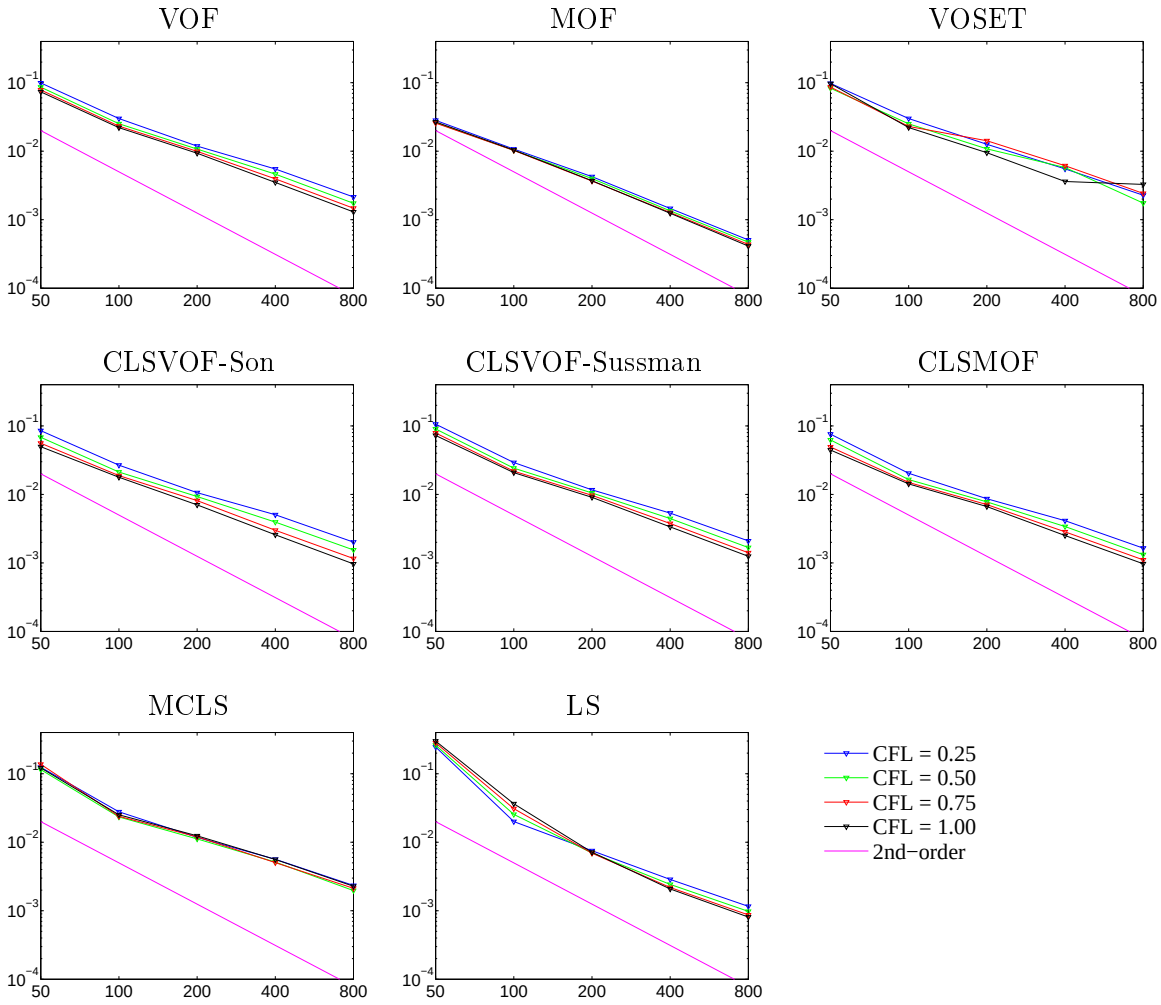

Figure 3.7: Zalesak disk after one full revolution. Location error E_1 as a function of the grid size (in one direction). Different colors correspond to different values of the CFL number. The pink line indicates the slope of a second-order method.

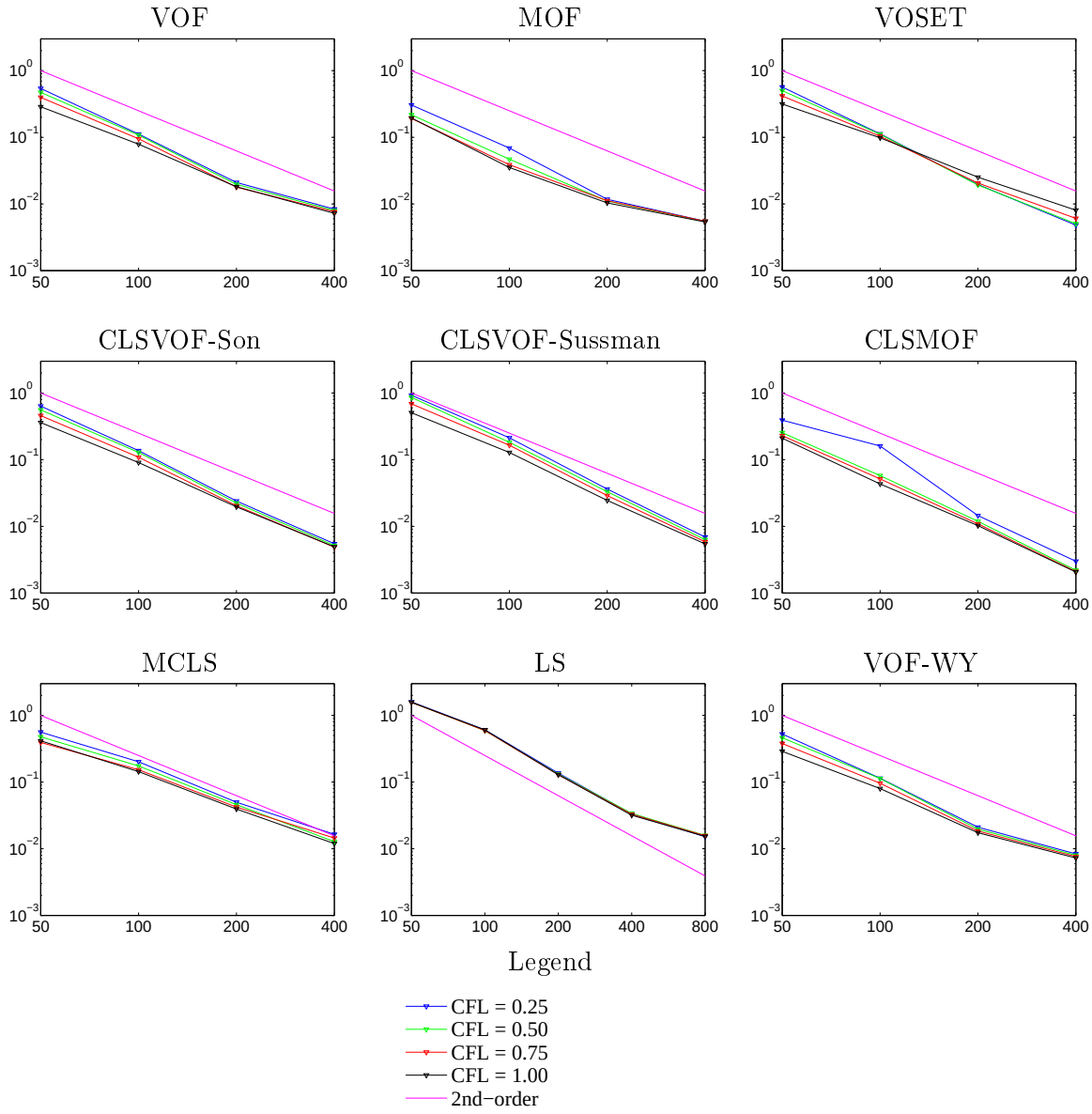

Figure 3.8: Single vortex at $t = T/2 = 3$. Location error E_1 as a function of the grid size (in one direction). Different colors correspond to different values of the CFL number. The pink line indicates the slope of a second-order method.

3.7 Conclusions

The mass error for the coupled methods, except the MCLS, are very small, and mainly due to the calculation of the mass from the level-set field. Mass is therefore correctly conserved by these methods. However, the MCLS method does not seem to conserve the total mass very well, which is most probably due to the accumulation of errors while determining the volume fractions field from the level-set field at each time step.

The MOF and CLSMOF methods are fast and more accurate than all other methods for all test cases. The CLSVOF-Son method is also more accurate than the CLSVOF-Sussman method for all the presented test cases. In fact, in these interface advection tests, the CLSVOF and CLSMOF methods seem to produce results that are very similar to that obtained with their uncoupled versions, respectively the VOF and MOF methods. They have the advantage over the uncoupled methods to produce also the level-set field. It remains to be seen, however, if this advantage can be exploited when combined with the Navier-Stokes equations.

The MCLS/STD/HJENO2 and the MCLS/STD/UPWIND1 methods are more accurate than the other MCLS tested variants. The overall results are, however, less accurate than the other methods. Only the level-set method seems to produce less accurate results. For the VOSET method, the interface is advected with the volume-of-fluid method, and therefore the present results cannot evaluate the benefits of this methods. It is shown, nonetheless, that only a few iterations for the level-set calculation seem to be sufficient to obtain an accurate results.

The CPU-time for each method is also represented in Table 3.1. It shows that the level-set and the MCLS methods need more computational resources than the other methods, which is in our case due to the reinitialization and advection that are performed on the whole domain (we have not used the local level-set method).

Chapter 4

Incompressible two-phase flow solver

4.1 Introduction

In the previous chapters, we have investigated different implementations of interface capturing methods based on Eulerian approaches, and compared the performances of these methods in accurately transporting an interface in an imposed velocity field.

In realistic flows, the velocity field is not prescribed but determined by solving the Navier-Stokes equations. The interface methods (level-set and volume-of-fluid) allow a one fluid approach for non-miscible fluids. The interface method predicts the location of the interface, and allows the evaluation of the fluid properties and the application of the jump conditions at the interface for the solution of the Navier-Stokes equations. On the other hand, the Navier-Stokes solver will provide the velocity field which is used to predict the evolution of the interface.

The aim of this chapter is to describe and test different implementations of the coupling between the interface methods and the Navier-Stokes solver for two-phase incompressible flows. Section 4.2 gives an overview of algorithms used to compute the flow equations in combination with the level-set and/or the VOF interface methods. The literature on this subject is vast and we do not claim to give an exhaustive overview. Instead, we focus on major issues, on the most popular methods and those who fit best in our finite-difference framework. Nonetheless, the number of possibilities to compose a numerical method for two-phase flows is practically unlimited. We have selected a limited number of methods, inspired largely from other studies and popular codes, which are described in detail in §4.3. Finally, in §4.4 we describe the numerical results obtained with these methods in some well-known test problems.

4.2 Navier-Stokes equations

The Navier-Stokes equations can be written in the conservative and non-conservative formulations. The latter is generally preferred in problems with non-miscible two-phase flows, probably since it avoids computation of gradients in the term $\nabla \cdot (\rho \vec{u}\vec{u})$ with jumps in the density. For an incompressible fluid, the Navier-Stokes equations read:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = -\frac{\nabla p}{\rho} + \frac{1}{\rho} \nabla \cdot \mu \left[(\nabla \vec{u}) + (\nabla \vec{u})^T \right] + \vec{f_\sigma} + \vec{g} \tag{4.1}$$

$$\nabla \cdot \vec{u} = 0 \tag{4.2}$$

where p is the dynamic pressure, ρ and μ are the density and dynamic viscosity of the fluid, $\vec{f_\sigma}$ is the continuous surface tension force (see §4.2.2) and $\vec{g}$ is the acceleration due to gravity.

We have limited this work to 2D problems only and use a staggered grid, with grid lines aligned to the coordinate axes, where the scalars are defined on cell centers and the velocity vectors on the cell faces as shown on Fig. 4.1. A grid cell $\Omega_{i,j}$ is defined by the region $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$. We will note the cell size in the x- and y-direction by respectively $\Delta x = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$ and $\Delta y = y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}}$. A uniform grid is used in this work, such that Δx and Δy are constants. Computational domain boundaries are always defined on the cell boundaries (for example, the x-coordinates of the vertical boundaries are $x_{\frac{1}{2}}$ or $x_{N+\frac{1}{2}}$ with N the number of cells in this direction).

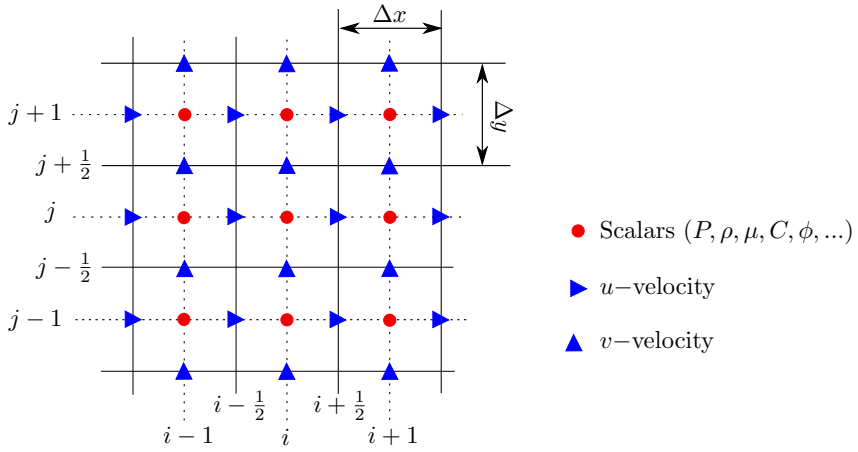

Figure 4.1: Staggered (MAC) grid for the flow solver

4.2.1 Fluid properties

Let ϕ be the level-set function such that $\phi = 0$ defines the liquid-gas interface and $\phi > 0$ and $\phi < 0$ define respectively the liquid and gas phase. We will consider two immiscible fluids with constant fluid properties, ρ_g and μ_g in the gas phase and ρ_l and μ_l in the liquid phase. The density and viscosity of the fluid may then be defined as

$$\rho(\phi) = \rho_g + (\rho_l - \rho_g)H(\phi) \tag{4.3}$$
$$\mu(\phi) = \mu_g + (\mu_l - \mu_g)H(\phi) \tag{4.4}$$

where $H(\phi)$ is the Heaviside function. In the VOF-based methods, the fluid properties are generally given by a volume-fraction weighted average (Puckett et al., 1997; Popinet, 2003):

$$\rho(C) = \rho_g + (\rho_l - \rho_g)C$$
$$\mu(C) = \mu_g + (\mu_l - \mu_g)C$$

The Heaviside function leads to a discontinuity in the fluid properties at the interface, and thereby to many numerical difficulties when using finite-difference approximations in a one-fluid approach. To avoid this problem, a smoothed Heaviside function is often employed to compute the fluid properties. In volume-of-fluid methods, regularization of the fluid properties is obtained by smoothing the volume fractions C.

4.2.1.1 Smoothed Heaviside and delta functions

A smoothed Heaviside function is sometimes used for the computation of ρ and/or μ (Eqs.4.3 and/or 4.4). The smoothed Heaviside function for fluid properties is often defined as (Sussman et al., 1994; van der Pijl et al., 2005):

$$H_\epsilon(x) = \begin{cases} 0, & x < -\epsilon \\ \frac{1}{2}\left[1 + \sin\left(\frac{\pi x}{2\epsilon}\right)\right], & |x| < \epsilon \\ 1, & x > \epsilon \end{cases}$$

where most often $\epsilon = 3\Delta x/2$. A slightly different function can also be used (Chang et al., 1996; Sussman et al., 1998; Sussman and Puckett, 2000; Son and Hur, 2002):

$$H_\epsilon(x) = \begin{cases} 0, & x < -\epsilon \\ \frac{1}{2}\left[1 + \frac{x}{\epsilon} + \frac{1}{\pi}\sin\left(\frac{\pi x}{\epsilon}\right)\right], & |x| < \epsilon \\ 1, & x > \epsilon \end{cases} \tag{4.5}$$

with $\epsilon = 3\Delta x/2$. Furthermore, to regularize the continuous surface tension force, a smoothed delta function is often used (Peskin, 1977; Unverdi and Tryggvason, 1992; Sussman et al., 1994; Chang et al., 1996):

$$\delta_\epsilon(x) = \begin{cases} 0, & |x| > \epsilon \\ \frac{1}{2\epsilon}\left[1 + \cos\left(\frac{\pi x}{\epsilon}\right)\right], & |x| < \epsilon \end{cases} \tag{4.6}$$

which is in fact the derivative of Eq. (4.5), $\delta_\epsilon = dH_\epsilon/dx$.

4.2.1.2 Smoothed volume fraction

It is generally preferable for both the accuracy and the stability of the method to use a smoothed version of the VOF function (Denner and van Wachem, 2013). A popular smoothing kernel is the K_8-kernel, defined by

$$K_8(r) = A\left[1 - \left(\frac{r^2}{\epsilon^2}\right)\right]^4 \quad \text{for} \quad r \leq \epsilon \tag{4.7}$$

and $K_8(r) = 0$ for $r > \epsilon$, where A is a normalization constant. On a 2D Cartesian uniform grid, this kernel would be implemented as:

$$\widetilde{C}_{i,j} = A\sum_{k=-M}^{M}\sum_{l=-M}^{M}\left[1 - \left(\frac{(k\Delta x)^2 + (l\Delta y)^2}{\epsilon^2}\right)\right]^4 C_{i+k,j+l}$$

with $M = \text{ceiling}(\epsilon/h)$ and A is defined by

$$A\sum_{k=-M}^{M}\sum_{l=-M}^{M}\left[1 - \left(\frac{(k\Delta x)^2 + (l\Delta y)^2}{\epsilon^2}\right)\right]^4 = 1$$

Another possibility is to use vertex averaging, in which case the smoothing kernel is defined by:

$$\widetilde{C}_{i,j} = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} C \tag{4.8}$$

Alternatively, the cosine delta function of Eq. 4.6 could be used as a smoothing kernel with $\epsilon = 3\Delta x/2$.

4.2.1.3 Smoothing by averaging

Vigneaux (2008) proposed a set of local averaging operations to compute fluid properties at different grid locations. More details on how this is done are provided in §4.3.3 and §4.3.5. The subsequent averaging steps can be regarded as a manner of smoothing the density and the viscosity.

4.2.2 Surface tension

In an inviscid flow with constant surface tension, the pressure jump condition at the interface (see for example Tryggvason et al., 2011) is reduced to the well-known Laplace formula, $p_l - p_g = \sigma\kappa$, which can be interpreted as a normal surface force per unit area given by $\sigma\kappa\vec{n}$. In viscous flows, additional terms have to be taken into account related to the jump in the viscosity (Kang et al., 2000). Nonetheless, the surface tension term often represents the major contribution to the pressure jump. Therefore, many numerical methods only take this term into account, and we will adopt the same approach. The continuous surface force approach (CSF) (Brackbill et al., 1992) rewrites the interface surface force as a volume force:

$$\vec{f_\sigma} = \frac{\sigma\kappa\vec{n}\delta_s}{\frac{1}{2}(\rho_l + \rho_g)}$$

where δ_s is a delta function concentrated at the interface surface[1]. In the continuum limit, we can write $\vec{n}\delta_s = -\nabla c$ where c is a smoothed color function varying from 0 to 1, for example the smoothed Heaviside function $H_\epsilon(\phi)$.

Level-set approach

According to Chang et al. (1996), since $\nabla H(\phi) = \delta(\phi)\nabla\phi$, the surface tension force can be expressed as:

$$\vec{f_\sigma} = \frac{1}{\rho}\sigma\kappa\delta(\phi)\nabla(\phi)$$

The density is evaluated using one of the methods described in §4.2.1. However, some authors (for example van der Pijl et al., 2005) prefer using the mean density, $(\rho_g + \rho_l)/2$. This approach tends to shift the surface tension force towards the heavier fluid, thereby stabilizing the numerical method. However, it is incompatible with the so-called balanced force approach (Francois et al., 2006), which states that the density in the pressure and surface tension terms should be the same in order to obtain a correct balance between these forces.

When ϕ defines the liquid and vapor phase as $\phi > 0$ and $\phi < 0$ respectively, the normal vector $\vec{n}$ and curvature of the interface κ may be calculated from the level-set field using

$$\begin{aligned} \vec{n} &= \frac{\nabla\phi}{|\nabla\phi|} \\ \kappa &= -\nabla\cdot\vec{n} \end{aligned} \tag{4.9}$$

[1] The delta function δ_s can be interpreted as follows. Whenever a volume integral contains the Dirac function δ_s, it may be replaced by the surface integral, in other words $\int_V f(\vec{x})\delta_s dV \equiv \int_S f(x)dS$ where S is the interface surface in V. In the continuum limit, δ_s designates the surface area per unit volume.

With these definitions, the normal vector points in the direction of the liquid phase.

Chang et al. (1996) used an expression for the curvature based on only first- and second-order spatial derivatives of the level-set function

$$\kappa = -\frac{\phi_y^2\phi_{xx} - 2\phi_x\phi_y\phi_{xy} + \phi_x^2\phi_{yy}}{(\phi_x^2 + \phi_y^2)^{3/2}}$$

applicable to two dimensional cases. An expression in three dimensions can be found in Osher and Fedkiw (2002, p.12).

The advection and reinitialization steps introduce small perturbations on the level-set field which tend to degrade the accuracy of curvature estimators based on finite differences. When using central-difference discretizations to compute the curvature from the normal divergence relation (Eq. 4.34), it is often preferable to add some smoothing in order to reduce the influence of spurious oscillation on the level-set field. In Sussman et al. (1999), the curvature is computed from Eq. (4.9) using a central difference scheme based on a compact 3×3 stencil instead of the 9-point larger stencil of standard central discretization. Vigneaux (2008) also used Eq. (4.9) to compute the curvature at the cell faces. Smoothing of the level-set function is obtained by applying a series of averaging procedures, resulting in a rather large stencil of 4×4 for each velocity node.

In order to obtain a second-order accurate curvature on a slightly perturbed level-set field, a least-square method is proposed by Marchandise et al. (2007). A multi-dimensional linear or quadratic expansion of the level-set function is constructed at each grid point near the interface. The coefficients of this expansion are determined from an evaluation of the polynomial function on a stencil of neighboring grid points. The stencil size exceeds the number of coefficients (resp. 3 and 6 for linear and quadratic expansions in 2D, and 4 and 10 in 3D) and the overdetermined system is solved using a least-squares method. On a two-dimensional uniform Cartesian grid, this method leads to the following convolution kernels for the spatial derivatives in the x- and y-direction using a 3×3 stencil:

$$\frac{1}{6}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}; \frac{1}{6}\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}; \tag{4.10}$$

and with a 5×5-point stencil:

$$\frac{1}{50}\begin{bmatrix} -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \end{bmatrix}; \frac{1}{50}\begin{bmatrix} -2 & -2 & -2 & -2 & -2 \\ -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \end{bmatrix};$$

These kernels are used to compute first the normal vector $\vec{n}$ from the level-set field and then a second time to evaluate the curvature from Eq. (4.9). The averaging effect of the larger stencil makes the least-squares method less sensitive to small perturbations. A truly second-order accurate curvature is then obtained, in contrast to other methods based on finite-difference operators (Marchandise et al., 2007; Desjardins et al., 2008).

Volume-of-fluid approach

A second-order accurate curvature estimation from the VOF function can be obtained using the Height Function (HF) method, developed initially by Helmsen et al. (1997) (see for example

Francois et al. (2006); Puckett (2010); Bornia et al. (2011) for a summary of recent developments). In the method by Sussman (2003), the curvature is computed from a height function reconstructed on a 3×7 stencil centered around the cell containing the interface. The orientation of the height function, $h(x)$ or $h(y)$, and the stencil is determined by the largest component of the interface normal. A discrete approximation of the height function is then obtained by a column-wise summation of the volume fractions on the stencil. The curvature is then computed from

$$\kappa = \frac{h''}{(1 + h'^2)^{3/2}}$$

where the spatial derivatives are computed with standard central difference schemes. For example when the columns are oriented in the y-direction we have $h(x)$ who's derivatives are approximated by $h' = (h_{i+1} - h_{i-1})/(2\Delta x)$ and $h'' = (h_{i+1} - 2h_i + h_{i-1})/(\Delta x^2)$. Cummins et al. (2005) tested the Height Function method against two other approaches based on finite-difference approximations, applied to either the smoothed volume-of-fluid function or the reconstructed signed-distance function (although they did not calculate the latter as the exact signed-distance function to the reconstructed interface as in the CLSVOF method). The height function showed superior accuracy and approximately second-order grid convergence in static interface tests, except when the interface structures are not well resolved. The inaccuracy of the Height Function method in regions with small-scale interface structures or high curvatures has been mentioned by other authors. For example, Hernández et al. (2008) proposed an improvement by adapting the stencil according to the local volume fraction field and introducing small local corrections of the volume fraction. Additional improvements of the Height Function method are proposed in subsequent papers (López et al., 2009; López and Hernández, 2010).

In an independent study, Popinet (2009) also pointed out that the 3×7 stencil is not always an appropriate choice for under-resolved interface structures, for example when the radius is of the order of the mesh size or when the two interfaces are separated by only a few grid points. He proposed an algorithm that adapts the stencil according to the local layout of the interface. The sizes of the columns are reduced or increased in order to contain both sides of the interface, without crossing two interfaces. When a consistent stencil is not found, the method falls back to a parabolic curve fitting approach. Numerical results show that the method, used in the CFD freeware Gerris (Popinet) and Basilisk (Popinet et al.), is robust and exhibits second-order behavior for a wide range of curvature scales.

Finally, in order to reduce the so-called spurious currents in surface-tension dominated flows, the balanced-force algorithm was developed by Francois et al. (2006) in which the surface tension term is discretized in exactly the same manner as the pressure gradient. Their method is based on the VOF-CSF methodology in which the surface term is modeled as:

$$\vec{f_\sigma} = \frac{1}{\rho} \sigma \kappa \nabla \widetilde{C}$$

where $\widetilde{C}$ is either a smoothed or unsmoothed version of the volume-of-fluid function C (VOF methods), or the smoothed Heaviside $\widetilde{C} = H_\epsilon(\phi)$ (level-set methods). Denner and van Wachem (2013) showed that in the VOF-based methods, smoothing of the volume fraction does not contribute to the accuracy and stability of the method and should be avoided. To obtain a balance by pressure and surface tension forces, it is necessary (in addition to an accurate curvature estimation) that, (1) the density at the cell face is the same as that used in the pressure gradient, (2) the discretization of the gradient $\nabla \widetilde{C}$ is equivalent to that for the pressure gradient.

Yet another approach to compute the surface tension term is the so-called Continuous Surface

Stress (CSS) (Lafaurie et al., 1994; Gueyffier et al., 1999) in which the surface tension is modelled as an additional stress term:

$$\vec{f}_\sigma = -\frac{1}{\rho}\nabla \cdot \overline{\overline{T}}$$

with

$$\overline{\overline{T}} = -\sigma(\overline{\overline{I}} - \vec{n} \otimes \vec{n})|\nabla H|$$

This method does not require the explicit computation of the curvature. When applied as a part of the VOF method, the delta function $|\nabla H| = |\nabla \widetilde{C}|$ should be computed from the smoothed function $\widetilde{C}$. Typical smoothing kernels of size 3×3 and 5×5 are

$$\frac{1}{36}\begin{bmatrix} 1 & 4 & 1 \\ 4 & 16 & 4 \\ 1 & 4 & 1 \end{bmatrix}; \frac{1}{3721}\begin{bmatrix} 4 & 30 & 54 & 30 & 4 \\ 30 & 225 & 405 & 225 & 30 \\ 54 & 405 & 729 & 405 & 54 \\ 30 & 225 & 405 & 225 & 30 \\ 4 & 30 & 54 & 30 & 4 \end{bmatrix};$$

The advantage of this approach is that the surface tension term is now in conservative form and that variations of the surface tension σ are properly taken into account. Not many studies have adopted this approach, however, and we have not included the CSS method in our numerical tests.

4.2.3 Projection methods

One of the difficulties in solving the Navier-Stokes equation (4.1) is the coupling between velocity and pressure by the incompressibility constraint (4.2). The projection method, initially proposed for uniform-density flows by Chorin (1986), efficiently decouples the computation of the velocity and the pressure. For two-phase flows, the jump of the density and the surface tension force at the interface needs to be taken into account. This section describes several projection methods that have been employed for interface problems.

In Bell and Marcus (1992), a projection scheme is proposed for variable-density flows without interfaces in which a temporary velocity and density field, $\vec{u}^*$ and ρ^*, is computed using the time-lagged pressure gradient:

$$\begin{aligned} &\frac{\rho^* - \rho^n}{\Delta t} + [\vec{u}^n \cdot \nabla \rho]^{n+\frac{1}{2}} = 0 \\ &\frac{\vec{u}^* - \vec{u}^n}{\Delta t} + [\vec{u}^n \cdot \nabla \vec{u}]^{n+\frac{1}{2}} = \frac{1}{\rho^{n+\frac{1}{2}}}\left[-\nabla p^{n-\frac{1}{2}} + \nabla \cdot \mu^{n+\frac{1}{2}}\left[(\nabla \vec{u}^*) + (\nabla \vec{u}^n)^T\right]\right] + \vec{g} \end{aligned}$$

where $\rho^{n+\frac{1}{2}} = (\rho^n + \rho^{n+1})/2$ and Δt is the time step. Note that part of the viscous terms are treated implicitly, but these equations can be solved efficiently using standard linear solvers. The temporary velocity $\vec{u}^*$ does not fulfill the incompressibility constraint (4.2). The projection step which corrects the velocity field to obtain an incompressible flow is given by:

$$\begin{aligned} \frac{\vec{u}^{n+1} - \vec{u}^*}{\Delta t} &= -\frac{1}{\rho^{n+\frac{1}{2}}}\nabla\left(p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}\right) \\ \nabla \cdot \vec{u}^{n+1} &= 0 \end{aligned}$$

Applying the divergence to the first and inserting the second equation leads to the equation for

the pressure:

$$\nabla \cdot \left(\frac{1}{\rho^{n+\frac{1}{2}}} \nabla \left(p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}} \right) \right) = \frac{1}{\Delta t} \left(\nabla \cdot \vec{u}^* \right)$$

This is a variable-coefficient Poisson equation, which will consume a large portion of the computational time. This equation can be solved for example with Krylov-based solvers, multi-grid solvers or preconditioned conjugate gradient methods (Golub and Loan, 1989). This projection scheme is said *incremental*, since the pressure of the previous time step is used in the provisional velocity equation and an equation for the increment of the pressure is solved.

In Puckett et al. (1997) and Sussman and Puckett (2000), this scheme is adapted to two-phase flows. First, the interface is advected from t^n to t^{n+1}. Flow properties at $t^{n+\frac{1}{2}}$ are then evaluated from

$$\begin{aligned} \phi^{n+\frac{1}{2}} &= (\phi^n + \phi^{n+1})/2 \\ \rho^{n+\frac{1}{2}} &= \rho(\phi^{n+\frac{1}{2}}) \\ \mu^{n+\frac{1}{2}} &= \mu(\phi^{n+\frac{1}{2}}) \end{aligned}$$

Surface tension is taken into account by adding $\vec{f}_\sigma^{\,n+\frac{1}{2}}$ to the equation for the preliminary velocity.

A scheme staggered in both space and time, similar to the one proposed for low Mach number flows by Pierce (2001) and Wall et al. (2002), is used by several authors for two-phase flows (Desjardins et al., 2008; van der Pijl et al., 2005). First, the interface is advanced from $t^{n-\frac{1}{2}}$ to $t^{n+\frac{1}{2}}$. The density $\rho^{n+\frac{1}{2}}$, viscosity $\mu^{n+\frac{1}{2}}$ and curvature $\kappa^{n+\frac{1}{2}}$ are then directly computed using the updated level-set function $\phi^{n+\frac{1}{2}}$ and no time interpolation is needed. Another major difference is that the pressure gradient and the surface tension term are not included in the equation for the provisional velocity (*non-incremental* projection scheme):

$$\frac{\vec{u}^* - \vec{u}^n}{\Delta t} + (\vec{u}^n \cdot \nabla \vec{u})^{n+\frac{1}{2}} = \frac{1}{\rho^{n+\frac{1}{2}}} \nabla \cdot \mu^{n+\frac{1}{2}} \left[(\nabla \vec{u}^*) + (\nabla \vec{u}^n)^T \right] \tag{4.11}$$

The projection step includes all "forgotten" terms and computes the new pressure field and the final incompressible flow field:

$$\begin{aligned} \frac{\vec{u}^{n+1} - \vec{u}^*}{\Delta t} &= -\frac{1}{\rho^{n+\frac{1}{2}}} \nabla p^{n+\frac{1}{2}} + \vec{f}_\sigma + \vec{g} \\ \nabla \cdot \vec{u}^{n+1} &= 0 \end{aligned} \tag{4.12}$$

The equation for the pressure is then:

$$\nabla \cdot \left(\frac{1}{\rho^{n+\frac{1}{2}}} \nabla p^{n+\frac{1}{2}} \right) = \nabla \cdot \left(\frac{\vec{u}^*}{\Delta t} + \vec{f}_\sigma + \vec{g} \right) \tag{4.13}$$

In several works (Desjardins et al., 2008; van der Pijl et al., 2005; Son and Dhir, 2007), the density jump in the pressure equation is implemented using the Ghost Fluid Method (GFM) described in §4.2.4. This allows a sharp representation of the density. In Desjardins et al. (2008), the surface tension term is also implemented directly as a sharp jump condition in the pressure equation using the Ghost Fluid Method, and, therefore, $\vec{f}_\sigma$ is omitted in Eqs. (4.12) and (4.13). Son and Dhir (2007) included $\vec{f}_\sigma$, but their method is strictly equivalent on the discrete level to the Ghost Fluid Method employed by Desjardins et al. (2008) due to a particular choice of

$\vec{f}_\sigma$ (see §4.2.4). On the other hand, van der Pijl et al. (2005) used the CSF approach and $\vec{f}_\sigma$ is computed from a smoothed delta-function.

Due to the strong coupling between the pressure gradient and the surface tension term, it seems important to always keep both terms together and at the same time level. For example, in an incremental projection method, both terms should be evaluated at the previous time step $t^{n-\frac{1}{2}}$ and included in the provisional velocity equation (4.11). But in the projection step, the old surface tension term should be replaced by the new one for $t^{n-\frac{1}{2}}$, in order to compute the new pressure. This approach is used in a collocated flow solver, detailed in §B.1, that is also tested.

4.2.4 Ghost Fluid Method

The Ghost Fluid Method (GFM) was first developed by Fedkiw et al. (1999) for the application to shock-waves and the solution of the Euler equations for (inviscid) compressible flows. Liu et al. (2000) applied this method to the variable coefficient Poisson equation with jump conditions at the interface. In Gibou et al. (2002), a similar Poisson equation is considered but with Dirichlet boundary conditions (instead of jump conditions), in which case the solution is not coupled across the interface. In Ng et al. (2009), an improved method for Dirichlet conditions is proposed that shows second-order behavior also for the gradients of the scalar field. Kang et al. (2000) applied the GFM to simulate flame surface propagation in an inviscid incompressible flow, taking into account the velocity jump at the interface.

The Ghost Fluid Method is basically an extrapolation technique which allows implementation of jump conditions at the interface in classical numerical methods, and which has gained most of its popularity in the field of two-phase flows. Probably for historical and practical reasons, the Ghost Fluid Method is often combined with the level-set method, although in principle any other interface method could be used. The ability to implement actual jump conditions avoids numerical smearing of fluid properties. In the Ghost Fluid Method, a robust scheme is obtained by applying standard spatial and temporal discretizations on grid points belonging only to one side of the interface. For stencils across the interface, additional ghost points are defined on the other side of the interface on which flow variables are extrapolated using the interface jump conditions. The technique consists of defining values at the ghost points across the interface as the sum of the local value and the jump condition at the interface.

Applied to the variable-coefficient Poisson equation (Liu et al., 2000), it is possible to take into account jump conditions in the coefficients, the normal gradient and the solution itself. Interestingly, the associated linear system is not modified and produces the standard symmetric matrix, for which efficient readily-available variable-coefficient solvers can be used. We will apply this method to the pressure equation, in which both the density and the pressure contains a jump at the interface. In the following, jump conditions across the interface Γ are noted as $[p]_\Gamma = p_l - p_g$. The model problem is then given by:

$$\nabla \cdot \left(\frac{1}{\rho} \nabla p \right) = \frac{1}{\Delta t} \left(\nabla \cdot \vec{u}^* \right) \equiv f \quad \text{for} \quad \vec{x} \in \Omega \tag{4.14}$$

$$[p]_\Gamma = \sigma \kappa \equiv a \quad \text{for} \quad \vec{x} \in \Gamma \tag{4.15}$$

Furthermore, we have a density jump $[\rho]_\Gamma = \rho_l - \rho_g$ at the interface. In a standard second-order finite-difference setting, the variable-coefficient Poisson equation without jump conditions would

be discretized in a two-dimensional flow as:

$$\frac{1}{\Delta x}\left[\frac{1}{\rho_{i+\frac{1}{2},j}}\frac{p_{i+1,j}-p_{i,j}}{\Delta x}-\frac{1}{\rho_{i-\frac{1}{2},j}}\frac{p_{i,j}-p_{i-1,j}}{\Delta x}\right] + \frac{1}{\Delta y}\left[\frac{1}{\rho_{i,j+\frac{1}{2}}}\frac{p_{i,j+1}-p_{i,j}}{\Delta y}-\frac{1}{\rho_{i,j-\frac{1}{2}}}\frac{p_{i,j}-p_{i,j-1}}{\Delta y}\right] = f_{i,j} \tag{4.16}$$

If all grid points belonging to the 5-point stencil are on the same side of the interface, this equation needs no modification. If, however, there is a change of sign in the level-set function on two or more grid points, an additional term should be included in the RHS of this equation and the cell-face density should be modified. We have several possibilities:

- When $\phi_{i,j}\phi_{i-1,j} < 0$, then define

$$\begin{aligned}\theta &= \frac{|\phi_{i-1,j}|}{|\phi_{i,j}|+|\phi_{i-1,j}|}\\ \widehat{\rho}_{i-\frac{1}{2},j} &= \rho_{i-1,j}\theta+\rho_{i,j}(1-\theta)\\ a_\Gamma &= a_{i,j}\theta+a_{i-1,j}(1-\theta)\end{aligned} \tag{4.17}$$

  where the densities $\rho_{i-1,j}$ and $\rho_{i,j}$ are determined in a sharp manner (either ρ_l or ρ_g depending on the value of ϕ). Note the difference in the averaging for $\widehat{\rho}_{i-\frac{1}{2},j}$ and a_Γ. The value $\widehat{\rho}_{i-\frac{1}{2},j}$ denotes the effective density along the path between $\vec{x}_{i-1,j}$ and $\vec{x}_{i,j}$ which replaces $\rho_{i-\frac{1}{2},j}$ in Eq. (4.16), whereas a_Γ denotes the interpolated value of a on the interface. In order to take into account the pressure jump, the term

$$\frac{S\left(\phi_{i-1,j}\right)a_\Gamma}{\widehat{\rho}_{i-\frac{1}{2},j}(\Delta x)^2}$$

  is added to the RHS of Eq. (4.16), where $S(\phi_{i-1,j})$ denotes the sign of $\phi_{i-1,j}$.

- When $\phi_{i,j}\phi_{i+1,j} < 0$, then define

$$\begin{aligned}\theta &= \frac{|\phi_{i+1,j}|}{|\phi_{i,j}|+|\phi_{i+1,j}|}\\ \widehat{\rho}_{i+\frac{1}{2},j} &= \rho_{i+1,j}\theta+\rho_{i,j}(1-\theta)\\ a_\Gamma &= a_{i,j}\theta+a_{i+1,j}(1-\theta)\end{aligned} \tag{4.18}$$

  Next, the term

$$\frac{S\left(\phi_{i+1,j}\right)a_\Gamma}{\widehat{\rho}_{i+\frac{1}{2},j}(\Delta x)^2} \tag{4.19}$$

  is added to the RHS of Eq. (4.16).

A similar treatment is used for the situations where $\phi_{i,j}\phi_{i,j-1} < 0$ or $\phi_{i,j}\phi_{i,j+1} < 0$. The resulting coefficient matrix remains symmetric. If for example $\phi_{i,j}\phi_{i-1,j} < 0$, then $\widehat{\rho}_{i-\frac{1}{2},j}$ is modified according to Eq. (4.17) in the row containing the discrete Poisson equation for the grid point (i,j). The corresponding entry on the other side of the diagonal, located on the row for the grid point $(i-1,j)$, contains exactly the same coefficient defined by Eq. (4.18). The advantage of a symmetric sparse matrix for the Poisson equation with respect to a non-symmetric matrix is the ability to use more efficient solvers.

Son and Dhir (2007) have shown that the extra term in the RHS of the pressure equation can also be taken into account implicitly by including an additional term for the surface tension in the provisional velocity equation, in a manner similar to the CSF method. Suppose for example that $\vec{u}^*$ is the provisional velocity that does not include the surface tension term and that $\phi_{i,j}\phi_{i+1,j} < 0$ so that the interface passes between the corresponding grid points. We may now define

$$\frac{\vec{u}^{**} - \vec{u}^*}{\Delta t} = \frac{1}{\widehat{\rho}}\sigma\kappa\nabla H(\phi) \tag{4.20}$$

For the velocity at the grid point $x_{i+\frac{1}{2},j}$ this gives

$$\begin{aligned}\frac{\vec{u}^{**}_{i+\frac{1}{2},j} - \vec{u}^*_{i+\frac{1}{2},j}}{\Delta t} &= \frac{(\sigma\kappa)_{i+\frac{1}{2},j}}{\widehat{\rho}_{i+\frac{1}{2},j}}\frac{H(\phi_{i+1,j}) - H(\phi_{i,j})}{\Delta x} \\ &= \frac{S(\phi_{i+1,j})(\sigma\kappa)_{i+\frac{1}{2},j}}{\widehat{\rho}_{i+\frac{1}{2},j}\Delta x}\end{aligned}$$

In the pressure projection step, we replace $\vec{u}^*$ by $\vec{u}^{**}$ in the RHS of Eq. (4.14) which gives for the grid point $x_{i,j}$:

$$\begin{aligned}\frac{1}{\Delta t}(\nabla\cdot\vec{u}^{**})_{i,j} &= \frac{u^{**}_{i+\frac{1}{2},j} - u^{**}_{i-\frac{1}{2},j}}{\Delta x} + \frac{v^{**}_{i,j+\frac{1}{2}} - v^{**}_{i,j-\frac{1}{2}}}{\Delta y} \\ &= \frac{u^{*}_{i+\frac{1}{2},j} - u^{*}_{i-\frac{1}{2},j}}{\Delta x} + \frac{v^{*}_{i,j+\frac{1}{2}} - v^{*}_{i,j-\frac{1}{2}}}{\Delta y} + \frac{S(\phi_{i+1,j})(\sigma\kappa)_{i+\frac{1}{2},j}}{\widehat{\rho}_{i+\frac{1}{2},j}(\Delta x)^2}\end{aligned}$$

The additional term in the RHS is exactly the term in Eq. (4.19), supposing $a_\Gamma = (\sigma\kappa)_{i+\frac{1}{2},j}$. Therefore, this term needs to be evaluated using the interpolation procedure stipulated in Eqs. (4.17) and (4.18), as well as the density $\widehat{\rho}_{i+\frac{1}{2},j}$ defined according to Eq. (4.20). The additional terms for other grid points (such as $p_{i,j}$) as well as other interface crossings on the same stencil are implicitly taken into account in a similar manner. In conclusion, the Ghost Fluid Method can be rewritten in terms of a continuous surface force $\vec{f_\sigma} = (\sigma\kappa)_\Gamma\nabla H(\phi)/\widehat{\rho}$, which can be integrated in any of the aforementioned projection-type methods provided we make a proper choice of $\widehat{\rho}$ and $(\sigma\kappa)_\Gamma$.

4.3 Implementation of the Navier-Stokes solver

There are many ways to implement a numerical method for the incompressible Navier-Stokes equations with liquid-gas interfaces. At different levels of the construction of a discrete model, several choices have to be made, each of which will have an effect, more or less known, on the accuracy and robustness of the simulations. In this work, we have focused on a limited number of numerical schemes. The selection of these schemes was guided on the one hand by the choices made by other studies in the literature and on the other hand from our numerical experience. These numerical schemes are described more in detail in this section.

4.3.1 Time integration

For the decoupling of pressure and velocity, we choose the non-incremental projection method which consists of calculating first a provisional velocity without the pressure gradient, followed by a correction step in which the velocity is projected onto a divergence-free vector space. A predictor-corrector time scheme has been used for the provisional velocity step. The predictor

step is given by:

$$\frac{\vec{u}^{*,0}-\vec{u}^{n}}{\Delta t}+[\vec{u}\cdot\nabla\vec{u}]^{n}=\frac{1}{\rho^{n+\frac{1}{2}}}\left[\nabla\cdot\mu^{n+\frac{1}{2}}\left[(\nabla\vec{u}^{*,0})+(\nabla\vec{u}^{n})^{T}\right]\right]$$

where the superscript 0 refers to the prediction step of the time scheme. Note that the first part in the viscosity term is treated implicitly. The resulting system of linear equations is solved using an iterative ADI approach (Knikker, 2009). At each iteration, the implicit term is solved separately in each spatial direction using a second-order accurate discretization. This leads to tridiagonal systems, which are solved efficiently using the Thomas algorithm.

The projection step is constructed from the velocity correction:

$$\begin{aligned}\frac{\vec{u}^{n+1,0}-\vec{u}^{*,0}}{\Delta t}&=-\frac{1}{\rho^{n+\frac{1}{2}}}\nabla_{\delta}p^{n+\frac{1}{2},0}+\vec{g}+\vec{f}_{\sigma}^{n+\frac{1}{2}}\\ \nabla_{\delta}\cdot\vec{u}^{n+1,0}&=0\end{aligned}$$

where $\nabla_{\delta}=(\delta/\delta x,\delta/\delta y)$ and $\nabla_{\delta}\cdot\vec{u}=\delta u/\delta x+\delta v/\delta y$. The notation $\delta/\delta x$ and $\delta/\delta y$ is used here for the standard second-order centered difference schemes. For example $(\delta\phi/\delta x)_{i+\frac{1}{2},j}=(\phi_{i+1,j}-\phi_{i,j})/\Delta x$. The projection step requires the solution of the following Poisson equation for the pressure

$$\nabla_{\delta}\cdot\left(\frac{1}{\rho^{n+\frac{1}{2}}}\nabla_{\delta}p^{n+\frac{1}{2},0}\right)=\nabla_{\delta}\cdot\left(\frac{\vec{u}^{*,0}}{\Delta t}+\vec{g}+\vec{f}_{\sigma}^{n+\frac{1}{2}}\right)$$

In order to solve this equation, the preconditioned conjugate gradient method is used in combination with the modified incomplete Cholesky decomposition (Golub and Loan, 1989) using the NSPCG package (Oppe et al., 1988).

The corrector step of the overall time scheme consists of repeating the above steps, except that the explicit terms are replaced by a second-order approximation. The provisional velocity step is now:

$$\begin{aligned}\frac{\vec{u}^{*,1}-\vec{u}^{n}}{\Delta t}+\frac{1}{2}\left([\vec{u}\cdot\nabla\vec{u}]^{n+1,0}+[\vec{u}\cdot\nabla\vec{u}]^{n}\right)=\\ +\frac{1}{\rho^{n+\frac{1}{2}}}\left[\nabla\cdot\mu^{n+\frac{1}{2}}\left[(\nabla\vec{u}^{*,1})+\tfrac{1}{2}(\nabla\vec{u}^{n+1,0})^{T}+\tfrac{1}{2}(\nabla\vec{u}^{n})^{T}\right]\right]\end{aligned}$$

The final projection step is:

$$\nabla_{\delta}\cdot\left(\frac{1}{\rho^{n+\frac{1}{2}}}\nabla p^{n+\frac{1}{2}}\right)=\nabla_{\delta}\cdot\left(\frac{\vec{u}^{*,1}}{\Delta t}+\vec{g}+\vec{f}_{\sigma}^{n+\frac{1}{2}}\right)$$

followed by

$$\vec{u}^{n+\frac{1}{2}}=\vec{u}^{**,1}+\Delta t\left[\vec{g}+\vec{f}_{\sigma}^{n+\frac{1}{2}}-\frac{1}{\rho^{n+\frac{1}{2}}}\nabla p^{n+\frac{1}{2}}\right]$$

The interface is defined at intermediate time levels, $t^{n+\frac{1}{2}}$, such that fluid properties and the surface tension term can be expressed directly at this instant. For example, when the level-set method is used, ϕ is first advanced from $\phi^{n-\frac{1}{2}}$ to $\phi^{n+\frac{1}{2}}$ using the velocity field $\vec{u}^{n}$, for example

$$\frac{\phi^{n+\frac{3}{2}}-\phi^{n+\frac{1}{2}}}{\Delta t}+\vec{u}^{n+1}\cdot\nabla\phi^{n+\frac{1}{2}}=0$$

A similar equation can be derived for the volume-of-fluid methods. The interface advection step is done only once each time step. The density, viscosity and surface tension term at $t^{n+\frac{1}{2}}$ is therefore the same for both the predictor and corrector step.

4.3.2 Time step restriction

Several references evoke time step calculation (Kang et al., 2000; Sussman et al., 1998; van der Pijl et al., 2005; Francois et al., 2006; Sussman et al., 2007; Denner and van Wachem, 2015). In the current work, we first apply the standard Courant-Friedrichs-Lewy (CFL) condition and viscous time step restriction:

$$\Delta t_u \leq \frac{\mathrm{CFL}}{\frac{|u|_{\max}}{\Delta x} + \frac{|v|_{\max}}{\Delta y} + \frac{|w|_{\max}}{\Delta z}}$$

where $\mathrm{CFL} = 0.5$ (except otherwise specified) and

$$\Delta t_\nu \leq \mathrm{Fo}\frac{\min(\Delta x, \Delta y, \Delta z)^2}{\max(\mu_g/\rho_g, \mu_l/\rho_l)}$$

with $\mathrm{Fo} = 2.0$.

According to Brackbill et al. (1992), the explicit treatment of the surface tension term is stable when the time step is sufficiently small to resolve capillary wave phenomena, which leads to the condition:

$$\Delta t_\sigma \leq \mathrm{CFL}_\sigma \sqrt{\frac{(\rho_l + \rho_g)\min(\Delta x, \Delta y, \Delta z)^3}{4\pi\sigma}}$$

When gravity forces are present, another time step restriction is added

$$\Delta t_g \leq \sqrt{\frac{\Delta z}{|g|}}$$

Finally, the time step is set to

$$\Delta t = \min(\Delta t_u, \Delta t_\nu, \Delta t_\sigma, \Delta t_g)$$

4.3.3 Density at the cell faces and centers

When the volume-of-fluid method is used to capture the interface, the density is defined by the volume-fraction weighted average:

$$\begin{cases} \rho_{i,j} & = \rho_g + (\rho_l - \rho_g)C_{i,j} \\ \rho_{i+\frac{1}{2},j} & = \rho_g + (\rho_l - \rho_g)C_{i+\frac{1}{2},j} \\ \rho_{i,j+\frac{1}{2}} & = \rho_g + (\rho_l - \rho_g)C_{i,j+\frac{1}{2}} \end{cases} \tag{4.21}$$

where $C_{i+\frac{1}{2},j} = (C_{i,j} + C_{i+1,j})/2$ and $C_{i,j+\frac{1}{2}} = (C_{i,j} + C_{i,j+1})/2$. When a smoothing kernel is used (see §4.2.1.2), the variable C is replaced by the smoothed fraction $\widetilde{C}$ in the above equations.

When the interface is described by the level-set method, we use one of the following methods to determine the density at the cell faces and centers:

- In order to obtain a sharp interface approach, the density is determined using the Ghost Fluid Method described in §4.2.4, in particular Eqs. (4.17) and (4.18). The density at the

cell center is provided by the Heaviside function as:

$$\rho_{i,j} = \rho_g + (\rho_l - \rho_g) H(\phi_{i,j})$$

- In the case of a smooth interface approach, the density is given by

$$\begin{cases} \rho_{i,j} & = \rho_g + (\rho_l - \rho_g) H_\epsilon(\phi_{i,j}) \\ \rho_{i+\frac{1}{2},j} & = \rho_g + (\rho_l - \rho_g) H_\epsilon(\phi_{i+\frac{1}{2},j}) \\ \rho_{i,j+\frac{1}{2}} & = \rho_g + (\rho_l - \rho_g) H_\epsilon(\phi_{i,j+\frac{1}{2}}) \end{cases} \tag{4.22}$$

where H_ϵ is the smoothed Heaviside function defined by Eq. (4.5) and $\phi_{i+\frac{1}{2},j}$ and $\phi_{i,j+\frac{1}{2}}$ are defined from the cell centered values of $\phi_{i,j}$ using averaging.

- Vigneaux (2008) proposed a set of averaging operations to compute fluid properties at different grid locations. In order to determine the density at the cell faces, the value at the cell center is first determined from

$$\rho_{i,j} = \rho_g + (\rho_l - \rho_g) H(\phi_{i,j})$$

The density at the cell vertices are then computed from:

$$\begin{cases} \overline{\rho}^u_{i+\frac{1}{2},j+\frac{1}{2}} = \dfrac{\rho_{i,j} \cdot \rho_{i,j+1}}{\rho_{i,j} + \rho_{i,j+1}} + \dfrac{\rho_{i+1,j} \cdot \rho_{i+1,j+1}}{\rho_{i+1,j} + \rho_{i,j+1}} \\ \overline{\rho}^v_{i+\frac{1}{2},j+\frac{1}{2}} = \dfrac{\rho_{i,j} \cdot \rho_{i+1,j}}{\rho_{i,j} + \rho_{i+1,j}} + \dfrac{\rho_{i,j+1} \cdot \rho_{i+1,j+1}}{\rho_{i,j+1} + \rho_{i,j+1}} \end{cases}$$

The density at the cell faces $\rho_{i+\frac{1}{2},j}$ and $\rho_{i,j+\frac{1}{2}}$ then follow from averaging of values at cell centers and cell vertices:

$$\begin{cases} \rho_{i+\frac{1}{2},j} = \dfrac{1}{4}\left[\overline{\rho}^u_{i+\frac{1}{2},j+\frac{1}{2}} + \overline{\rho}^u_{i+\frac{1}{2},j-\frac{1}{2}} + \rho_{i,j} + \rho_{i+1,j}\right] \\ \rho_{i,j+\frac{1}{2}} = \dfrac{1}{4}\left[\overline{\rho}^v_{i+\frac{1}{2},j+\frac{1}{2}} + \overline{\rho}^v_{i-\frac{1}{2},j+\frac{1}{2}} + \rho_{i,j} + \rho_{i,j+1}\right] \end{cases} \tag{4.23}$$

The subsequent averaging steps can be regarded as a manner of smoothing the density.

4.3.4 Advection terms

The advection terms in the momentum equations are discretized using the second-order ENO scheme, which seems to be a popular choice among other works on two-phase flows (Chang et al., 1996; Son and Dhir, 2007). We use:

$$[\vec{u} \cdot \nabla u]^u_{i+\frac{1}{2},j} = u_{i+\frac{1}{2},j} \left[\frac{\partial u}{\partial x}\right]^{\text{ENO}}_{i+\frac{1}{2},j} + \overline{v}_{i+\frac{1}{2},j} \left[\frac{\partial u}{\partial y}\right]^{\text{ENO}}_{i+\frac{1}{2},j} \tag{4.24}$$

$$[\vec{u} \cdot \nabla v]^u_{i,j+\frac{1}{2}} = \overline{u}_{i,j+\frac{1}{2}} \left[\frac{\partial v}{\partial x}\right]^{\text{ENO}}_{i,j+\frac{1}{2}} + v_{i,j+\frac{1}{2}} \left[\frac{\partial v}{\partial y}\right]^{\text{ENO}}_{i,j+\frac{1}{2}} \tag{4.25}$$

where

$$\overline{v}_{i+\frac{1}{2},j} = \frac{1}{4}\left(v_{i,j+\frac{1}{2}} + v_{i,j-\frac{1}{2}} + v_{i+1,j+\frac{1}{2}} + v_{i+1,j-\frac{1}{2}}\right)$$
$$\overline{u}_{i,j+\frac{1}{2}} = \frac{1}{4}\left(u_{i+\frac{1}{2},j} + u_{i-\frac{1}{2},j} + u_{i+\frac{1}{2},j+1} + u_{i-\frac{1}{2},j+1}\right)$$

The advection terms are upwind approximations determined from:

$$u\left[\frac{\partial\phi}{\partial x}\right]^{\text{ENO}} = \begin{cases} u\phi_x^- & \text{if } u \geq 0 \\ u\phi_x^+ & \text{if } u < 0 \end{cases}$$

The spatial derivatives ϕ_x^- and ϕ_x^+ are calculated using the second-order ENO scheme. Defining,

$$\begin{aligned} D\phi_i^- &= \phi_i - \phi_{i-1} \\ D\phi_i^+ &= \phi_{i+1} - \phi_i \\ D^2\phi_i^- &= \phi_i - 2\phi_{i-1} + \phi_{i-2} \\ D^2\phi_i^0 &= \phi_{i+1} - 2\phi_i + \phi_{i-1} \\ D^2\phi_i^+ &= \phi_{i+2} - 2\phi_{i+1} + \phi_i \end{aligned}$$

the ENO scheme is formulated as

$$\begin{aligned} \phi_x^- &= \frac{1}{h}D\phi^- + \frac{1}{2h}\text{minmod}\left(D^2\phi^0, D^2\phi^-\right) \\ \phi_x^+ &= \frac{1}{h}D\phi^+ - \frac{1}{2h}\text{minmod}\left(D^2\phi^0, D^2\phi^+\right) \end{aligned}$$

where

$$\text{minmod}(a,b) = \begin{cases} a & \text{if } |a| < |b| \text{ and } ab > 0 \\ b & \text{if } |b| < |a| \text{ and } ab > 0 \\ 0 & \text{otherwise} \end{cases}$$

4.3.5 Viscous terms

In a staggered grid arrangement, discretization of the viscous terms is straight-forward. Despite the different locations of the velocity components, additional interpolations are not needed. The implicit viscous terms are discretized as:

$$\begin{aligned} \frac{1}{\rho_{i+\frac{1}{2},j}}\left[\nabla\cdot\mu(\nabla u)\right]_{i+\frac{1}{2},j} &= \frac{1}{\rho_{i+\frac{1}{2},j}\Delta x}\left[\mu_{i+1,j}\frac{u_{i+\frac{3}{2},j} - u_{i+\frac{1}{2},j}}{\Delta x} - \mu_{i,j}\frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x}\right] \\ &+ \frac{1}{\rho_{i+\frac{1}{2},j}\Delta y}\left[\mu^u_{i+\frac{1}{2},j+\frac{1}{2}}\frac{u_{i+\frac{1}{2},j+1} - u_{i+\frac{1}{2},j}}{\Delta y} - \mu^u_{i+\frac{1}{2},j-\frac{1}{2}}\frac{u_{i+\frac{1}{2},j} - u_{i+\frac{1}{2},j-1}}{\Delta y}\right] \\ \frac{1}{\rho_{i,j+\frac{1}{2}}}\left[\nabla\cdot\mu(\nabla v)\right]_{i,j+\frac{1}{2}} &= \frac{1}{\rho_{i,j+\frac{1}{2}}\Delta x}\left[\mu^v_{i+\frac{1}{2},j+\frac{1}{2}}\frac{v_{i+1,j+\frac{1}{2}} - v_{i,j+\frac{1}{2}}}{\Delta x} - \mu^v_{i-\frac{1}{2},j+\frac{1}{2}}\frac{v_{i,j+\frac{1}{2}} - v_{i-1,j+\frac{1}{2}}}{\Delta x}\right] \\ &+ \frac{1}{\rho_{i,j+\frac{1}{2}}\Delta y}\left[\mu_{i,j+1}\frac{v_{i,j+\frac{3}{2}} - v_{i,j+\frac{1}{2}}}{\Delta y} - \mu_{i,j}\frac{v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{\Delta y}\right] \end{aligned}$$

The explicit viscous terms in the equations for u and v are calculated from respectively:

$$\begin{aligned}
\frac{1}{\rho_{i+\frac{1}{2},j}}\left[\nabla\cdot\mu(\nabla\vec{u})^T\right]^u_{i+\frac{1}{2},j} &= \frac{1}{\rho_{i+\frac{1}{2},j}\Delta x}\left[\mu_{i+1,j}\frac{u_{i+\frac{3}{2},j}-u_{i+\frac{1}{2},j}}{\Delta x}-\mu_{i,j}\frac{u_{i+\frac{1}{2},j}-u_{i-\frac{1}{2},j}}{\Delta x}\right] \\
&+\frac{1}{\rho_{i+\frac{1}{2},j}\Delta y}\left[\mu^u_{i+\frac{1}{2},j+\frac{1}{2}}\frac{v_{i+1,j+\frac{1}{2}}-v_{i,j+\frac{1}{2}}}{\Delta x}-\mu^u_{i+\frac{1}{2},j-\frac{1}{2}}\frac{v_{i+1,j-\frac{1}{2}}-v_{i,j-\frac{1}{2}}}{\Delta x}\right] \\
\frac{1}{\rho_{i,j+\frac{1}{2}}}\left[\nabla\cdot\mu(\nabla\vec{u})^T\right]^v_{i,j+\frac{1}{2}} &= \frac{1}{\rho_{i,j+\frac{1}{2}}\Delta x}\left[\mu^v_{i+\frac{1}{2},j+\frac{1}{2}}\frac{u_{i+\frac{1}{2},j+1}-u_{i+\frac{1}{2},j}}{\Delta y}-\mu^v_{i-\frac{1}{2},j+\frac{1}{2}}\frac{u_{i-\frac{1}{2},j+1}-u_{i-\frac{1}{2},j}}{\Delta y}\right] \\
&+\frac{1}{\rho_{i,j+\frac{1}{2}}\Delta y}\left[\mu_{i,j+1}\frac{v_{i,j+\frac{3}{2}}-v_{i,j+\frac{1}{2}}}{\Delta y}-\mu_{i,j}\frac{v_{i,j+\frac{1}{2}}-v_{i,j-\frac{1}{2}}}{\Delta y}\right]
\end{aligned}$$

Different methods are used in this work to define the viscosity at the cell center and vertices:

- In the work of Vigneaux (2008), the viscosity is defined as:

$$\begin{cases}
\mu_{i,j} &= \mu_g+(\mu_l-\mu_g)H(\phi_{i,j}) \\
\mu^u_{i+\frac{1}{2},j+\frac{1}{2}} &= \dfrac{\mu_{i,j}\cdot\mu_{i,j+1}}{\mu_{i,j}+\mu_{i,j+1}}+\dfrac{\mu_{i+1,j}\cdot\mu_{i+1,j+1}}{\mu_{i+1,j}+\mu_{i,j+1}} \\
\mu^v_{i+\frac{1}{2},j+\frac{1}{2}} &= \dfrac{\mu_{i,j}\cdot\mu_{i+1,j}}{\mu_{i,j}+\mu_{i+1,j}}+\dfrac{\mu_{i,j+1}\cdot\mu_{i+1,j+1}}{\mu_{i,j+1}+\mu_{i,j+1}}
\end{cases} \tag{4.26}$$

- In a level-set based method, the viscosity can also be evaluated using the smoothed Heaviside function (4.5):

$$\begin{cases}
\mu_{i,j} &= \mu_g+(\mu_l-\mu_g)H_\epsilon(\phi_{i,j}) \\
\mu_{i+\frac{1}{2},j+\frac{1}{2}} &= \mu_g+(\mu_l-\mu_g)H_\epsilon(\phi_{i+\frac{1}{2},j+\frac{1}{2}})
\end{cases} \tag{4.27}$$

  where

$$\phi_{i+\frac{1}{2},j+\frac{1}{2}}=(\phi_{i,j}+\phi_{i+1,j}+\phi_{i,j+1}+\phi_{i+1,j+1})/4$$

  Here, the superscripts $()^u$ and $()^v$ are dropped, since both terms are identical.

- In a volume-of-fluid based method, the viscosity is evaluated using:

$$\begin{cases}
\mu_{i,j} &= \mu_g+(\mu_l-\mu_g)C_{i,j} \\
\mu_{i+\frac{1}{2},j+\frac{1}{2}} &= \mu_g+(\mu_l-\mu_g)C_{i+\frac{1}{2},j+\frac{1}{2}}
\end{cases} \tag{4.28}$$

  where

$$C_{i+\frac{1}{2},j+\frac{1}{2}}=(C_{i,j}+C_{i+1,j}+C_{i,j+1}+C_{i+1,j+1})/4$$

  where C is replaced by $\widetilde{C}$ when a smoothing kernel is applied (see §4.2.1.2).

  Note that when the viscosity is smeared out using one of the methods above, the first derivative is continuous at the interface, and hence the second-derivatives can be determined using standard central differencing to compute the viscous terms.

- When the viscosity is represented in a sharp fashion, jump conditions should be taken into account when evaluating the viscous terms (Kang et al., 2000). This approach is rather

complicated, however, and is not used in this work. Instead, we use the method proposed by Son and Dhir (2007), which is a simplified version of the Ghost Fluid Method by Kang et al. (2000). In this approach, the viscous terms are computed from slightly different expressions:

$$\begin{aligned}
\frac{1}{\rho_{i+\frac{1}{2},j}}\left[\nabla\cdot\mu(\nabla u)\right]_{i+\frac{1}{2},j} &= \frac{1}{\rho_{i+\frac{1}{2},j}\Delta x}\left[\mu^x_{i+1,j}\frac{u_{i+\frac{3}{2},j}-u_{i+\frac{1}{2},j}}{\Delta x}-\mu^x_{i,j}\frac{u_{i+\frac{1}{2},j}-u_{i-\frac{1}{2},j}}{\Delta x}\right]\\
&+\frac{1}{\rho_{i+\frac{1}{2},j}\Delta y}\left[\mu^y_{i+\frac{1}{2},j+\frac{1}{2}}\frac{u_{i+\frac{1}{2},j+1}-u_{i+\frac{1}{2},j}}{\Delta y}-\mu^y_{i+\frac{1}{2},j-\frac{1}{2}}\frac{u_{i+\frac{1}{2},j}-u_{i+\frac{1}{2},j-1}}{\Delta y}\right]\\
\frac{1}{\rho_{i,j+\frac{1}{2}}}\left[\nabla\cdot\mu(\nabla v)\right]_{i,j+\frac{1}{2}} &= \frac{1}{\rho_{i,j+\frac{1}{2}}\Delta x}\left[\mu^x_{i+\frac{1}{2},j+\frac{1}{2}}\frac{v_{i+1,j+\frac{1}{2}}-v_{i,j+\frac{1}{2}}}{\Delta x}-\mu^x_{i-\frac{1}{2},j+\frac{1}{2}}\frac{v_{i,j+\frac{1}{2}}-v_{i-1,j+\frac{1}{2}}}{\Delta x}\right]\\
&+\frac{1}{\rho_{i,j+\frac{1}{2}}\Delta y}\left[\mu^y_{i,j+1}\frac{v_{i,j+\frac{3}{2}}-v_{i,j+\frac{1}{2}}}{\Delta y}-\mu^y_{i,j}\frac{v_{i,j+\frac{1}{2}}-v_{i,j-\frac{1}{2}}}{\Delta y}\right]\\
\frac{1}{\rho_{i+\frac{1}{2},j}}\left[\nabla\cdot\mu(\nabla\vec{u})^T\right]^u_{i+\frac{1}{2},j} &= \frac{1}{\rho_{i+\frac{1}{2},j}\Delta x}\left[\mu^x_{i+1,j}\frac{u_{i+\frac{3}{2},j}-u_{i+\frac{1}{2},j}}{\Delta x}-\mu^x_{i,j}\frac{u_{i+\frac{1}{2},j}-u_{i-\frac{1}{2},j}}{\Delta x}\right]\\
&+\frac{1}{\rho_{i+\frac{1}{2},j}\Delta y}\left[\mu^x_{i+\frac{1}{2},j+\frac{1}{2}}\frac{v_{i+1,j+\frac{1}{2}}-v_{i,j+\frac{1}{2}}}{\Delta x}-\mu^x_{i+\frac{1}{2},j-\frac{1}{2}}\frac{v_{i+1,j-\frac{1}{2}}-v_{i,j-\frac{1}{2}}}{\Delta x}\right]\\
\frac{1}{\rho_{i,j+\frac{1}{2}}}\left[\nabla\cdot\mu(\nabla\vec{u})^T\right]^v_{i,j+\frac{1}{2}} &= \frac{1}{\rho_{i,j+\frac{1}{2}}\Delta x}\left[\mu^y_{i+\frac{1}{2},j+\frac{1}{2}}\frac{u_{i+\frac{1}{2},j+1}-u_{i+\frac{1}{2},j}}{\Delta y}-\mu^y_{i-\frac{1}{2},j+\frac{1}{2}}\frac{u_{i-\frac{1}{2},j+1}-u_{i-\frac{1}{2},j}}{\Delta y}\right]\\
&+\frac{1}{\rho_{i,j+\frac{1}{2}}\Delta y}\left[\mu^y_{i,j+1}\frac{v_{i,j+\frac{3}{2}}-v_{i,j+\frac{1}{2}}}{\Delta y}-\mu^y_{i,j}\frac{v_{i,j+\frac{1}{2}}-v_{i,j-\frac{1}{2}}}{\Delta y}\right]
\end{aligned}$$

The viscosity is evaluated using an interpolation method similar to the Ghost Fluid Method (Eqs. (4.17) and (4.18)):

$$\begin{cases}
(\mu^x_{i+\frac{1}{2},j+\frac{1}{2}})^{-1} &= \dfrac{\mu^{-1}_{i+1,j+\frac{1}{2}}|\phi_{i+1,j+\frac{1}{2}}|+\mu^{-1}_{i,j+\frac{1}{2}}|\phi_{i,j+\frac{1}{2}}|}{|\phi_{i+1,j+\frac{1}{2}}|+|\phi_{i,j+\frac{1}{2}}|}\\
(\mu^y_{i+\frac{1}{2},j+\frac{1}{2}})^{-1} &= \dfrac{\mu^{-1}_{i+\frac{1}{2},j+1}|\phi_{i+\frac{1}{2},j+1}|+\mu^{-1}_{i+\frac{1}{2},j}|\phi_{i+\frac{1}{2},j}|}{|\phi_{i+\frac{1}{2},j+1}|+|\phi_{i+\frac{1}{2},j}|}\\
(\mu^x_{i,j})^{-1} &= \dfrac{\mu^{-1}_{i+\frac{1}{2},j}|\phi_{i+\frac{1}{2},j}|+\mu^{-1}_{i-\frac{1}{2},j}|\phi_{i-\frac{1}{2},j}|}{|\phi_{i+\frac{1}{2},j}|+|\phi_{i-\frac{1}{2},j}|}\\
(\mu^y_{i,j})^{-1} &= \dfrac{\mu^{-1}_{i,j+\frac{1}{2}}|\phi_{i,j+\frac{1}{2}}|+\mu^{-1}_{i,j-\frac{1}{2}}|\phi_{i,j-\frac{1}{2}}|}{|\phi_{i,j+\frac{1}{2}}|+|\phi_{i,j-\frac{1}{2}}|}
\end{cases} \tag{4.29}$$

with

$$\begin{aligned}
\mu_{i+\frac{1}{2},j} &= \mu_g+(\mu_l-\mu_g)H(\phi_{i+\frac{1}{2},j})\\
\mu_{i,j+\frac{1}{2}} &= \mu_g+(\mu_l-\mu_g)H(\phi_{i,j+\frac{1}{2}})
\end{aligned}$$

and

$$\begin{aligned}
\phi_{i+\frac{1}{2},j} &= (\phi_{i+1,j}+\phi_{i,j})/2\\
\phi_{i,j+\frac{1}{2}} &= (\phi_{i,j+1}+\phi_{i,j})/2
\end{aligned}$$

4.3.6 Surface tension

Following the review in §4.2.2 and the generalization of the Ghost Fluid Method in §4.2.4, the surface tension term can be expressed as

$$\vec{f_\sigma} = \frac{\sigma}{\rho}\kappa\nabla H(\phi) \tag{4.30}$$

We suppose that σ is constant. The density in this term is defined as the value at the cell face, $\rho = \rho_{i+\frac{1}{2},j}$ or $\rho = \rho_{i,j+\frac{1}{2}}$. Furthermore, the Heaviside function can be either the sharp interface representation (which is then equivalent to the Ghost Fluid Method, Eq. (4.20)), or the smoothed Heaviside function (4.5). A VOF-based CSF implementation is given by:

$$\vec{f_\sigma} = \frac{\sigma}{\rho}\kappa\nabla C \tag{4.31}$$

where the variable C is replaced by $\widetilde{C}$ when a smoothing kernel is used. What remains are the different choices for the computation of the curvature, κ. We have used the following algorithms:

- The curvature, $\kappa = -\nabla\cdot\vec{n}$, is computed directly at the cell face from normal vectors derived at the cell centers and vertices (Vigneaux, 2008). To achieve this, the level-set at the cell vertices are first computed from:

$$\overline{\phi}_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{4}\left(\phi_{i,j} + \phi_{i+1,j} + \phi_{i,j+1} + \phi_{i+1,j+1}\right)$$

  followed by

$$\begin{aligned}\overline{\phi}^u_{i+\frac{1}{2},j} &= \frac{1}{4}\left(\overline{\phi}_{i+\frac{1}{2},j+\frac{1}{2}} + \overline{\phi}_{i+\frac{1}{2},j-\frac{1}{2}} + \phi_{i,j} + \phi_{i+1,j}\right)\\ &= \frac{1}{16}\left(\phi_{i,j+1} + \phi_{i+1,j+1} + 5\phi_{i,j} + 5\phi_{i+1,j} + \phi_{i,j-1} + \phi_{i+1,j-1}\right)\end{aligned}$$

$$\begin{aligned}\overline{\phi}^v_{i,j+\frac{1}{2}} &= \frac{1}{4}\left(\overline{\phi}_{i+\frac{1}{2},j+\frac{1}{2}} + \overline{\phi}_{i-\frac{1}{2},j+\frac{1}{2}} + \phi_{i,j} + \phi_{i,j+1}\right)\\ &= \frac{1}{16}\left(\phi_{i+1,j} + \phi_{i+1,j+1} + 5\phi_{i,j} + 5\phi_{i,j+1} + \phi_{i-1,j} + \phi_{i-1,j+1}\right)\end{aligned}$$

  The interface normal is then computed at the cell center using

$$\vec{n} = \left(\frac{\delta\overline{\phi}^u}{\delta x}, \frac{\delta\overline{\phi}^v}{\delta y}\right) \Big/ \sqrt{\left(\frac{\delta\overline{\phi}^u}{\delta x}\right)^2 + \left(\frac{\delta\overline{\phi}^v}{\delta y}\right)^2} \tag{4.32}$$

  For the normal vector at the cell vertices, we use

$$\vec{\overline{n}} = \left(\frac{\delta\overline{\phi}^v}{\delta x}, \frac{\delta\overline{\phi}^u}{\delta y}\right) \Big/ \sqrt{\left(\frac{\delta\overline{\phi}^v}{\delta x}\right)^2 + \left(\frac{\delta\overline{\phi}^u}{\delta y}\right)^2}$$

  The curvature at the cell faces are then expressed as:

$$\begin{aligned}\kappa_{i+\frac{1}{2},j} &= -\left(\frac{\delta n_x}{\delta x} + \frac{\delta\overline{n}_y}{\delta y}\right)\\ \kappa_{i,j+\frac{1}{2}} &= -\left(\frac{\delta\overline{n}_x}{\delta x} + \frac{\delta n_y}{\delta y}\right)\end{aligned} \tag{4.33}$$

The surface terms in the momentum equation are then defined as:

$$(f_\sigma^u)_{i+\frac{1}{2},j} = \frac{\sigma}{\rho_{i+\frac{1}{2},j}} \kappa_{i+\frac{1}{2},j} \frac{\delta H}{\delta x}$$
$$(f_\sigma^v)_{i,j+\frac{1}{2}} = \frac{\sigma}{\rho_{i,j+\frac{1}{2}}} \kappa_{i,j+\frac{1}{2}} \frac{\delta H}{\delta y}$$

where $H(\phi)$ is defined at the cell center.

- The curvature is computed at the cell center from

$$\kappa = -\nabla \cdot \vec{n} \tag{4.34}$$

using standard central differencing. The values at the cell center are then extrapolated to the cell faces as defined for a_Γ in Eq. (4.17).

- The curvature is computed using the alternative expression proposed by Chang et al. (1996), which uses only first- and second-order spatial derivatives of the level-set function:

$$\kappa = -\frac{\phi_y^2 \phi_{xx} - 2\phi_x \phi_y \phi_{xy} + \phi_x^2 \phi_{yy}}{(\phi_x^2 + \phi_y^2)^{3/2}} \tag{4.35}$$

The curvature is evaluated at the cell center using standard central difference schemes. The extrapolation to the cell faces is done using the standard interpolation.

- The spatial derivatives for the interface normal and curvature are computed using the kernels in Eq. (4.10) defined by the least-squares approach proposed by Marchandise et al. (2007), see §4.2.2. The extrapolation to the cell faces is then performed according to the Ghost Fluid Method in equation Eq. (4.17).

- The curvature is computed as $\kappa = -\nabla \cdot \vec{n}$ at the cell center using central difference discretization with some extra smoothing (Sussman et al., 1999). The gradient of the level-set function is computed at the vertices using the formulas:

$$\left(\frac{\partial \phi}{\partial x}\right)_{i+\frac{1}{2},j+\frac{1}{2}} \simeq \frac{\phi_{i+1,j+1} + \phi_{i+1,j} - \phi_{i,j+1} - \phi_{i,j}}{2\Delta x}$$
$$\left(\frac{\partial \phi}{\partial y}\right)_{i+\frac{1}{2},j+\frac{1}{2}} \simeq \frac{\phi_{i+1,j+1} + \phi_{i,j+1} - \phi_{i+1,j} - \phi_{i,j}}{2\Delta y}$$

The unit normal vector is then obtained by normalization of the level-set gradient as in Eq. (4.32). The curvature at the cell center is then computed using the relations:

$$\left(\frac{\partial n^x}{\partial x}\right)_{i,j} \simeq \frac{n^x_{i+\frac{1}{2},j+\frac{1}{2}} + n^x_{i+\frac{1}{2},j-\frac{1}{2}} - n^x_{i-\frac{1}{2},j+\frac{1}{2}} - n^x_{i-\frac{1}{2},j-\frac{1}{2}}}{2\Delta x}$$
$$\left(\frac{\partial n^y}{\partial y}\right)_{i,j} \simeq \frac{n^y_{i+\frac{1}{2},j+\frac{1}{2}} + n^y_{i-\frac{1}{2},j+\frac{1}{2}} - n^y_{i+\frac{1}{2},j-\frac{1}{2}} - n^y_{i-\frac{1}{2},j-\frac{1}{2}}}{2\Delta y} \tag{4.36}$$

The curvature is then interpolated to the cell faces similarly as for a_Γ in Eq. (4.17).

- The curvature is computed from the volume-of-fluid function using the Height Function method. In each cell containing a line interface (determined previously by the VOF-PLIC advection method), the largest component normal vector defines the orientation of the

height function. In the following, we will assume (without restricting the method) that $|n_y| > |n_x|$ and $n_y > 0$ (empty cells at the top of the columns and filled cells at the bottom). For the interface cell ($\Omega_{i,j}$) and its two neighbors ($\Omega_{i+1,j}$ and $\Omega_{i-1,j}$), the top and bottom cell of each column (i, j_t) and (i, j_b) are defined according to the method by Popinet (2009), within the limit of the usual 3×7 stencil. The discrete height functions $h_{i-1,j}$, $h_{i,j}$ and $h_{i+1,j}$ are then obtained by summing up the values of the volume fraction in each column and correcting for the differences in the coordinates of the bottom cells of each column, for example:

$$h_{i-1,j} = \Delta y \left[\sum_{j'=j_b(i-1)}^{j_t(i-1)} C_{i-1,j'} - j_b(i-1) + j - \frac{1}{2} \right]$$

The curvature is then defined by

$$\kappa = \frac{h''}{(1+h'^2)^{3/2}} \tag{4.37}$$

where $h' = (h_{i+1} - h_{i-1})/(2\Delta x)$ and $h'' = (h_{i+1} - 2h_i + h_{i-1})/(\Delta x)^2$. The curvature is required at the cell faces, while in this example it is defined at $(x_i, y_i + h_i \Delta y)$.

In case no valid stencil is found for one of the columns, the same method is applied in the other direction. If this does not work either, the method falls back to a parabolic fitting method through the points that are validated in the above procedure. If the number of valid points (columns) is below 3, however, a parabolic fit is tried through the centroids of the line elements in a 3×3 grid around the cell. The parabolic fitting procedure is normally only used for under-resolved interface structures.

In addition to the method by Popinet (2009), which is referred to in this work by HF1, we have implemented the method by Hernández et al. (2008). They proposed an improvement of the earlier Height Function methods by adapting the stencil according to the local volume fraction field and by introducing small local corrections of the volume fraction. For more details on this method, noted here as HF2, the reader is referred to Hernández et al. (2008), and the subsequent papers (López et al., 2009; López and Hernández, 2010).

The Height Function method gives an estimate for the curvature at the center of the cell indicated by the height function value h_i (most of times this is simply the cell $\Omega_{i,j}$). In order to obtain values at the cell face, we interpolate from the cell-centered values if the curvature is available in both adjoining cells. Otherwise, we just copy the value from the cell containing the curvature. The cell averaging is based on the expression by Renardy and Renardy (2002), which puts the weight on cells containing large portions of the interface and rules out inaccurate tiny interface segments. For example, for the u-component we use:

$$\sigma\kappa \left.\frac{\partial C}{\partial x}\right|_{i+\frac{1}{2},j} = \sigma \left(\frac{w_1\kappa_{i,j} + w_2\kappa_{i+1,j}}{w_1 + w_2} \right) \frac{1}{\Delta x}(C_{i+1,j} - C_{i,j})$$

with $w_1 = C_{i,j}(1 - C_{i,j})$ and $w_2 = C_{i+1,j}(1 - C_{i+1,j})$.

The curvature computed this way is not precise, we therefore use the same formula as in the free CFD software Basilisk (Popinet et al.). The interface curvature at the cell face is given by: $\kappa_{i+\frac{1}{2},j} = (\kappa_{i,j} + \kappa_{i+1,j})/2$ if a valid curvature is available on the two cell centers (i, j) and $(i+1, j)$, while we use the same value as curvature at the cell center if only $\kappa_{i,j}$ or $\kappa_{i+1,j}$ is valid. otherwise, the curvature is set to zero.

4.3.7 Overview of tested methods

In the previous chapters, we have verified the ability of different interface capturing methods in accurately predicting the transport of an interface placed in a known velocity field. From now on, we consider only one level-set (LS) method, two volume-of-fluid (VOF) methods and one moment-of-fluid (MOF) method as in §3.6.1. Note that the local level-set approach of §1.4 is used here in combination with the LS parameters of §3.6.1. We couple each of these interface capturing methods with the Navier-Stokes solver described in the previous section.

The coupling between the interface method and the Navier-Stokes solver is established through the determination of the fluid properties and surface tension force. Taking example from the numerical implementations available in the literature, we have implemented different methods that are described hereafter.

LS-LA: Similar to the method by Vigneaux (2008), based on the level-set (LS) method, the density and viscosity are determined from local averaging (LA) of cell-centered values, Eqs. (4.23) and (4.26). Two methods for the surface tension are derived following the manner in which the curvature is computed:

- **LS-LA1** Following the work by Vigneaux (2008), the curvature is directly determined at the cell-face from normal vectors derived at cell centers and vertices, Eqs. (4.33).
- **LS-LA2** The curvature is determined at the cell centers using the central differencing formula (4.36) proposed by Sussman et al. (1999), and then interpolated at the cell-face using the formula (4.17) and (4.18).

In the absence of surface tension forces, these two variants are identical, and we note them as LS-LA.

LS-CSF: This level-set implementation takes its features from the popular method by Chang et al. (1996). The smoothed Heaviside function (4.5) is used in the determination of fluid properties. The surface tension is implemented using the continuous surface force (CSF) approach, Eq. (4.30). The curvature is computed by expression (4.35).

LS-GFM: This is the sharp-interface level-set method in which the density and surface tension at the cell face is determined using the Ghost Fluid Method (GFM), see §4.2.4. Depending on the manner of determining the viscosity, we have the following variants:

- **LS-GFM1** Following the work by Son and Dhir (2007), the viscosity is determined in a sharp-interface manner using the simplified expressions (4.29) of the GFM, while the curvature at the cell center is computed from $\kappa = -\nabla \cdot \vec{n}$ in a straightforward manner.
- **LS-GFMϵ** The viscosity is determined using the smoothed Heaviside function (4.5) and the curvature is computed directly as in Chang et al. (1996), Eq. (4.35).
- **LS-GFMk** Following the work by Desjardins et al. (2008), the viscosity is determined using the smoothed Heaviside function (4.5) and the curvature and level-set normals are computed using the least-squares method of Marchandise et al. (2007), Eq. (4.10).
- **LS-GFM2** In the absence of surface tension, these two previous methods are identical and are simply denoted as LS-GFM2. In contrast to the method LS-GFM1, the viscosity is smoothed near the interface.

For the volume-of-fluid methods, the density and viscosity at the cell-faces are determined using the volume-of-fluid weighted average, Eqs. (4.21) and (4.28). No smoothing kernel is used, except when mentioned explicitly. We have used different methods depending on the volume-of-fluid advection and reconstruction method and the curvature calculation:

VOF-HF1: The volume-of-fluid is advected using the consistent and conservative implicit/explicit geometrical mapping given by Algorithm 2.1 in §2.5. The ELVIRA reconstruction by Pilliod and Puckett (2004) is used. This interface method (VOF) has been already defined in §3.6.1. The curvature is computed using the Height Function method by Popinet (2009), named HF1, and we use the implementation from the Basilisk code (Popinet et al.).

MOF-HF1: We use the moment-of-fluid advection and reconstruction proposed by Dyadechko and Shashkov (2005), already detailed in §3.6.1. The curvature is determined using the Height Function method HF1 by Popinet (2009).

VOF-HF2: The curvature is computed using the slightly different Height Function method proposed by Hernández et al. (2008), named HF2. Furthermore, the density at cell-faces used in the surface tension term (Eq. 4.31) is computed as the mean density. For example, the density at the $(i+1/2, j)$-th cell face is defined as:

$$\rho_{i+1/2,j} = (\rho_l + \rho_g)/2 \tag{4.38}$$

The interface method (VOF) is identical to that of VOF-HF1.

VOF-WY: The volume-of-fluid is advected following the alternative conservative and consistent scheme by Weymouth and Yue (2010) described in §3.6.1. As in VOF-HF1, the curvature is determined using the Height Function method HF1 by Popinet (2009).

VOF-WYC: This method has been largely inspired by (and is as close as possible to) the Basilisk code (Popinet et al.). A collocated grid setting is used, in which all variables including the velocity are located at the cell center. This solver, based also on a predictor-corrector time scheme, is described in detail in §B.1. The flow solver is coupled to the VOF method, which is identical to that used for the VOF-WY method.

The different methods are summarized on Table 4.1.

Table 4.1: Description of the different numerical methods for the Navier-Stokes equations for two-phase flows: expressions used to compute the curvature κ, the density at the cell-face and the viscosity.

Method	$\boldsymbol{\kappa}$	$\boldsymbol{\rho}$	$\boldsymbol{\mu}$	**Conv. term**	**Time adv.**
LS-LA1	Eq. (4.33)	Local Averaging		2nd-order ENO §4.3.4	Crank-Nicolson predictor-corrector §4.3.1
LS-LA2	Eq. (4.36)	Eq. (4.23)	Eq. (4.26)		
LS-CSF	Eq. (4.35)	H_ϵ Eq.(4.22)	H_ϵ Eq. (4.27)		
LS-GFMϵ		Eqs. (4.17) and (4.18)	H_ϵ Eq. (4.27)		
LS-GFMk	Eq. (4.10)				
LS-GFM1	Eq. (4.34)		Eq. (4.29)		
VOF-HF1	HF1	Weighted-Average			
MOF-HF1					
VOF-WY		Eq. (4.21)	Eq. (4.28)		
VOF-HF2	HF2				
VOF-WYC	HF1				

LS-LA	if $\sigma = 0$, then LS-LA1 and LS-LA2 are identical.
LS-GFM2	if $\sigma = 0$, then LS-GFMk and LS-GFMϵ are identical.
HF1	Height Function method by Popinet (2009).
HF2	Height Function method by Hernández et al. (2008). The density at the cell-faces in the surface tension term (Eq. 4.31) is defined as in Eq. (4.38).

4.4 Results and discussions

4.4.1 Spurious currents in a stationary circular bubbles

In the absence of gravity, a circular bubble will reach a state of equilibrium between surface tension and pressure forces, with the flow field at rest and a difference between the pressure inside and outside of the bubble given by the Young-Laplace equation. In numerical simulations of this basic two-phase flow problem, the velocity will not be exactly zero, but show fluctuations known as parasitic or spurious currents. These currents can be quite strong in certain cases and have a strong impact on the dynamic behavior of the bubbles when simulating two-phase flows in general. The evaluation of the magnitude of the spurious currents is therefore a well known test case and has been employed in many works involving the development of numerical schemes for two-phase flows.

We will adopt the flow parameters used in Francois et al. (2006), Herrmann (2008) and Popinet (2009). The computational domain is defined as the unit square with symmetry boundary conditions on all four boundaries. The bubble is located in the center of the domain and has a diameter of 0.4. The surface tension is set to $\sigma = 73$ and the density outside the bubble $\rho_1 = 1$. The density inside the bubble is defined by the density ratio ρ_2/ρ_1 which can be varied between 0 and 1. One can generally identify two contributions to the production of the spurious currents: the imbalance at the discrete level between the surface tension and pressure forces, and the error committed when estimating the curvature.

4.4.1.1 Spurious currents due to unbalanced forces

Francois et al. (2006) presented some results after one time step $\Delta t = 10^{-6}$ with $\mu_1 = \mu_2 = 0$ in which the curvature in the computation is replaced by the exact value $\kappa = 1/R$. If the numerical scheme is able to reproduce an exact equilibrium, the resulting velocity field will remain at rest and the pressure difference will be exactly $\Delta p = 36.5$. For this case, results obtained with the Ghost Fluid Methods (LS-GFMϵ, LS-GFMk and LS-GFM1) are nearly identical. The same is true for the VOF methods, and also for those based on the local averaging (LS-LA1 and LS-LA2). The results are given in Table 4.2.

All the methods reproduce the exact balance in which the velocity field remains zero to machine precision. Indeed, in these schemes the surface tension term is discretized in the same manner as the pressure gradients, as described by the balanced-force approach (Francois et al., 2006).

Methods based on the Ghost Fluid Method (LS-GFM) and the local averaging (LS-LA) reproduce the exact pressure field within machine precision. With the continuous surface force method, however, the pressure field is smeared out and a significant error remains. But these errors are concentrated near the interface and an exact pressure difference is recovered far from the interface, although this cannot be observed in the presented results.

4.4.1.2 Spurious currents due to inaccurate curvature computation

The balanced-force approach allows an exact balance between the surface tension and pressure terms, but only when the curvature is constant. Errors in the estimation of the curvature also lead to spurious currents. The following results are therefore obtained with the full method in which the curvature is computed in different manners as shown on Table 4.1. We will first present the results after one time step of $\Delta t = 10^{-6}$ on a 40×40 grid, see Table 4.3. In this case, the initial level-set and volume-of-fluid fields are exact and the spurious currents are mainly due

Table 4.2: Spurious currents of a stationary circular bubble at zero viscosity and curvature replaced by its exact value. Maximum absolute value of the velocity and maximum error on the pressure field after one time step of $\Delta t = 10^{-6}$. Grid size 40×40 ($R/h = 10$).

Method	ρ_2/ρ_1	$\|\boldsymbol{u}\|_{\max}$	$\|\boldsymbol{p} - \boldsymbol{p}_{exact}\|_{\max}$
VOF-HF1	1	1.71×10^{-17}	14.97
	10^{-3}	3.27×10^{-16}	14.97
	10^{-5}	9.88×10^{-16}	14.97
VOF-WYC	1	1.60×10^{-17}	14.97
	10^{-3}	3.16×10^{-16}	14.97
	10^{-5}	9.02×10^{-16}	14.97
LS-CSF	1	1.62×10^{-17}	16.42
	10^{-3}	1.16×10^{-15}	16.42
	10^{-5}	2.26×10^{-14}	16.42
LS-LA	1	2.01×10^{-17}	2.22×10^{-11}
	10^{-3}	6.28×10^{-16}	3.02×10^{-11}
	10^{-5}	1.42×10^{-14}	5.78×10^{-11}
LS-GFM	1	2.01×10^{-17}	2.22×10^{-11}
	10^{-3}	6.75×10^{-16}	2.93×10^{-11}
	10^{-5}	5.53×10^{-15}	5.88×10^{-11}

to the error in the curvature estimation. It is seen that the LS-LA2, LS-GFMϵ and the VOF methods give the most precise results in this case, nearly independent of the density ratio.

The results obtained with a smoothing kernel are presented on Table 4.4. For the VOF methods determining the curvature through the HF1 method, the use of the smoothed fraction $\tilde{C}$ instead of C has not much influence on the maximum velocity and the pressure difference. On the other hand, with the VOF-HF2 method, the smoothing kernel slightly increases the amplitude of the spurious currents. This is probably due to the use of the mean density $(\rho_g + \rho_l)/2$ in the surface tension term, whereas Eq. (4.21) is used for the density in the pressure equation. According to Francois et al. (2006), it is then not possible to obtain a correct balance between surface tension and pressure forces. (see Table 4.1).

4.4.1.3 Time evolution of the spurious currents

Now we turn our attention to the time evolution of the spurious currents in fluids with non-zero viscosity. The additional relevant non-dimensional number is the Laplace number defined as:

$$La \equiv \frac{\sigma D}{\rho_1 \nu_1^2}$$

We set $\mu_1 = \mu_2 = 0.1$ in our computations, in addition to $D \equiv 0.4$ and $\sigma = 73$, which leaves the density as the only free parameter to choose the Laplace number. Three characteristic time scales can be defined in this problem:

$$t_\sigma = \sqrt{\frac{\rho_1 D^3}{\sigma}}; \qquad t_\nu = \frac{D^2}{\nu} \quad \text{and} \qquad t_{ref} = \frac{\mu_1 D}{\sigma}$$

Table 4.3: Spurious currents of a stationary circular bubble at zero viscosity. Maximum absolute value of the velocity and maximum error on the pressure field after one time step of $\Delta t = 10^{-6}$. Grid size 40×40 ($R/h = 10$).

Method	$\boldsymbol{\rho_2/\rho_1}$	$\boldsymbol{\|u\|_{\max}}$	$\boldsymbol{\|p - p_{exact}\|_{\max}}$
VOF-HF1	1	1.02×10^{-7}	14.85
and	10^{-3}	1.73×10^{-7}	14.88
VOF-WY	10^{-5}	1.73×10^{-7}	14.88
	1	8.86×10^{-8}	14.86
VOF-HF2	10^{-3}	1.87×10^{-7}	21.76
	10^{-5}	1.87×10^{-7}	21.78
	1	9.97×10^{-8}	14.85
VOF-WYC	10^{-3}	1.68×10^{-7}	14.88
	10^{-5}	1.68×10^{-7}	14.88
	1	2.06×10^{-7}	17.24
LS-CSF	10^{-3}	1.64×10^{-5}	17.26
	10^{-5}	5.88×10^{-4}	17.26
	1	3.33×10^{-6}	0.47
LS-LA1	10^{-3}	1.09×10^{-5}	0.78
	10^{-5}	1.10×10^{-5}	0.80
	1	2.52×10^{-7}	0.15
LS-LA2	10^{-3}	5.45×10^{-7}	0.22
	10^{-5}	5.46×10^{-7}	0.22
	1	3.48×10^{-7}	0.29
LS-GFM1	10^{-3}	1.21×10^{-6}	0.32
	10^{-5}	1.22×10^{-6}	0.32
	1	1.27×10^{-7}	6.60×10^{-2}
LS-GFMϵ	10^{-3}	4.79×10^{-7}	4.07×10^{-2}
	10^{-5}	4.81×10^{-7}	4.06×10^{-2}
	1	4.50×10^{-7}	0.28
LS-GFMk	10^{-3}	9.19×10^{-7}	0.39
	10^{-5}	9.23×10^{-7}	0.39

Table 4.4: Spurious currents of a stationary circular bubble at zero viscosity. Maximum absolute value of the velocity and maximum error on the pressure field after one time step of $\Delta t = 10^{-6}$ using different smoothing kernels with the VOF-HF2 method. Grid size 40×40 ($R/h = 10$).

Kernel	$\boldsymbol{\rho_2/\rho_1}$	$\boldsymbol{\lvert u\rvert_{\max}}$	$\boldsymbol{\lvert p - p_{exact}\rvert_{\max}}$
none	10^{-3}	1.87×10^{-7}	21.76
	10^{-5}	1.87×10^{-7}	21.78
K_8	10^{-3}	5.94×10^{-6}	21.06
	10^{-5}	6.02×10^{-6}	21.07
δ_ϵ	10^{-3}	8.04×10^{-6}	20.42
	10^{-5}	8.10×10^{-6}	20.44
Vertex-Avg	10^{-3}	9.50×10^{-6}	19.44
	10^{-5}	9.56×10^{-6}	19.45

which are related by $t_\nu/t_\sigma = \sqrt{La}$ and $t_\nu/t_{ref} = La$. Popinet (2009) presented results for a time period of $t = t_\nu$ and we will use the same value for our simulations. Desjardins et al. (2008) used a time period of $250t_{ref}$, which is a shorter period for high values of the Laplace number ($La > 250$). The velocity is usually presented in terms of the capillary number,

$$Ca \equiv \frac{|u|_{\max}\mu}{\sigma}$$

We will also adopt this definition to present our results.

Note that VOF-HF1, VOF-WYC and VOF-WY produce almost identical results, which is due to the fact that the curvature and fluid properties are determined in the same manner. Only the results obtained with the VOF-HF1 method are therefore presented. Note also that the time step have only a minor influence on the time evolution of the spurious currents. For brevity sake, these results are not shown here.

Whole domain versus quarter domain simulations

The results for the tested methods are shown on Fig. 4.2 for a density ratio equal to unity, $\text{CFL}_\sigma = 1$ and La= 12000. For each method, we have performed a simulation on the whole domain and the quarter of the domain (assuming symmetry conditions at the boundaries).

The simulations over the whole domain (red lines) often show an increase of the spurious currents towards the end of the simulation. Only the LS-CSF and LS-LA1 methods seem to be able to maintain the equilibrium solution over a long time period. However, when the quarter of the domain is simulated, all the methods maintain a constant level of the spurious currents. These observations can be explained by the fact that the bubble tends to drift away from the origin when the whole domain is simulated. This effect is suppressed by computing only a quarter of the bubble, as is done in Popinet (2009), in which case the bubble center is fixed by the symmetry conditions and cannot move around. The MOF-HF1 method shows cycles of increase and decrease in the spurious currents. In fact these spurious currents are due to round-off errors in the interface method (MOF) and occur at very small CFL numbers, as pointed out in §2.6.2. These errors may be suppressed in a future implementation by more carefully programming the MOF method.

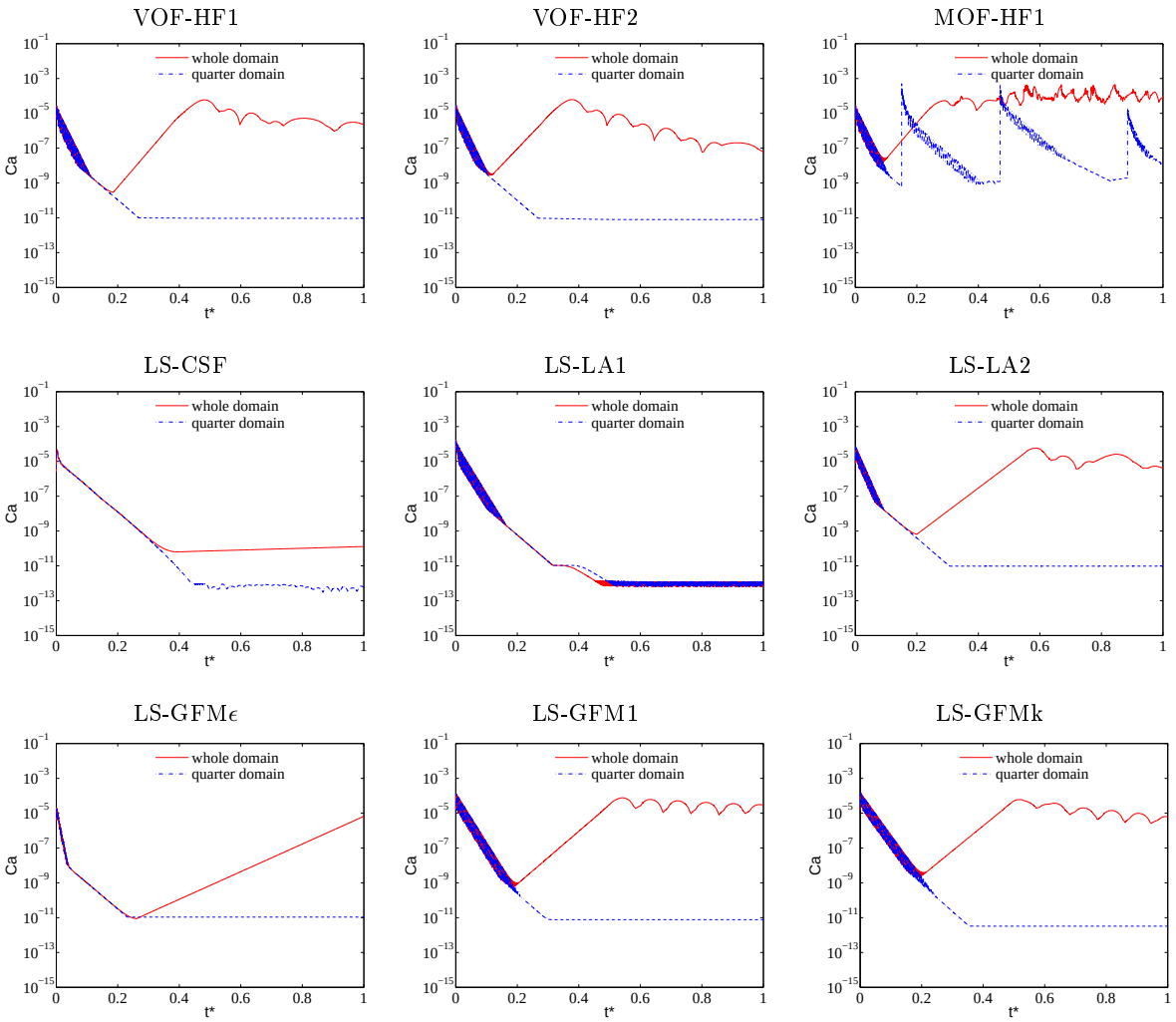

Figure 4.2: Spurious currents of a stationary circular bubble: evolution of the maximum velocity for $La = 12000$ when the whole domain is simulated vs. the quarter of the bubble. The time is made non-dimensional by the viscous time scale t_ν. The density ratio is $\rho_1/\rho_2 = 1$. Grid size 32×32 ($R/h = 6.4$). No reinitialization of the level-set field. The VOF-WY and VOF-WYC methods yield similar results as the VOF-HF1 method.

Influence of the Laplace number and density ratio

Now, we investigate the influence of the Laplace number and the density ratio on the spurious currents. The density ratio has been varied from 1 to 1000, and the Laplace number from 120 to 120000.

We have noticed that for some methods the results are insensitive to variations of the density ratio, and the spurious currents are mainly dependent on the Laplace number. Fig. 4.3 plots the temporal evolution of the spurious currents produced with these methods for a unit density ratio. All methods, except the continuous surface force approach (LS-CSF), reach an equilibrium state with a very low amplitude of spurious currents over the entire range of Laplace numbers.

The remaining methods (MOF-HF1, LS-LA1, LS-LA2, LS-GFMk and LS-GFM1) are more or less affected by the density ratio and/or the Laplace number. Fig. 4.4 shows the results for different density ratios and for three different Laplace numbers: 120 (left column), 1200 (central column) and 120000 (right column).

The MOF-HF1 method gives excellent results at low values to the Laplace number, but clearly

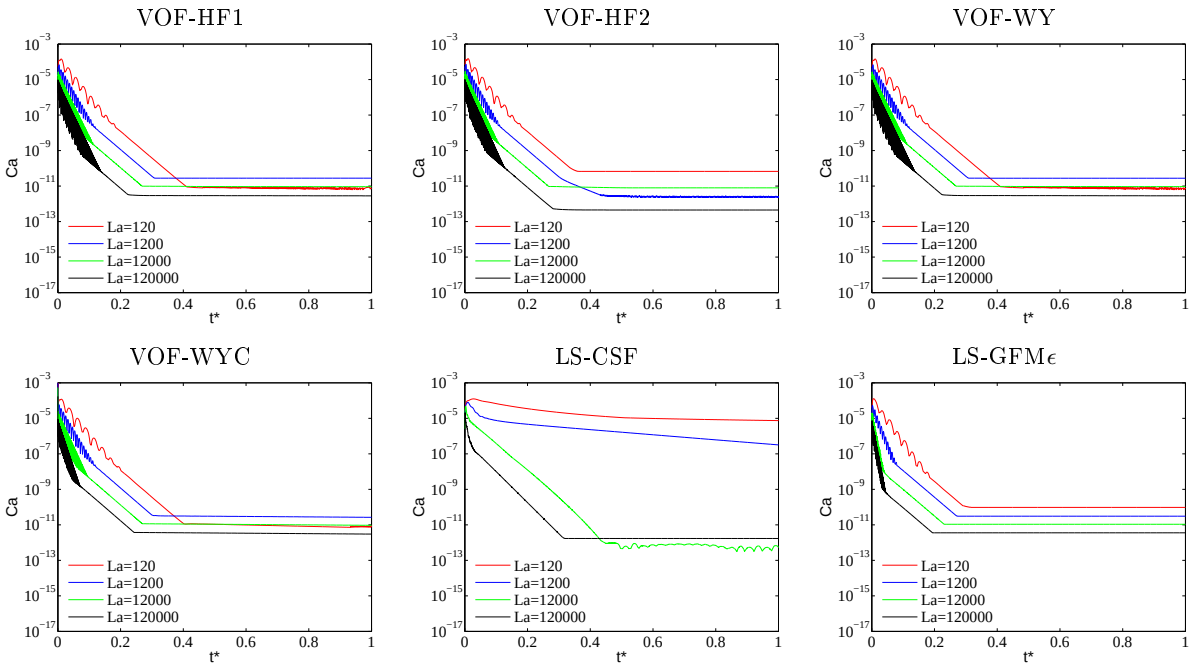

Figure 4.3: Spurious currents of a stationary circular bubble: influence of the Laplace numbers on the evolution of the maximum velocity for a uniform density. The time is made non-dimensional by the viscous time scale t_ν. Only a quarter of the bubble is simulated. The grid size is 16×16 ($R/h = 6.4$) and $\mathrm{CFL}_\sigma = 1.0$. No reinitialization of the level-set field.

suffers from numerical instability at higher values. The VOF-HF1 method, which differs only in the way the interface normal is reconstructed, did not show this numerical instability (Fig. 4.3). Again, a future improvement of the MOF implementation might solve this problem. The methods LS-LA1, LS-GFM1 and LS-GFMk also have a difficulty in maintaining the equilibrium state at La=120000. Only the LS-LA1 gives a stable equilibrium state for all density ratios and Laplace numbers. In conclusion, all VOF methods and the LS-GFMϵ and LS-LA2 level-set methods are able to produce a correct equilibrium state of a stationnary bubble in a wide range of parameters. All other methods may produce significant spurious currents depending on the values of the density ratio and Laplace number.

Influence of the level-set reinitialization

One important parameter omitted in the previous results (and generally also in results shown in the literature) is the reinitialization of the level-set field. Fig. 4.5 shows the more realistic case of the level-set method with reinitialization every Nreinit time step. It is seen that the reinitialization procedure introduces a perturbation of the equilibrium state and a sudden increase of the spurious currents. When the level-set is reinitialized each time step, the magnitude of the spurious currents reaches a constant level. Increasing Nreinit helps to diminish the spurious currents, but the levels are still much higher than that obtained without reinitialization or with the VOF methods. Among the level-set methods, LS-GFMϵ and LS-LA2 still produce the best results, but with reinitialization the LS-GFMk and LS-GFM1 methods are not far behind.

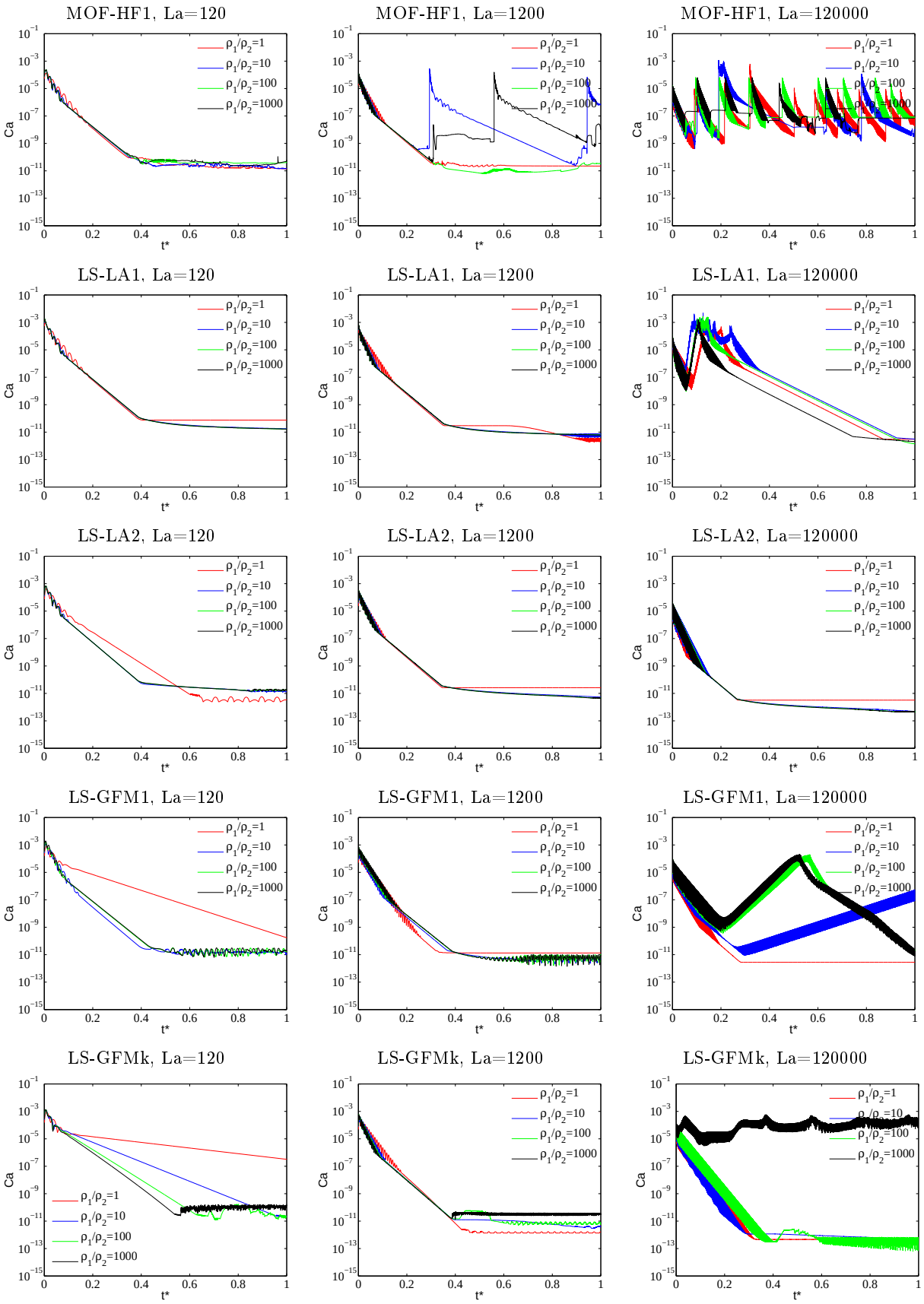

Figure 4.4: Spurious currents of a stationary circular bubble: influence of density ratio on the evolution of the maximum velocity for $La = 120$. The time is made non-dimensional by the viscous time scale t_ν. Only a quarter of the bubble is simulated. The grid size is 16×16 ($R/h = 6.4$) and $\mathrm{CFL}_\sigma = 1.0$. No reinitialization of the level-set field.

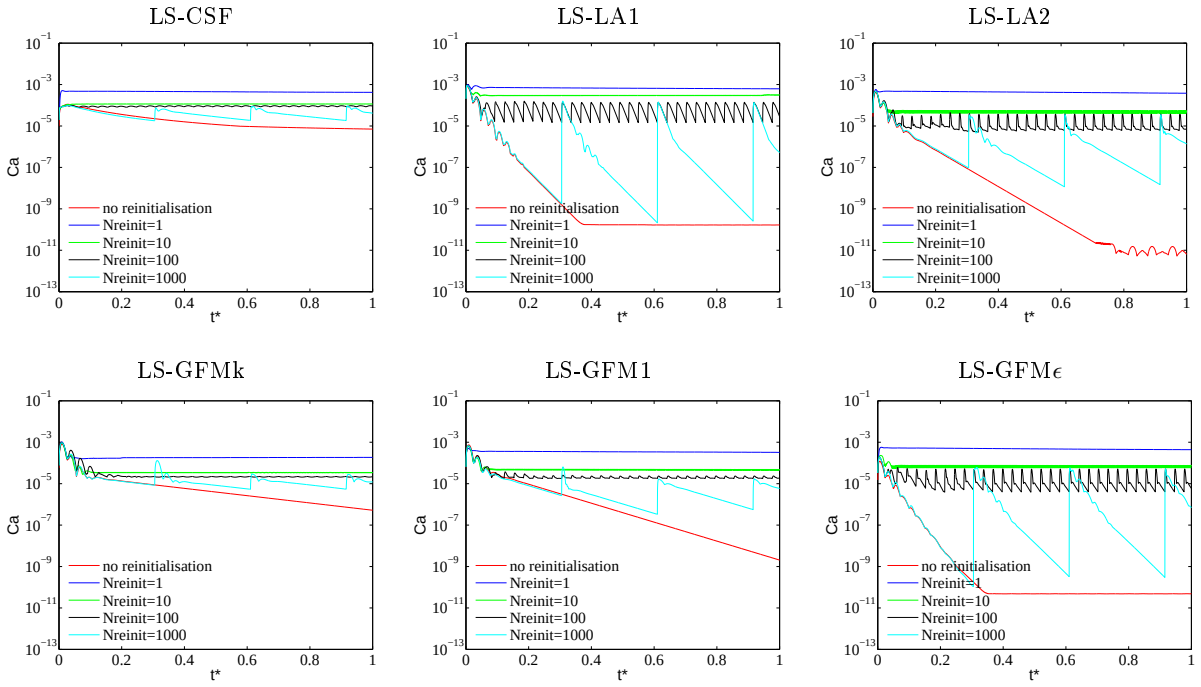

Figure 4.5: Spurious currents of a stationary circular bubble: influence of the reinitialization of the level-set field on the evolution of the maximum velocity for $La = 120$. The time is made non-dimensional by the viscous time scale t_ν. Only a quarter of the bubble is simulated on a 16×16 grid size ($R/h = 6.4$) with $\mathrm{CFL}_\sigma = 1.0$. The density ratio $\rho_1/\rho_2 = 1$.

4.4.2 Oscillating drops

Now, we study the oscillating drop problem. An initially elliptical drop is placed in a square domain with zero gravity. Due to the surface tension forces, the drop will oscillate around its equilibrium state (the perfectly circular drop at rest). This test problem has been widely used to validate the surface tension implementation in flow solvers. The case for two-dimensional cartesian grids can be found for example in Torres and Brackbill (2000); Shin et al. (2005); Francois et al. (2006); Tong and Wang (2007); Herrmann (2008) and Wang and Tong (2010) and for cylindrical coordinate systems in Sussman and Puckett (2000); Aulisa et al. (2006); Sussman et al. (2007) and Jemison et al. (2013).

In the case of viscous fluids, the drop oscillates with a decreasing amplitude due to the viscous dissipation. But, in the absence of the viscosity, there will be no damping. This can only be obtained, however, with non-diffusive numerical methods. In the following, we will treat both cases: zero viscosity in §4.4.2.1 and viscous fluids in §4.4.2.2.

4.4.2.1 Inviscid drop oscillations

Torres and Brackbill (2000) considered a drop with an initial radius given by:

$$r(\theta) = r_0 + \alpha \cos(n\theta) \tag{4.39}$$

where θ runs from 0 to 2π. In the case of inviscid fluids, the theoretical frequency of oscillations is (Fyfe et al., 1988):

$$\omega_n^2 = \frac{(n^3 - n)\sigma}{(\rho_d + \rho_e) r_0^3} \tag{4.40}$$

with ρ_d the drop density and ρ_e that of its exterior.

Torres and Brackbill (2000) used a density ratio of 100, which is relatively small. A similar test case with a density ratio of 1000 can be found in the test suite of the free CFD Software Gerris (Popinet). We use the same parameters: $r_0 = 0.1$, $\alpha = 0.005$, $n = 2$. The initial interface is therefore an ellipse of semi-major and semi-minor axis 0.105 and 0.095, respectively. The domain is 2×2 and only a quarter is simulated with symmetric boundary conditions. The drop density is equal to unity while that of its surrounding fluid is set to 0.001, and the surface tension coefficient σ is set to unity. The simulations are performed until time $t = 1$.

Since the drop and its surrounding fluid are inviscid and no energy enters the domain boundaries, the total energy is conserved. The kinetic energy should oscillate in time with constant amplitude and frequency in the ideal case. The peaks correspond to the maximum of the kinetic energy which is reached when the drop is circular (maximum velocity and minimum surface tension forces), and the minima are attained when the drop is an ellipse (minimum velocity and high curvature and surface tension). The total kinetic energy is computed as:

$$K_E = \frac{1}{2} \int_\Omega \rho |\vec{u}|^2 dv = \frac{\Delta x \Delta y}{2} \sum_{i,j} \rho_{i,j} (\overline{u}_{i,j}^x + \overline{v}_{i,j}^y)^2 \tag{4.41}$$

where $\overline{u}_{i,j}$ and $\overline{v}_{i,j}$ are the velocity components at the cell center. In the case of a staggered grid, they are computed by averaging of the values at the cell faces, for example $\overline{u}_{i,j} = (u_{i+\frac{1}{2},j} + u_{i-\frac{1}{2},j})/2$. The density at the cell center, $\rho_{i,j}$, is determined according to the approach the method uses (see §4.3.3).

We tested the different methods in Table 4.1 with a CFL_σ equal to 1.0 on different grid sizes ranging from 32^2 to 256^2). We have noticed that the CFL_σ does not have a great influence on the results in the range $[0.2 - 1]$.

The MOF-HF1 method showed a non-physical increase of the kinetic energy for all tested grid sizes due to the spurious currents of high magnitude (see for example the corresponding results to this method on Fig. 4.4 for La=120000). For this reason, the results with this method are not presented here.

The temporal evolution of the total kinetic energy, K_E, on two different grids (64^2 and 256^2) is plotted on Fig. 4.6. The ticks on the x-axis correspond to the theoretical period given by Eq. (4.40). The error in the frequency of oscillation is tabulated on Table 4.5. The errors in mass conservation are given in Table 4.6. Note that for the level-set methods, reinitalization is applied at each time step.

On the 64^2 grid, the amplitude of oscillations decreases in time for all VOF methods. It decreases even on the refined grid of 256^2 for the VOF-HF1, VOF-WY and VOF-HF2 methods, which is perhaps due to velocity interpolation from cell faces to cell centers. The VOF-WYC method seems to be less diffusive than the other methods, probably to the centered framework that this method uses. On the other hand, the frequency of the oscillations is better predicted with the VOF-HF1, VOF-HF2 and VOF-WY methods than with the VOF-WYC method. The results obtained with the free CFD software Gerris (Popinet) are also plotted on Fig. 4.6. The kinetic energy decreases less than with our VOF methods, which is due probably to the cell centered framework of Gerris. However, concerning the frequency error [2] our volume-of-fluid implementation seems to be more accurate on grid sizes finer than 64^2.

A strong damping of the oscillations of the kinetic energy is observed on the 64^2 grid with the LS-CSF and LS-LA1 methods. For the LS-LA1 method, this could be due to the increase of the spurious currents (see the corresponding results on Fig. 4.4 for La=120000). For the LS-CSF method, we see on Table 4.6 that the mass gain is very important for coarse grids (around 42% and 21% on the 32^2 and 64^2 grids respectively). Knowing that the density of the drop is 1000 times that of the surrounding fluid, this increases the kinetic energy until it stops from oscillating. The other level-set methods exhibit much smaller mass errors (around 10% on coarse grids). The LS-LA2 method is again more accurate than the LS-LA1 method for this inviscid problem. The Ghost Fluid Methods (LS-GFMk, LS-GFMϵ and LS-GFM1) slightly overestimate the kinetic energy on the coarse grid of 64^2, but give accurate results on fine grids.

Effect of the reinitialization

The perturbation introduced by the reinitialization of the level-set field can be reduced by increasing Nreinit. This is done here by using a CFLreinit as is described in §1.6.2.2. The results are significantly enhanced only for the LS-CSF method, for which the temporal evolution of the kinetic energy is plotted on Fig. 4.7. The mass error is now very small (see Table 4.6) compared to the case where the level-set is reinitialized after each advection. It does not exceed the 8% on the 32^2 grid. The error on the frequency (presented on Table 4.5) are also much better than when the level-set is reinitialized each time step.

Effect of the smoothing kernel

The kinetic energy profile obtained with the vertex-averaged smoothing kernel is plotted on Fig. 4.8 for the two grid sizes 64^2 and 256^2. The frequency error is summarized on Table 4.7 for all the smoothing kernels. The smoothing kernel generally improves the accuracy of the VOF methods, both in terms of frequency error and when comparing the decrease of kinetic energy without smoothing kernel. For the VOF-WYC method, the best results are obtained using the

[2] Note that the frequency errors in Table 4.6 are different from those presented on the Gerris website, since we have used a different method to compute the error.

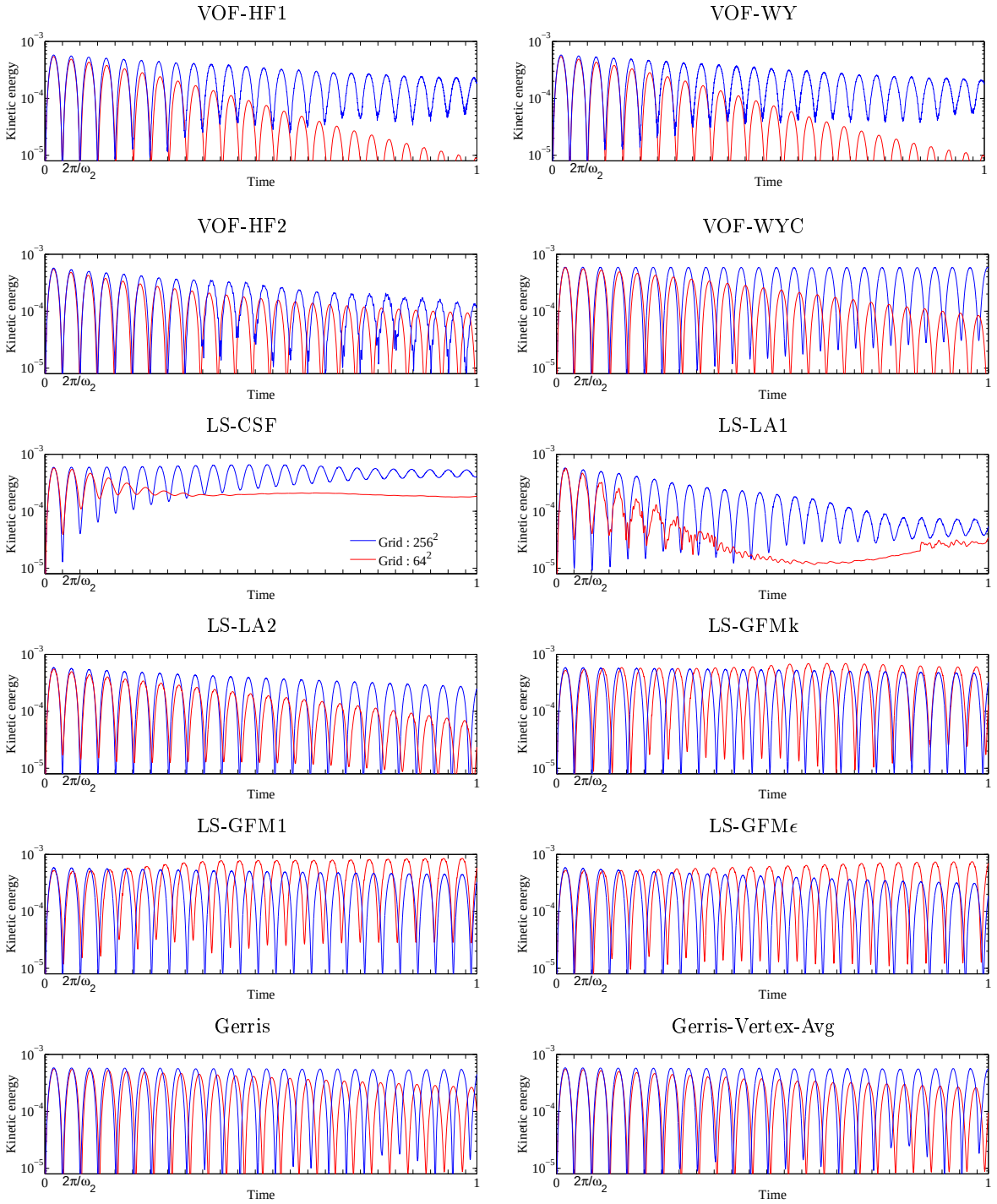

Figure 4.6: Inviscid oscillating drop: total kinetic energy versus time on different grid sizes with $\mathrm{CFL}_\sigma = 1$. Red: 64^2 grid and blue: 256^2 grid. The ticks on the x-axis correspond to the theoretical period of oscillations. The level-set field is reinitialized at each time step.

Table 4.5: Inviscid oscillating drop: error in the oscillation frequency (in %) for the different methods and grid sizes. In the case of a fluctuating frequency, the error has not been computed. The level-set field is reinitialized at each time step except for the last row.

Method/Grid	32^2	64^2	128^2	256^2
VOF-HF1	–	1.08	0.01	0.09
VOF-WY	–	1.07	0.003	0.009
VOF-HF2	–	1.40	0.24	0.09
VOF-WYC	–	2.59	0.94	0.41
LS-CSF	–	–	–	1.65
LS-LA1	–	–	1.60	1.25
LS-LA2	5.95	1.95	0.66	0.93
LS-GFMk	–	–	2.28	1.41
LS-GFMϵ	–	5.49	2.01	1.37
LS-GFM1	–	–	2.21	1.34
Gerris	4.71	3.43	1.84	1.17
Gerris-Vertex-Avg	2.87	1.79	0.70	0.59
LS-CSF, CFLreinit=0.2	–	3.89	2.14	0.35

Table 4.6: Inviscid oscillating drop: Mass error $E_M(\%)$ of the level-set methods at the final time on different grid sizes with $\mathrm{CFL}_\sigma = 1$. The level-set field is reinitialized at each time step except for the last row.

Method/Grid	32^2	64^2	128^2	256^2
LS-GFMk	9.64	−0.70	−1.77	−0.83
LS-GFMϵ	10.34	−1.50	−1.09	0.63
LS-GFM1	9.03	−1.13	−1.10	−0.56
LS-LA1	5.91	6.85	9.58	1.83
LS-LA2	10.71	2.35	2.17	0.12
LS-CSF	42.31	20.85	8.33	4.89
LS-CSFwith CFLreinit=0.2	9.01	7.37	4.88	−0.58

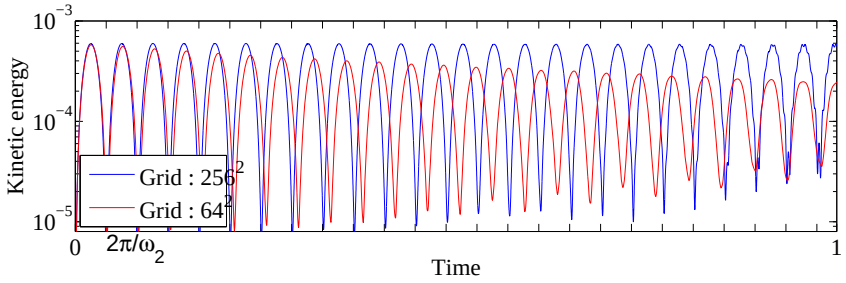

Figure 4.7: Inviscid oscillating drop: kinetic energy versus time for different grid sizes using LS-CSF method with $\mathrm{CFL}_\sigma = 1$ and CFLreinit= 0.2.

vertex-averaged volume fraction. For the VOF-HF1 and VOF-WY methods, the δ_ϵ-kernel gives the best result in terms of frequency error. The VOF-HF2 method results, on the other hand, are deteriorated with the use of the smoothing kernel, probably due to the usage of the mean density in the surface tension term.

Table 4.7: Inviscid drop: frequency error (in %) for the different volume-of-fluid methods using different smoothing kernels.

Method	Kernel/Grid	64	128	256
VOF-HF1	none	1.08	0.01	0.09
	K_8	0.70	0.60	0.35
	δ_ϵ	0.02	0.30	0.20
	Vertex-Avg	1.76	1.12	0.62
VOF-WYC	none	2.59	0.94	0.41
	K_8	1.34	0.23	0.11
	δ_ϵ	1.85	0.54	0.20
	Vertex-Avg	0.47	0.25	0.15
VOF-WY	none	1.07	0.003	0.009
	K_8	0.71	0.62	0.40
	δ_ϵ	0.02	0.30	0.19
	Vertex-Avg	1.78	1.12	0.59

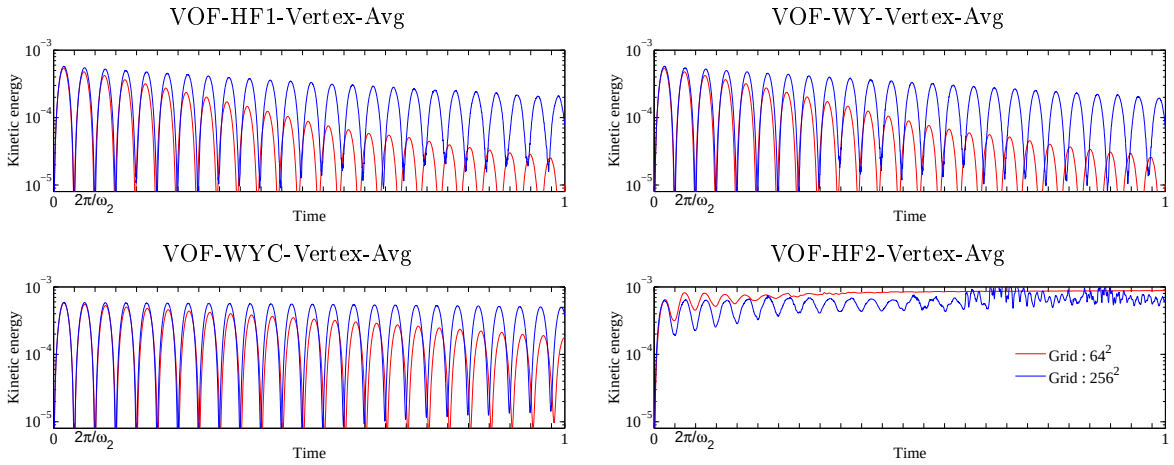

Figure 4.8: Inviscid drop: total kinetic energy versus time on two different grids (Red: 64^2 and Blue: 256^2). Effect of the Vertex-Avg smoothing kernel on the volume-of-fluid methods with $\mathrm{CFL}_\sigma = 1$. The ticks on the x-axis correspond to the theoretical period of oscillations.

4.4.2.2 Viscous drop oscillations

Now, we consider the same configuration as in Torres and Brackbill (2000), where the initial interface is an ellipse given by $x^2/9 + y^2/4 = 1$. The drop of density $\rho_d = 1.0$ and viscosity $\mu_d = 0.01$ is surrounded by a fluid with density $\rho_e = 0.01$ and viscosity $\mu_e = 5 \cdot 10^{-5}$. The surface tension coefficient is $\sigma = 1.0$. The drop is centered in a $[0, 20]^2$ domain with symmetric boundary conditions. Under the influence of the surface tension forces, the drop will oscillate. Due to the viscous nature of both fluids, the total kinetic energy will decrease.

The temporal evolution of the total kinetic energy is plotted on Fig. 4.9. The exponential envelope of the kinetic energy provided by Torres and Brackbill (2000) using a Lagrangian marker method is also presented (solid red lines). The VOF based methods predict correctly the kinetic energy evolution at the onset of the simulation, except for the VOF-WYC method. For this method, viscous effects are somehow underestimated resulting in slowly vanishing oscillations. The MOF-HF1 method is working better with this test case, except at the end of the simulation where some increase in the kinetic energy and an apparent variation in the frequency of the simulation could be remarked. The LS-CSF method underestimates the damping of the kinetic energy, which could be related to the mass gain of the drop. The LS-GFM1, LS-GFMk and LS-GFMϵ methods overestimates the kinetic energy towards the end of the simulations. The LS-LA1 method is very diffusive and the LS-LA2 method seems to work well for the inviscid case as well as for the viscous one.

The relative mass error of the level-set methods is plotted on Fig. 4.10. The difference with the inviscid case is that the mass error is decreasing for most methods then increases, while it is monotonically increasing or decreasing in the inviscid case (not shown here).

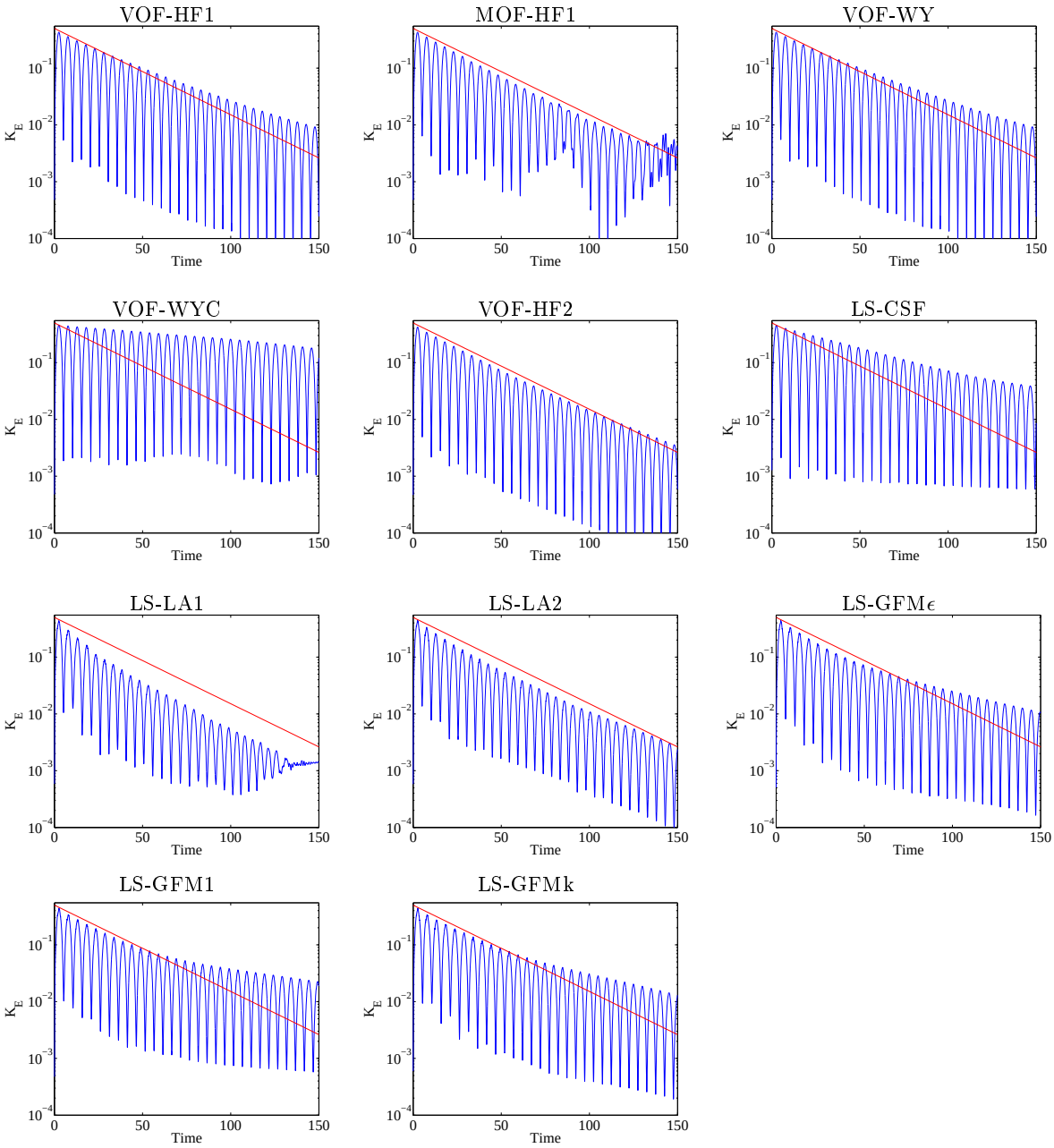

Figure 4.9: Temporal evolution of the total kinetic energy (blue) for the viscous droplet oscillations test case for the different methods on a 64^2 grid. $\mathrm{CFL}_\sigma = 1.0$ and $\mathrm{CFL} = 0.5$. The level-set is reinitialised after each advection step, while no kernel smoothing is used for the volume-of-fluid based methods. In red, the exponential envelope of the kinetic energy versus time using a front tracking method according to Torres and Brackbill (2000, data with the unconnected points).

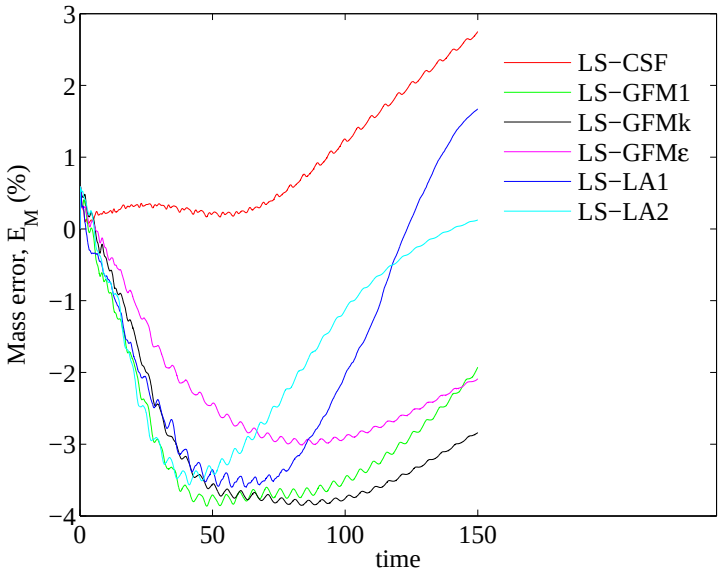

Figure 4.10: Relative mass error of the viscous drop as a function of time for the different level-set methods on a 64^2 grid. $\mathrm{CFL}_\sigma = 1.0$. The level-set is reinitialised after each advection step

4.4.3 Capillary standing wave

The capillary wave test case verifies the coupling between the surface tension, inertial effects and viscosity. In a rectangular domain with zero gravity, two fluids A and B are separated by a horizontal interface. A sinusoidal perturbation of small amplitude λ_0 is imposed on the initial location of the interface. A damped oscillation of the interface will result with angular frequency given by the dispersion relation

$$\omega_0^2 = \sigma k^3/(\rho_1 + \rho_2)$$

where k is the wavenumber. In the case of small amplitudes of the wave and equal kinematic viscosity of the fluids, Prosperetti (1981) and Cortelezzi and Prosperetti (1981) provided an analytical solution which can be written as (Popinet and Zaleski, 1999):

$$\frac{\lambda(\tau)}{\lambda_0} = \frac{4(1-4\beta)\epsilon^2}{8(1-4\beta)\epsilon^2+1}\mathrm{erfc}(\sqrt{\epsilon\tau}) + \sum_{i=1}^{4}\frac{z_i}{Z_i}\frac{\omega_0^2}{z_i^2-\epsilon\omega_0}\exp\left[\frac{(z_i^2-\epsilon\omega_0)\tau}{\omega_0}\right]\mathrm{erfc}\left(z_i\sqrt{\frac{\tau}{\omega_0}}\right) \quad (4.42)$$

where $\lambda(\tau)$ is the amplitude at the dimensionless time $\tau = t\omega_0$ and $\epsilon = \nu k^2/\omega_0$ is the non-dimensional viscosity. The z_i's are the four roots of the algebraic equation

$$z^4 - 4\beta(\epsilon\omega_0)^{\frac{1}{2}}z^3 + 2(1-6\beta)\epsilon\omega_0 z^2 + 4(1-3\beta)(\epsilon\omega_0)^{3/2}z + (1-4\beta)(\epsilon\omega_0)^2 + \omega_0^2 = 0$$

and $Z_i = \prod_{j=1,j\neq i}^{4}(z_j - z_i)$. The dimensionless parameter β is defined as $\beta = \rho_1\rho_2/(\rho_1+\rho_2)^2$.

In the following, we consider that the initial perturbation is a cosine wave of amplitude $\lambda_0 = L/100$ where $L = 1$ is the domain width. The height of the computational domain is three times its width, which reduces the boundary effects (Popinet, 2009). The initial position of the interface is given by $y = \lambda_0\cos(kx) + y_0$ with $y_0 = 3L/2$. The gravity is zero and we perform the simulations until time $\tau = 25$. Only the density and viscosity are varied in the next subsections. We will start first by treating the case of equal density and viscosity in §4.4.3.1 then the case of air-water capillary wave in §4.4.3.2.

The simulated interface location $y_S(\tau)$ at $x = 0$ is determined either from the PLIC reconstruction in the case of the volume-of-fluid methods, or in the case of the level-set methods by subdividing the mixed cell into 1000×1000 sub-cells. The interface position $y_A(\tau) = \lambda(\tau)\cos(\omega_0\tau) + y_0$ is then determined from the analytical solution (4.42) at the same discrete time levels in order to evaluate the RMS error, defined as

$$E_{RMS} = \frac{1}{\lambda_0}\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_S(\tau_i) - y_A(\tau_i))^2}$$

where N is the number of time steps.

4.4.3.1 Fluids with equal densities and viscosities

The fluid properties are set to the same values as in Popinet (2009): $\rho_1 = \rho_2 = 1$, $\sigma = 1$ and $\nu_1 = \nu_2 = 0.0182571749236$, which gives a corresponding Laplace number of La$\equiv \sigma L/\rho\nu^2 = 3000$. Fig. 4.11 shows the relative height, $y - y_0$, of the interface at $x = 0$ as a function of the non-dimensional time, τ, for different grid sizes. The analytical solution given by Eq. (4.42) is represented by the blue symbols.

It is found that the VOF-HF1 and VOF-HF2 methods give similar results, therefore only the result of the first is shown in Fig. 4.11. Due to the small amplitude of the wave, the

interface remains nearly flat and the curvature is rather small. The Height Function method is particularly well suited to this problem, since the orientation of the interface is always in the positive y-direction. This explains also the similar results obtained with the two versions HF1 and HF2 of the Height Function method.

The VOF-HF1 and MOF-HF1 methods are very accurate and recover the frequency even on the coarse grid of 8×24. The VOF-WYC method shows an increase in the period on the coarse grid, which is most probably due to the cell-centered discretization of the Navier-Stokes equations. A similar behavior is observed with the free CFD software Gerris (Popinet), which also uses a cell-centered grid setting.

The increase of the oscillation period is also observed for the LS methods on the grids 8×24 and 16×48. With these methods, the amplitude is also overestimated on these coarse grids.

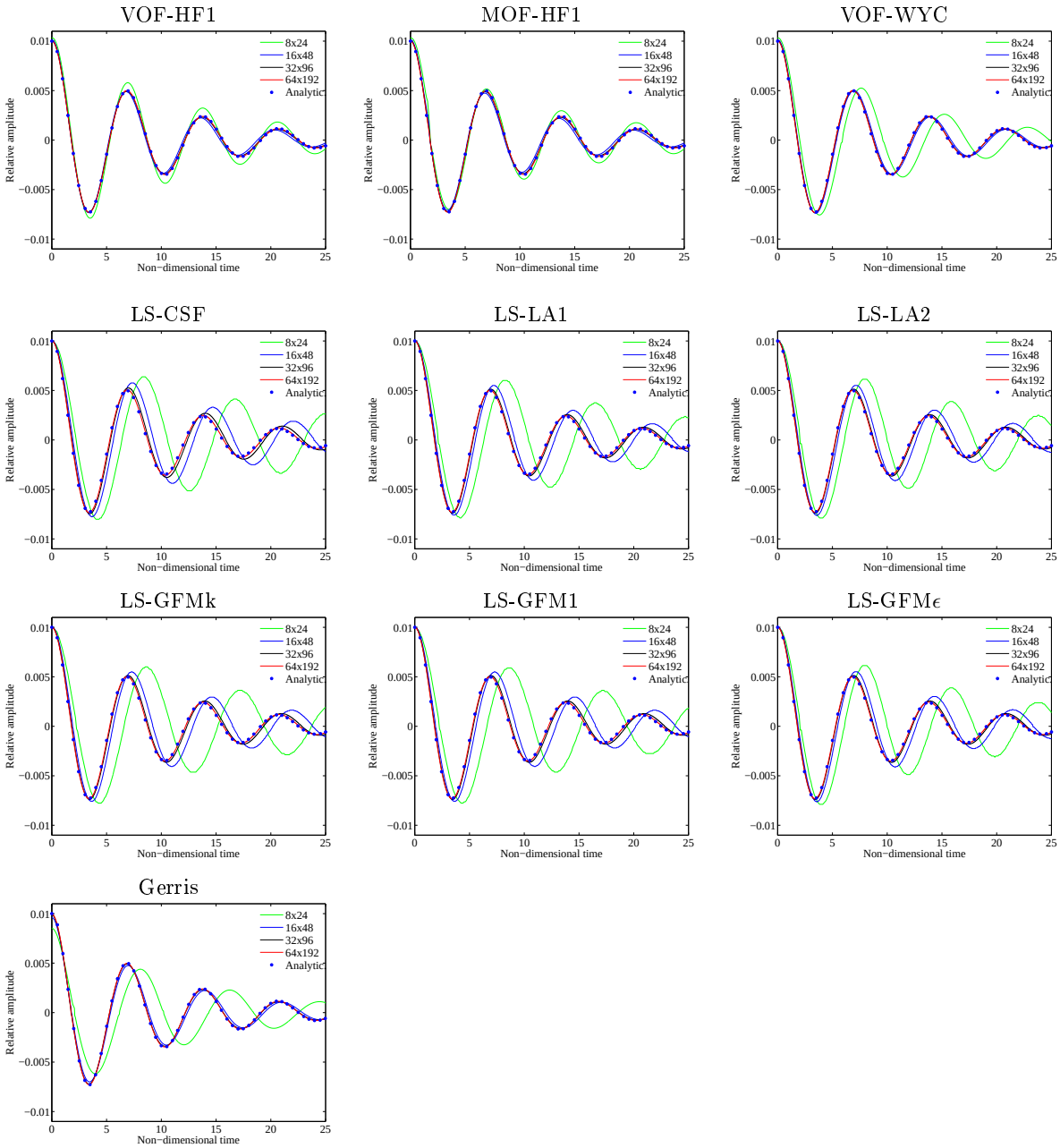

Figure 4.11: Capillary standing wave for a uniform density and viscosity: temporal evolution of the interface height at $x = 0$ for CFL=CFL$_\sigma = 0.5$.

The RMS error E_{RMS} between the analytical solution and the simulation is plotted on Fig. 4.12[3]. The VOF methods (in blue) are clearly more accurate than the LS methods. However, the LS methods keep a constant second-order convergence, while the VOF-HF1 and MOF-HF1 methods fail to do so on the finest grids. The best results are obtained with the free CFD software Gerris (Popinet).

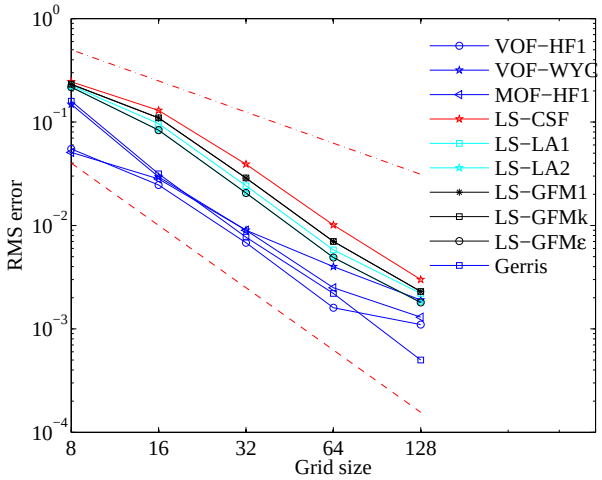

Figure 4.12: Capillary standing wave for fluids with uniform density and viscosity: RMS error of the temporal evolution of the interface height as a function of grid size. $\mathrm{CFL}_\sigma = 0.5$. The dashed lines denote 1st- and 2nd-order convergence. The level-set is reinitialized after each advection step.

Since the problem is surface-tension driven, the time step is limited by the CFL_σ number. We will check here the influence of CFL_σ on the accuracy of the simulations. The RMS error E_{RMS} is plotted on Fig. 4.13 for different grid sizes and CFL_σ values for two methods. The influence of CFL_σ is in general weak. We see that the RMS error slightly decreases with the CFL_σ for the VOF-HF1 method. For the level-set method, the RMS slightly increases for decreasing CFL_σ number, which is most probably due to the reinitialization of the level-set which is performed more often.

4.4.3.2 Air-water capillary wave

We now set the density ratio equal to 850 and the dynamic viscosity ratio to 55.72, as in Popinet (2009). The analytical solution data are taken from the Gerris website (Popinet). We use cubic interpolation to interpolate our data on the same time steps for which the analytical solution is available, in order to compute the RMS error. The temporal evolution of the interface is not shown here since it is almost similar to those of the first test case. We only plot the RMS error on Fig. 4.14[4]. The results of the Gerris code are obtained using the smoothed density (vertex-averaged smoothing kernel of Eq. (4.8)), while no smoothing kernel is used here for the VOF methods.

We see that on the coarse grid of 8×24, the LS-CSF method provides the smallest error, but for finer grids the VOF methods give the best results. The LS-GFMk and LS-GFMϵ methods are only first-order accurate, while the LS-GFM1 method shows a second-order convergence on fine grids. The LS-LA methods give good results on coarse grids, but are less accurate on the finer grids.

[3] The numerical values of the errors are presented in Appendix B on Table B.1.
[4] The numerical values of the errors are presented in Appendix B on Table B.2.

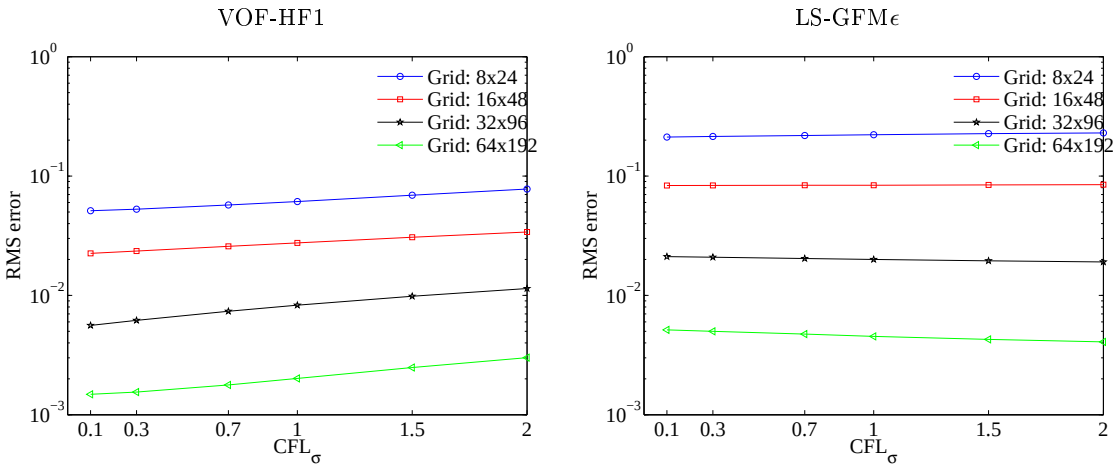

Figure 4.13: Capillary standing wave for a uniform density and viscosity: influence of the time step on the RMS error E_{RMS}. The level-set is reinitialized after each advection step.

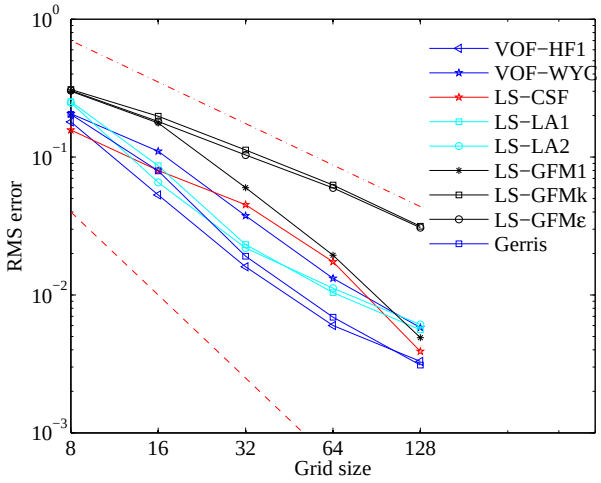

Figure 4.14: Capillary standing wave for water-air: RMS error on the interface height evolution as a function of grid size. $\text{CFL}_\sigma = 0.5$. The red-dashed lines denote 1st- and 2nd-order convergence. The level-set is reinitialized after each advection step.

Effect of the smoothing kernel

The influence of the smoothing kernel on the VOF-HF1 method is summarized on Table 4.8[5]. The results are clearly improved, in particular on the coarse grids. The Vertex-Avg smoothing kernel seems to give the best results.

[5]The data for the other VOF methods are reported in Appendix B on Table B.3

Table 4.8: Capillary standing wave for water-air: influence of the smoothing kernel on the RMS error with the VOF-HF1 method on different grid sizes. CFL=CFL$_\sigma$=0.5.

method	kernel /Grid	8	16	32	64	128
VOF-HF1	none	0.1798	0.0531	0.0160	0.0060	0.0033
	K_8	0.1670	0.0414	0.0153	0.0062	0.0034
	δ_ϵ	0.1746	0.0501	0.0161	0.0062	0.0034
	Vertex-Avg	0.1496	0.0198	0.0134	0.0062	0.0034

4.4.4 Rayleigh-Taylor instability

The test case of the Rayleigh-Taylor instability is similar to the previous one (capillary wave) but with gravity forces included. It consists of considering a sinusoidal perturbation on a horizontal interface separating two immiscible fluids, the lighter fluid is placed beneath the heavier one. Under the influence of gravity forces, the heavy fluid falls and forms the instability. The interface initially separating the two fluids is given by:

$$y = \lambda_0 \cos(kx) + y_0 \tag{4.43}$$

where λ_0 and $k = 2\pi$ are the initial amplitude and wavenumber of the perturbation, respectively.

Many variants of this test case can be found in literature. The main difference is in the physical phenomena that are taken into account (surface tension, viscosity). A brief overview is given here.

- Gómez et al. (2005) studied the instability with viscous fluids, both with surface tension forces (simulations up to the time $t = 1.3$) and without (until $t = 0.9$).
- Desjardins and Pitsch (2009) have studied the instability with viscous fluids and surface tension until $t = 1.2$ on grids going from 32×128 to 512×2048. The authors computed the penetration depth and compared it to the results on the finest grid of 512×2048.
- Brackbill et al. (1992) and Gerlach et al. (2006) have studied the instability with inviscid fluids and surface tension, and compared their results to the analytical solution available in this case.

In the current work, we will start first by studying the case of viscous fluids without surface tension in §4.4.4.1. Next, we add the surface tension in §4.4.4.2. Finally, the case of inviscid fluids with surface tension is treated in §4.4.4.3.

4.4.4.1 Viscous fluids without surface tension

The domain is $[0, 1] \times [0, 4]$. The heavy (light) fluid has a density $\rho_1 = 1.225$ ($\rho_2 = 0.1694$) and a viscosity $\mu_1 = 0.00313$ ($\mu_2 = \mu_1$) (Gómez et al., 2005). The initial perturbation is given by Eq. (4.43) with an amplitude of $\lambda_0 = 0.05$ and $y_0 = 2$. The acceleration due to gravity is $g = 1$.

Since there is no surface tension, the LS-GFM1 and LS-GFMϵ methods are identical in this particular case. Similar for the LS-LA1 and LS-LA2 methods, and the VOF-HF1 and VOF-HF2 methods. In addition, results for the VOF-WY and VOF-HF1 methods are indistinguishable and only the latter is shown.

The interface is plotted on Fig. 4.15 for the times $t = 0.6$ and 1.0 on a 64×256 grid. The interface on the grid 128×512 is plotted at the final time on Fig. 4.16. The results obtained with

the software Gerris (Popinet) with a maximum time step[6] of $\Delta t_{\max} = 5 \times 10^{-4}$ are also added to this comparison. At times $t = 0.6$, all methods give similar results except the LS-LA method which has a deformed interface. At the final time, the VOF-WYC method and Gerris, both with a cell-centered grid setting, show very similar results. The interface obtained with the other methods are almost similar, except for the LS-LA method for which there is no break-off and the neck is thinner. If we compare the results on the fine grid (128×512) in Fig. 4.16, we remark that all methods give a similar interface shape, except the LS-LA method which seems less accurate for this case. On the other hand, Gerris and the VOF-WYC method already presented a rather accurate prediction on the coarse grid.

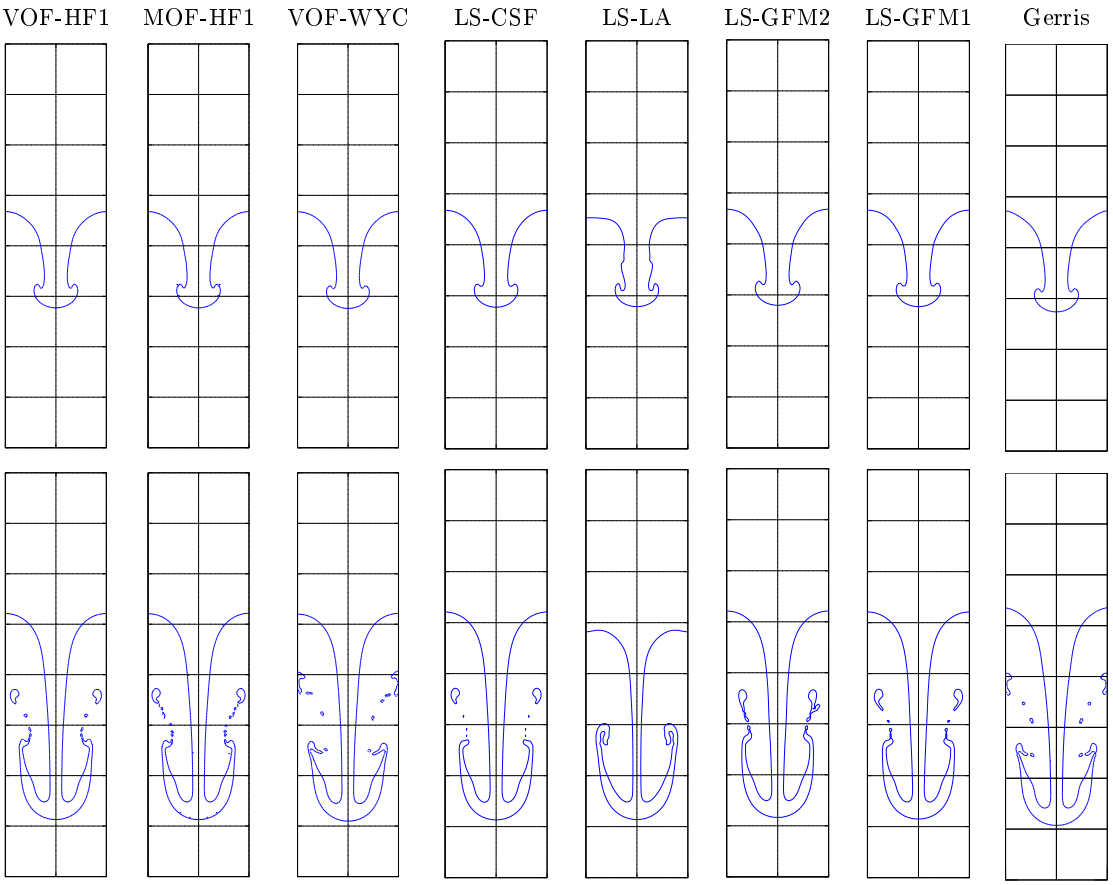

Figure 4.15: Rayleigh-Taylor instability with viscous fluids without surface tension: interface at $t = 0.6$ (upper row) and $t = 1$ (lower row) on a 64×256 grid with CFL$= 0.2$. No smoothing kernel is used with the VOF methods. The level-set field is reinitialized each time step.

4.4.4.2 Viscous fluids with surface tension

When the surface tension is added to the previous case, the penetration depth will be smaller since the surface tension tends to retain the interface. We set $\sigma = 0.1337$ as in Gómez et al. (2005) while $\text{CFL}_\sigma = 0.5$ and CFL$= 0.2$. For Gerris, we use $\Delta t_{\max} = 5 \times 10^{-4}$.

The interface is plotted on the 64×256 grid at times $t = 1.1$ and 1.3 on Fig. 4.17 for all methods except the VOF-HF2 method which fails to calculate up to the final time. The interface at the final time $t = 1.3$ on the 128×512 grid is plotted on Fig. 4.18.

[6] Note that a larger time step of $\Delta t_{\max} = 5 \times 10^{-3}$ is used in Popinet, which resulted however in a too large penetration depth.

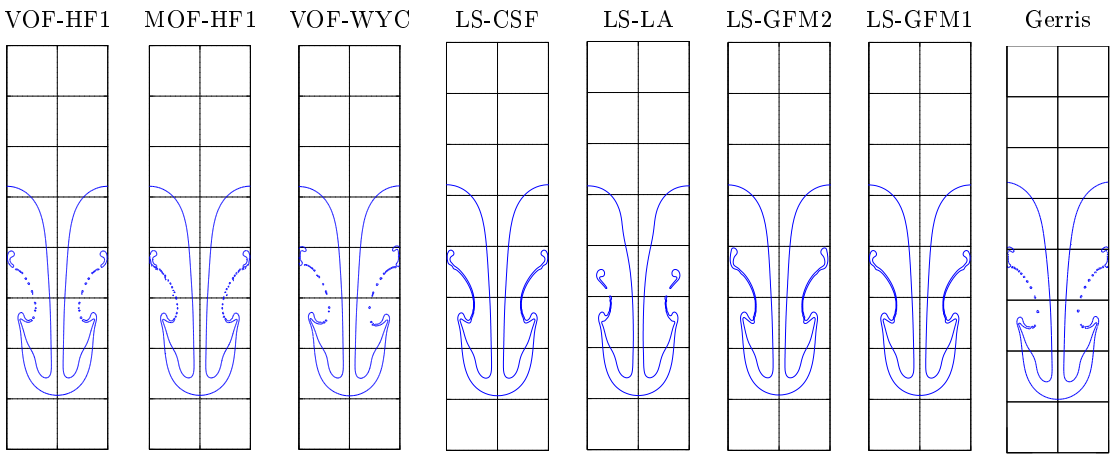

Figure 4.16: Rayleigh-Taylor instability with viscous fluids without surface tension: interface at $t = 1$ on a 128×512 grid with CFL=0.2. No smoothing kernel is used with the VOF methods. The level-set field is reinitialized each time step.

On the fine grid, all methods again give similar results, although the spike is deeper with Gerris than with our VOF and LS methods. On the coarse grids, the interface structures are much thinner, with break-off for some methods. In the case without surface tension, the LS-LA method does not provide good results. However, for the present case with surface tension, the two methods LS-LA1 and LS-LA2 perform very well. The LS-LA1 gives results comparable to the other methods, while the LS-LA2 method produces similar results on both the fine and coarse grids. The results provided by the LS-GFM methods are slightly different. They resemble more to those obtained at the coarse grids, which might indicate poorer accuracy.

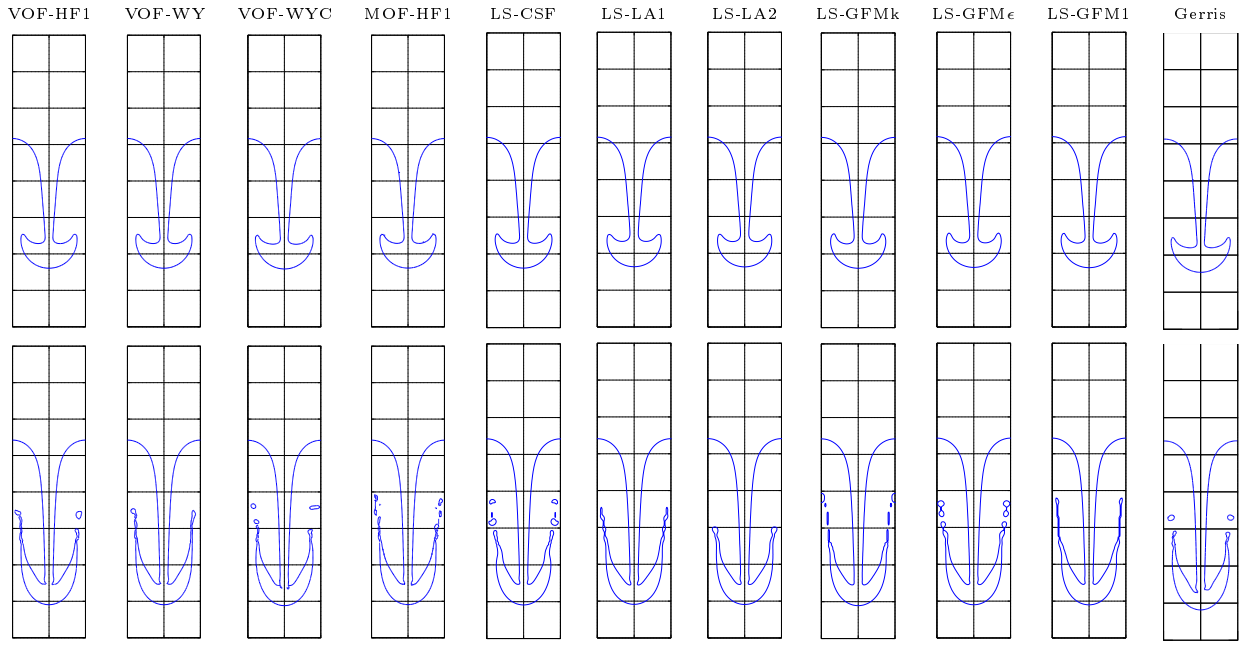

Figure 4.17: Rayleigh-Taylor instability with surface tension: interface at $t = 1.1$ (upper row) and $t = 1.3$ (lower row) on a 64×256 grid with CFL=0.2 and $\text{CFL}_\sigma = 1.0$. No smoothing kernel is used with the VOF methods. The level-set field is reinitialized each time step.

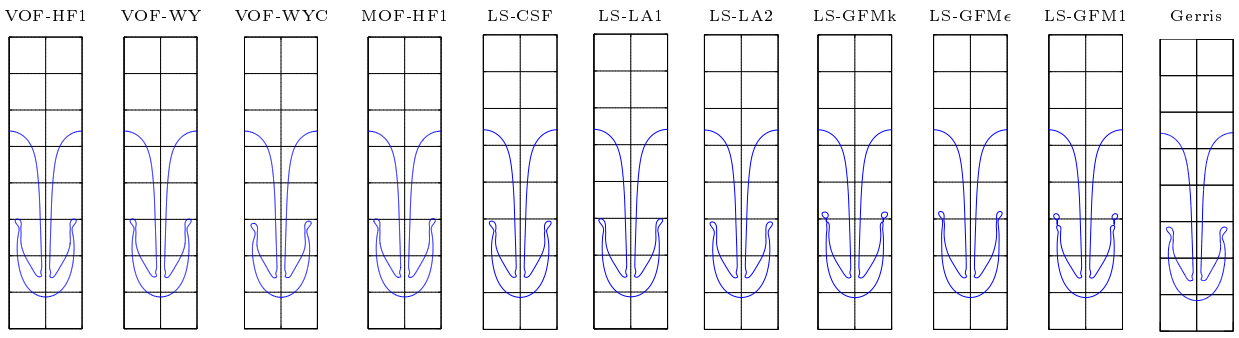

Figure 4.18: Rayleigh-Taylor instability with surface tension: interface at $t = 1.3$ on a 128×512 grid with CFL=0.2 and $\text{CFL}_\sigma = 1.0$. No smoothing kernel is used with the VOF methods. The level-set field is reinitialized each time step.

4.4.4.3 Inviscid fluids with surface tension

The Rayleigh-Taylor instability simulated in the previous sections concerns the case of viscous fluids with or without surface tension effects. The comparison between the different methods was based on the shape of the interface at a given time of the simulation. In the case of inviscid fluids with surface tension, the spike grows exponentially as $\exp(nt)$ where n is the growth rate. The two physical phenomena governing the growth of the instability are the acceleration due to gravity, g, and the surface tension, σ, which tends to delay the growth of the spike. An analytical expression of the growth rate, n, for this specific problem is provided by Bellman and Pennington (1953):

$$n^2 = kg\text{A}\left[1 - \frac{k^2\sigma}{(\rho_1 - \rho_2)g}\right] \quad \text{where} \quad \text{A} = \frac{\rho_1 - \rho_2}{\rho_1 + \rho_2} \tag{4.44}$$

with A the Atwood number, k the wavenumber and ρ_1 and ρ_2 the densities of the heavy and light fluids, respectively.

The critical surface tension, σ_C, is determined from Eq. (4.44) by setting $n = 0$. Daly (1969) has introduced a dimensionless parameter, which is the ratio of the surface tension coefficient to the critical value:

$$\xi = \frac{\sigma}{\sigma_c} = \frac{\sigma k^3}{(\rho_1 - \rho_2)g} \tag{4.45}$$

This dimensionless parameter is a measure of the importance of surface tension forces to gravity forces. If it is close to zero, the gravity forces are much greater than the surface tension forces. If ξ is close to unity, the surface tension dominates the growth process of the spike.

In our simulations, the depth of the spike, λ, is determined by finding the location of the interface at the half-width of the domain.

Brackbill et al. (1992) and Gerlach et al. (2006) have compared the growth rate to the analytical solution given by Eq. (4.44). In this work, we adopt the same parameters as Gerlach et al. (2006). The domain is rectangular, of width equal to 2π and wavenumber $k = 1$. The domain height is three times its width with slip lateral boundaries and no-slip top and bottom boundaries. The Atwood number is chosen equal to $\text{A} = 0.6$ and ξ is varying in the range $[0.05, 0.9]$. We define a dimensionless time $\tau = t/\sqrt{kg\text{A}}$. The initial perturbation (Eq. (4.43)) has an amplitude of $\lambda_0 = 0.001$. We set CFL= 0.5 and CFL_σ=1. Furthermore, we take $g = 1$ and $\rho_1 = 1$. The simulations are performed until time $\tau = 10$.

The temporal evolution of the penetration depth is plotted on Fig. 4.19 for different values

of ξ. The VOF-HF2 method did not reach the final time and is not presented. The VOF-HF1, MOF-HF1 and VOF-WY methods provide similar results. Therefore, only the former is shown in this section.

The perturbation grows slower with the VOF-WYC method for $\xi = 0.9$ while the perturbation vanishes with the VOF-HF1 and MOF-HF1 methods towards the end of the simulations.

With the LS-LA1 and LS-LA2 methods, the perturbation starts to oscillate with constant amplitude, even for larger grid sizes. These two methods are therefore not shown here.

When we refine the grid, LS-GFM1 and LS-GFMk methods give very similar results (see Fig. 4.20) even though the surface tension is computed differently in both methods. The anomalies appear only when the grid is refined and ξ is close to unity (high surface tension). In these conditions, the value of the CFL_σ number has little influence on the evolution of the penetration depth. The parameter that has a much more influence is the amplitude of the initial perturbation, λ_0, as we can see it on Fig. 4.21.

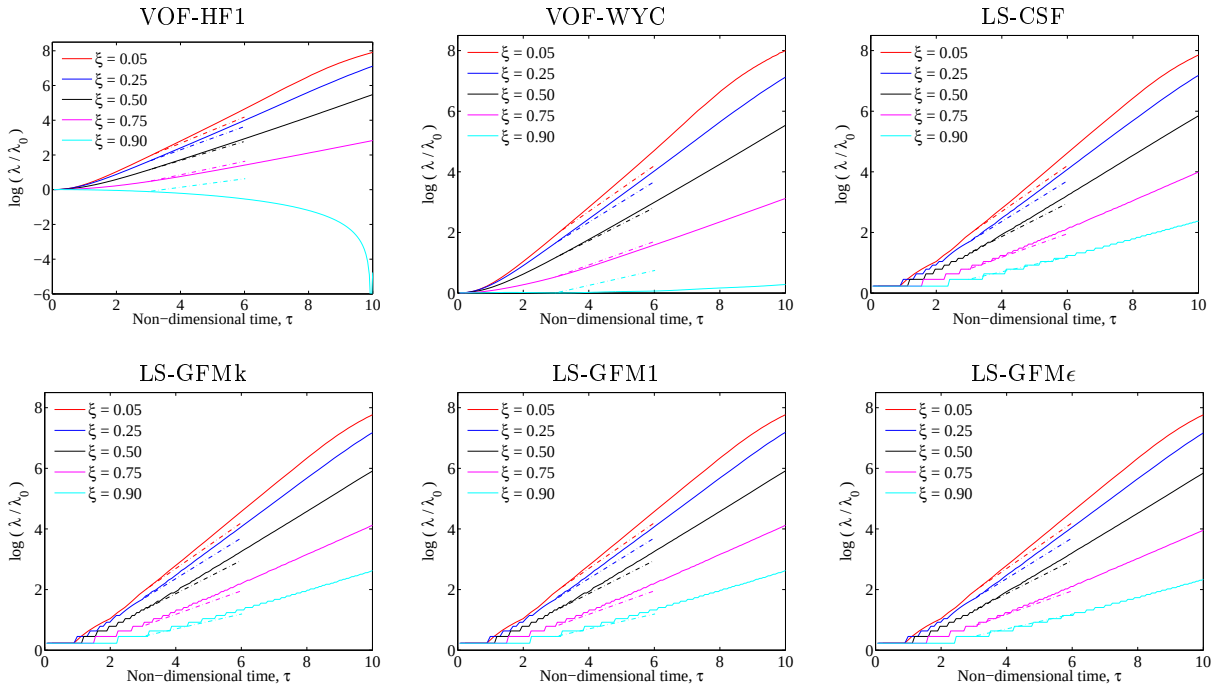

Figure 4.19: Rayleigh-Taylor instability with inviscid fluids: evolution of the penetration depth (solid lines). The analytical slopes of the growth are also plotted (dashed lines). Grid size 20×60. No smoothing kernel is used with the VOF methods. The level-set field is reinitialized each time step.

The interface at the final time $\tau = 10$ on the finest grid is plotted on Fig. 4.22 for different values of ξ. Little differences appear between the different methods, although the interface shape with the VOF-WYC method is really different from the others for $\xi = 0.9$.

The growth rate, determined between $\tau = 3$ and $\tau = 6$, is plotted on Fig. 4.23. However, these results should be compared with precaution. Indeed, it is sometimes difficult to find a linear region in the logarithm of the amplitude as we can see on Figs. 4.19 and 4.20. The growth rate depends on the initial amplitude of the perturbation, λ_0.

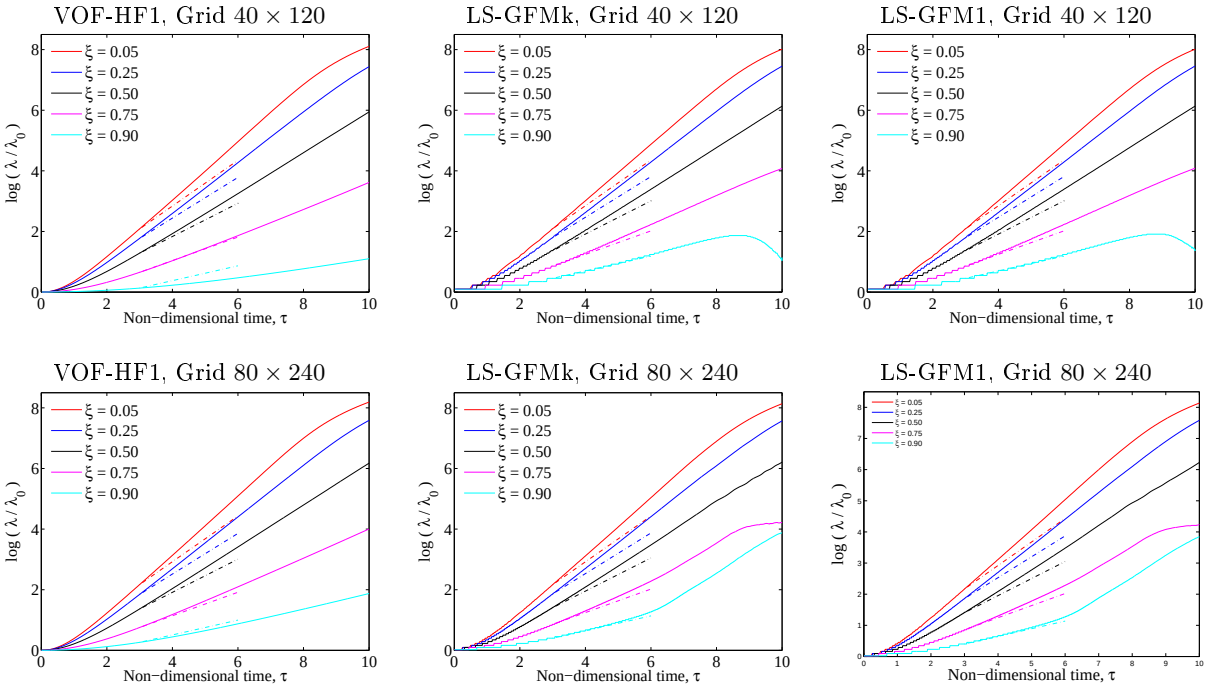

Figure 4.20: Rayleigh-Taylor instability with inviscid fluids: evolution of the penetration depth (solid lines) as function of ξ. The analytical slopes of the growth are also plotted (dashed lines). No smoothing kernel is used with the VOF methods. The level-set field is reinitialized each time step.

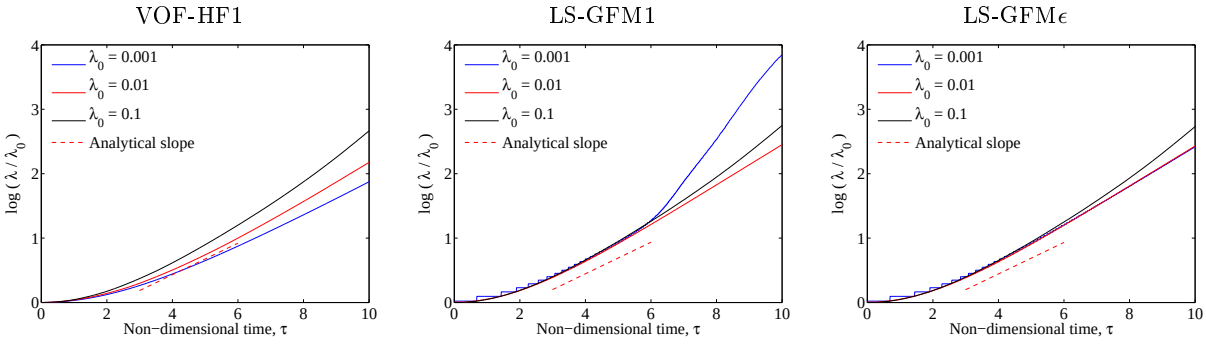

Figure 4.21: Rayleigh-Taylor instability with inviscid fluids: evolution of the penetration depth (solid lines) with three different methods as function of the initial perturbation amplitude λ_0. The analytical slope of the growth is plotted (dashed line). $\xi = 0.9$ and the grid is 80×240. No smoothing kernel is used with the VOF methods. The level-set field is reinitialized each time step.

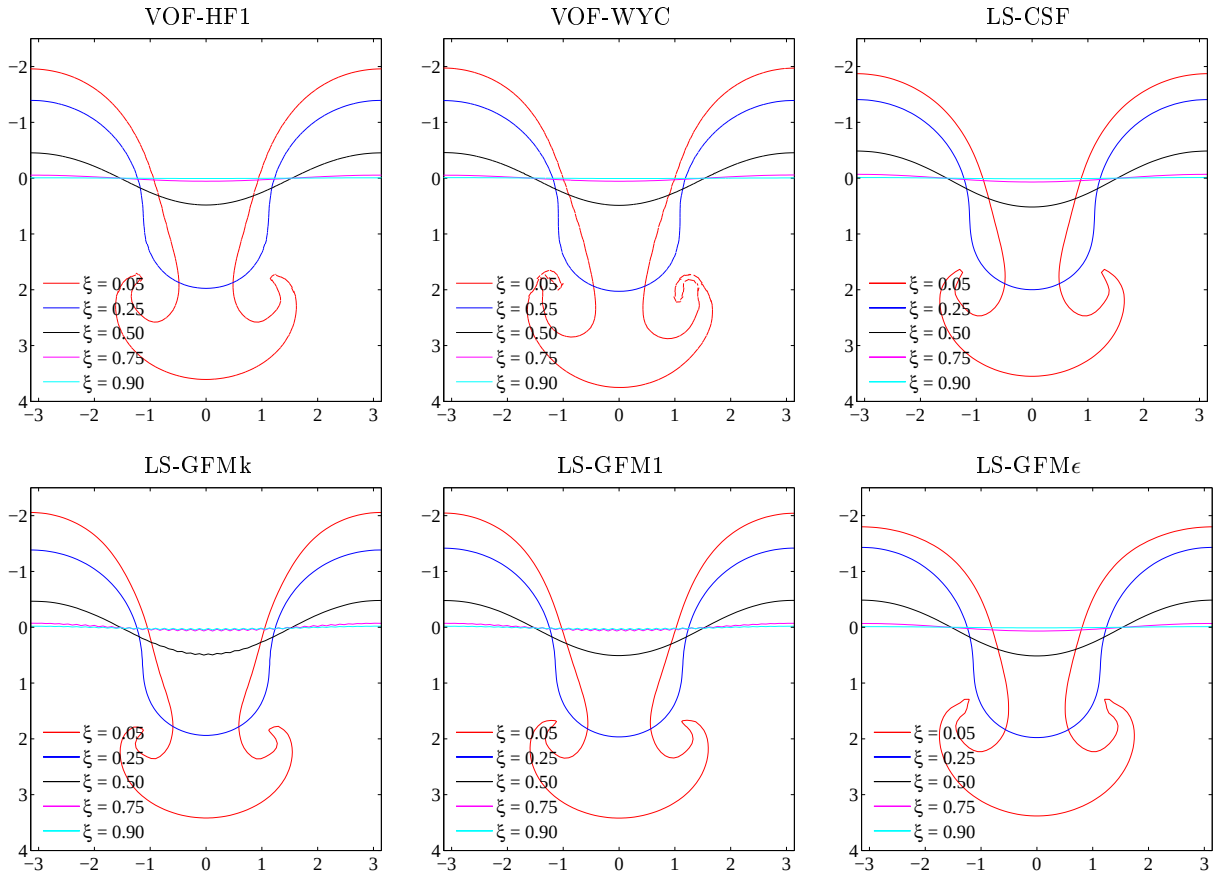

Figure 4.22: Rayleigh-Taylor instability with inviscid fluids: interface at time $\tau = 10$ obtained on a 80×240 grid for different values of ξ. $\lambda_0 = 0.001$, CFL$= 0.5$ and CFL$_\sigma = 1.0$. No smoothing kernel is used with the VOF methods. The level-set field is reinitialized each time step.

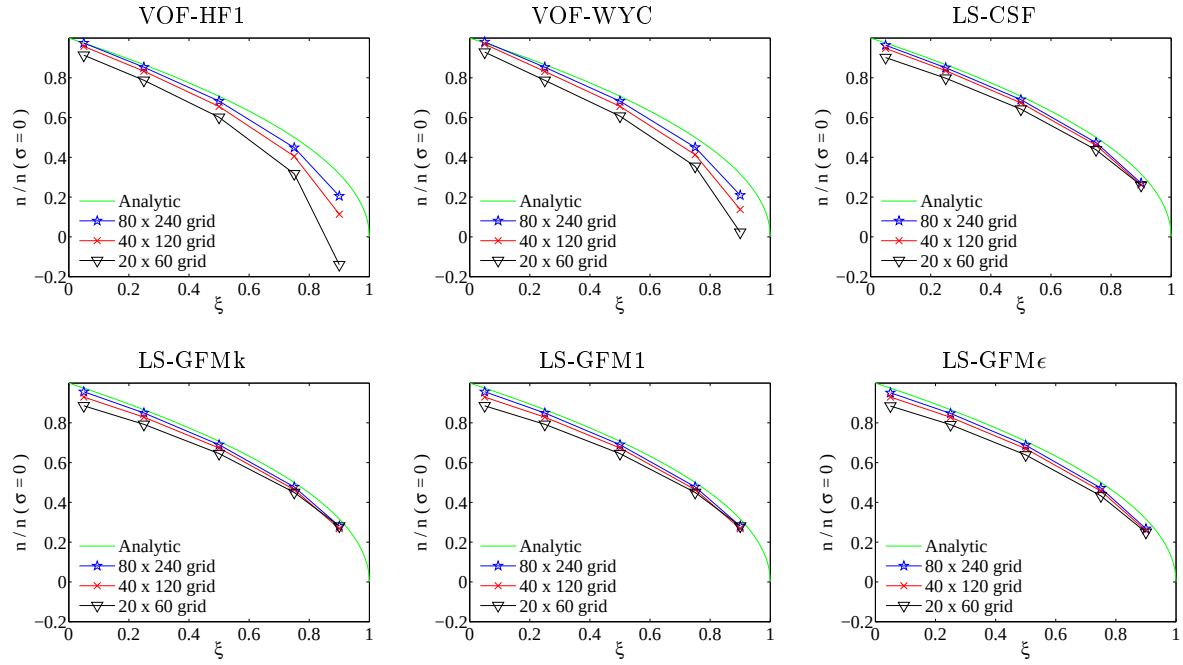

Figure 4.23: Rayleigh-Taylor instability with inviscid fluids. Growth rate n as a function of ξ on different grid sizes. $\lambda_0 = 0.001$, CFL$= 0.5$ and CFL$_\sigma = 1.0$. No smoothing kernel is used with the VOF methods. The level-set field is reinitialized each time step.

4.4.5 Rising bubble

The rising bubble is a classical test case to validate two-phase flow solvers. It tests the overall behavior of the solver, i.e. the accuracy of the viscous terms, the surface tension, for relatively high density and viscosity ratios. For axisymmetric and fully three dimensional flows, the numerical results are often compared to experimental data - see for example Van Sint Annaland et al. (2005); Ohta et al. (2005); Hua and Lou (2007); Sussman et al. (2007); Hua et al. (2008) and Albadawi et al. (2013) - but neither experimental data nor analytical solutions exist for the two-dimensional bubbles. Hysing et al. (2009) proposed a benchmark for the simulation of two cases of a two-dimensional rising bubble. The first case employs a relatively low density and viscosity ratio, and the second one deals with high ratios. The initial configuration is depicted in Fig. 4.24.

The initially spherical bubble of diameter $D = 0.5$ is centered at $(0, 0.5)$ in a $[-0.5, 0.5] \times [0, 2]$ domain. Symmetry conditions are applied to the lateral boundaries and no-slip conditions are used for the upper and lower walls. The fluid properties for the two cases are summarized on Table 4.9 (Hysing et al., 2009; Aland and Voigt, 2012).

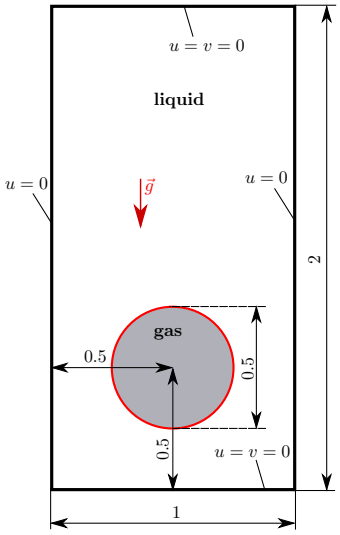

Figure 4.24: Rising bubble benchmark: computational domain, boundary conditions and initial bubble configuration.

Table 4.9: Rising bubble benchmark: fluid properties

Physical parameters	ρ_l	ρ_g	μ_l	μ_v	g	σ
Benchmark 1	1000	100	10	1	0.98	24.5
Benchmark 2	1000	1	10	0.1	0.98	1.96

The bubble is simulated with $\text{CFL}_\sigma = \text{CFL} = 0.5$. Our results are compared to those provided by the MooNMD group (Hysing et al., 2009; Featflow), we consider it as a reference solution since it is based on a Lagrangian interface method. The different benchmark quantities are the centroid, circularity and mean rising velocity of the bubble. The centroid of the bubble is given by:

$$\vec{X_c} = (X_c, Y_c) = \frac{\int_{\Omega_2} \vec{x} ds}{\int_{\Omega_2} 1 ds} \tag{4.46}$$

where Ω_2 is the bubble region. The circularity is given by:

$$c = \frac{P_a}{P_b} = \frac{\text{Perimeter of area-equivalent circle}}{\text{Perimeter of bubble}} = \frac{\pi d_a}{P_b} \tag{4.47}$$

where P_a denotes the perimeter of a circle with diameter d_a which has an area equal to that of the bubble with perimeter P_b. The circularity is equal to unity for a perfectly circular bubble, and less than unity for non-circular shapes. The mean velocity is defined by

$$\vec{U_c} = \frac{\int_{\Omega_2} \vec{u} ds}{\int_{\Omega_2} 1 ds} \tag{4.48}$$

In this section, the level-set field is reinitialized after each time step. The effect of the smoothing kernel on the volume-of-fluid methods is not investigated.

4.4.5.1 Benchmark 1

The first benchmark is characterized by a low density and viscosity ratios ($\rho_l/\rho_g = \mu_l/\mu_g = 10$) and high surface tension coefficient as shown on Table 4.9. The bubble is initially circular at rest, and the bubble ascension is simulated with the different methods until time $t = 3$. The bubble shape at this particular time is plotted on Fig. 4.25 for grid sizes ranging from 32×64 to 256×512. The bubble does not undergo major topological changes due to the relatively low density and viscosity ratios and high surface tension. In fact, in this test case, the shape of the interface is surface tension controlled.

The comparison with the reference solution (Featflow) is excellent, in particular for the VOF methods. For these methods, the result is almost independent from the grid size. The VOF methods are more accurate than the LS methods which seem to suffer from mass loss on coarse grids. The MOF method is not shown since it gives very similar results as that obtained with the VOF-HF1 and VOF-WY methods. The results for the VOF-HF2 method on the grid 32×64 is also not shown, since it does not reach the end of the simulations due to a problem in the computation of the curvature. The LS-CSF and LS-GFMϵ methods behave slightly better than the other level-set methods, which is probably due to the identical manner of computing curvature. The relative mass errors are, however, of the same magnitude for all LS methods[7].

The instantaneous mean rising velocity is plotted on Fig. 4.26 for three methods. Similar profiles are obtained with the other methods. The reference solution (red stars) is also reported on this figure for the sake of comparison. Initially, the bubble is stagnant with zero rising velocity, then under the influence of buoyancy forces, it starts to move upward until it reaches a steady-state velocity. The rising velocity reaches first a maximum value, then decreases towards the steady-state velocity. The correct velocity profile is recovered with all methods. Slight differences are observed only on the coarsest grid. For this grid, the VOF-WYC method provides the most accurate solution.

Values of the maximum velocity and the time of occurrence are summarized on Table B.4. The L2-norm of the error between the computed mean velocity profile and the one provided by the reference solution is shown on Fig. 4.28. For the LS-GFMk and LS-GFMϵ methods, the error is very high on the fine grid due to the convergence of the maximum velocity to a value different from that of the reference solution. It confirms that the VOF methods are more precise than the LS methods for this particular case.

The bubble circularity is plotted on Fig. 4.27[8]. As with the mean rising velocity, the cir-

[7] Numerical values are available on Table B.4

[8] Values of the minimum circularity and the time of occurrence are reported on Table B.4.

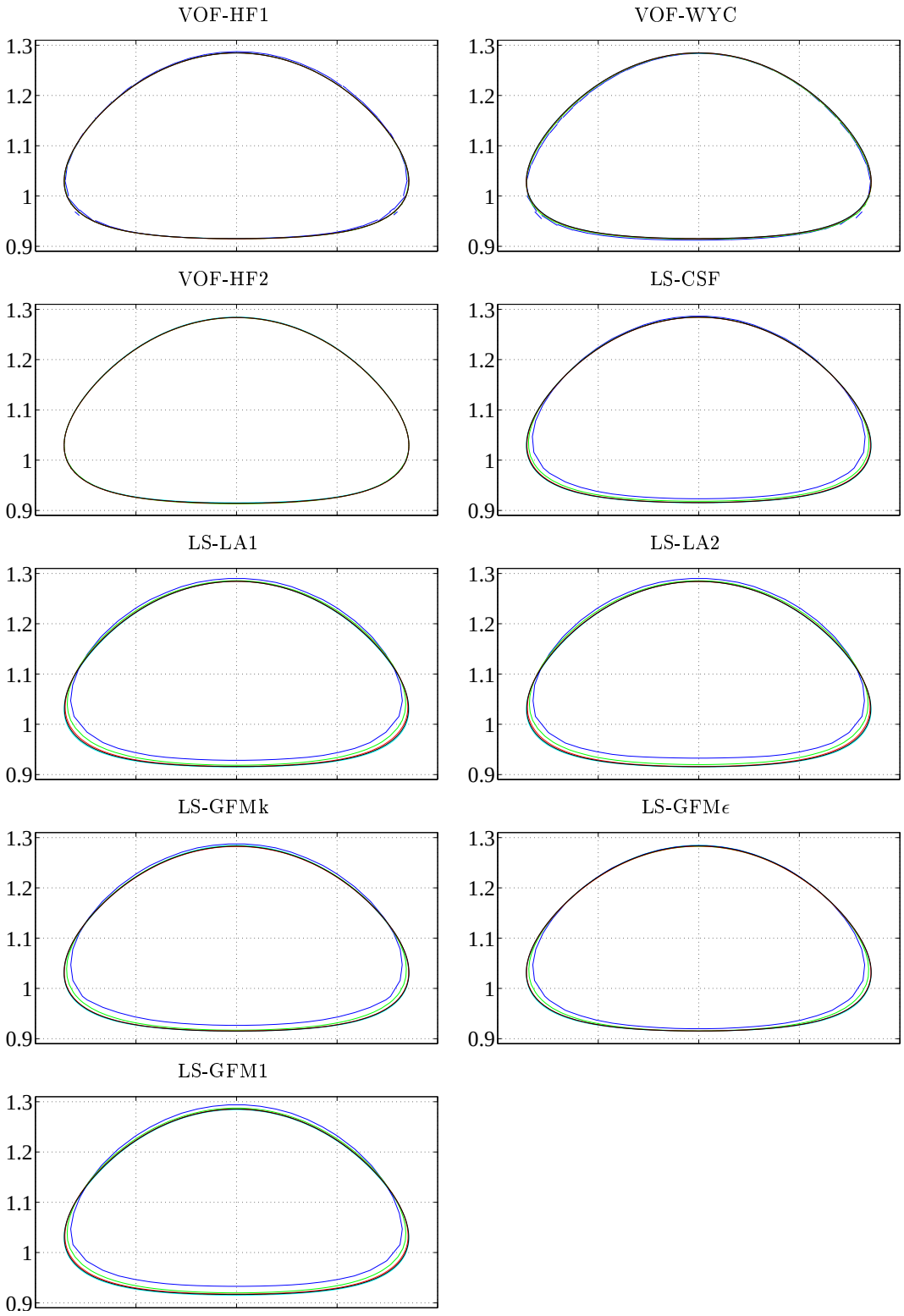

Figure 4.25: Rising bubble benchmark 1: Bubble shape at the final time $t = 3.0$ for different grid sizes: blue (32×64), green (64×128), red (128×256), black (256×512) and cyan (reference solution).

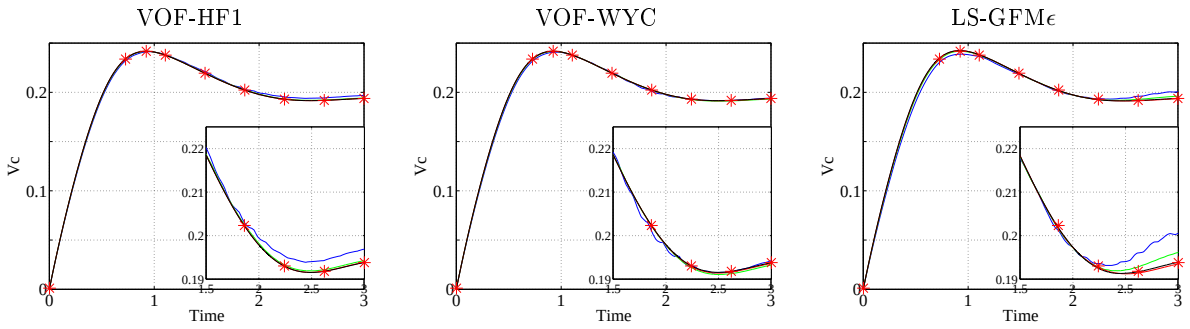

Figure 4.26: Rising bubble benchmark 1: Mean rising velocity of the bubble as a function of time on different grid sizes, and a close up for $t \in [1.5, 3]$. Colors correspond to the grid size: blue (32×64), green (64×128), red (128×256), black (256×512) and red stars(reference solution).

cularity also converges to the reference data for all methods on fine grids. The circularity is greater than unity for the VOF methods during the first time steps, because the predicted bubble perimeter (obtained with the PLIC reconstruction) is smaller than the theoretical one. We assume that the PLIC reconstruction is also responsible for the erratic behavior of the circularity profile. The L2-norm of the error of the circularity profile is plotted on Fig. 4.28. It shows that the circularity profile is better predicted with the solvers using the volume-of-fluid methods.

The temporal evolution of the bubble's centroid y-position is presented on Fig. B.1 as a function of time[9]. All the profiles obtained are similar and are superposed on that of the reference solution.

The grid convergence for the circularity, the rising velocity and centroid are plotted on Fig. 4.28. The VOF and MOF methods are clearly more accurate than the LS methods. Only the VOF-HF2 method seems to have some problems on the fine grids. Among the LS methods, the LS-CSF method shows the best results, followed by the GFM methods.

[9]The final positions of the center of mass are reported on Table B.4.

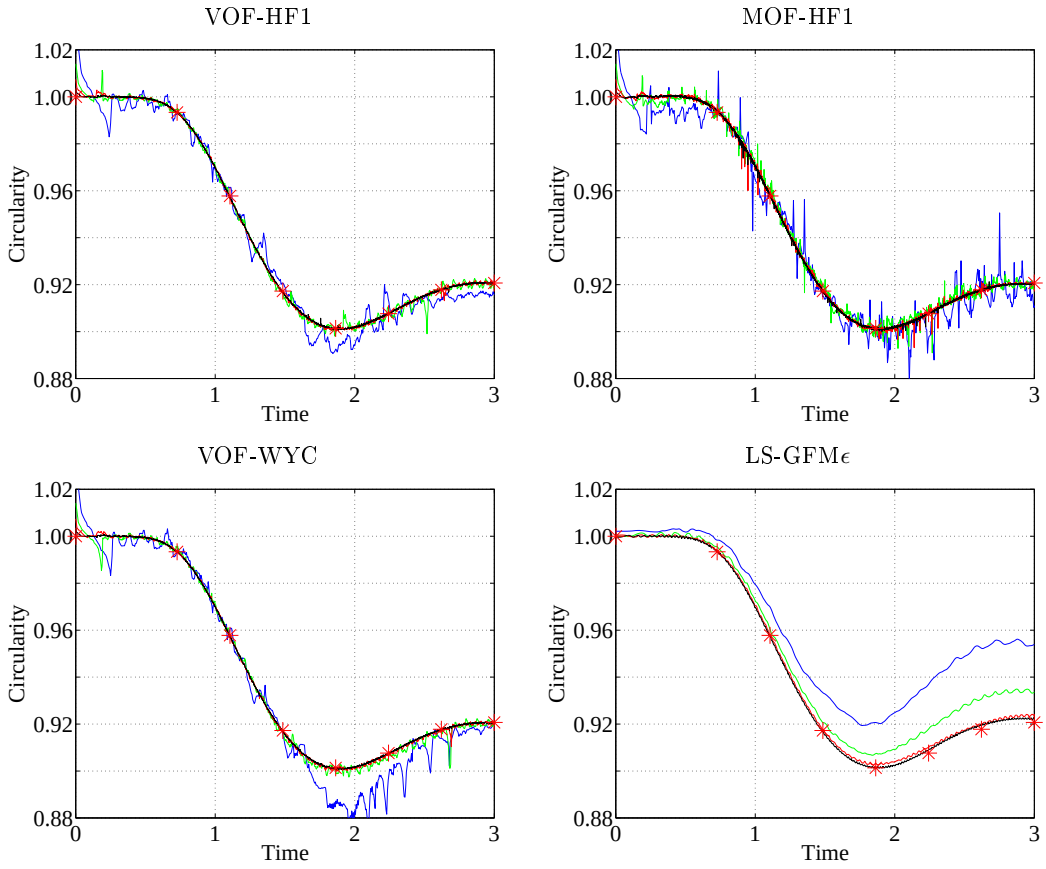

Figure 4.27: Rising bubble benchmark 1: Circularity of the bubble as a function of time on different grid sizes. Colors correspond to the grid size: blue (32×64), green (64×128), red (128×256), black (256×512) and red stars (reference solution).

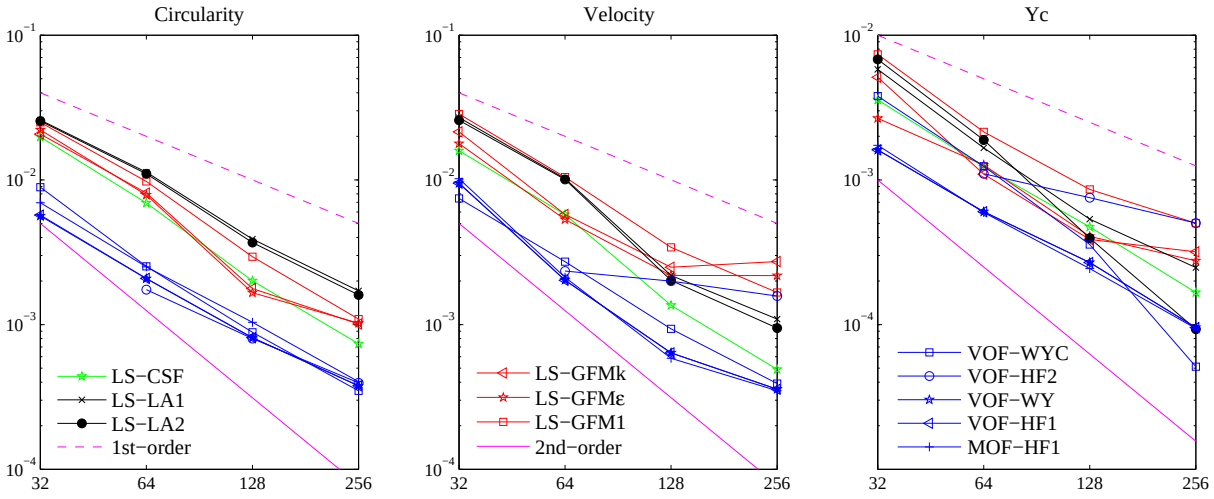

Figure 4.28: Rising bubble benchmark 1: L2-error of the temporal evolution of the different benchmark quantities as a function of the grid size in the x-direction.

4.4.5.2 Benchmark 2

The second case, characterized by a high density and viscosity ratios, is a more difficult test problem. The interface at the final time $t = 3.0$ is plotted on Fig. 4.29. The VOF-HF2 method is the only one not reaching the end of the simulation, due to a failure in the curvature computation. The reference solution (Featflow) is also shown. However, in the absence of experimental results, it is not clear whether this numerical result represents correctly the actual solution of this problem, in particular with respect to the length and shape of the thin filaments. This has to be taken into account in the comparison of the different methods.

The VOF-HF1, MOF-HF1 and VOF-WY methods give very similar results in which the thin filament breaks up in small bubbles. This is a typical behavior for VOF methods that is also evidenced in Chapter 2. On the other hand, the LS methods tend to prevent the formation of the thin filaments (probably along with some loss of mass). The LS-CSF method, which uses smoothing of the fluid properties, is very unstable and fails to compute the interface on the finest grid. Some difficulties are also observed for the LS-GFM1 method. The VOF-WYC method converges best towards the reference solution. When refining the grid, the thin filaments are well reproduced without fragmentation.

The instantaneous relative mass error for this benchmark is plotted on Fig. 4.30[10]. The mass error is much larger for the second benchmark, since the bubble undergoes more topological changes. The mass error decreases when increasing the grid size, except for the LS-CSF and LS-GFM1 methods.

The mean rising velocity is plotted on Fig. 4.31[11]. All the presented methods seem to reproduce the reference values on coarse grids until $t = 1.5$, except the LS-LA1 and LS-LA2 methods for which the velocity profile diverges rapidly from the reference velocity.

The instantaneous bubble circularity is plotted on Fig. 4.32. The circularity for the reference solution (red stars) steadily decreases in time due to the formation of the tails. The tested methods, however, produce all very different profiles. These large differences between methods were also observed in the work of Hysing et al. (2009). The VOF and MOF methods follow the reference solution rather well, due to the formation of the fragments. The LS methods prevent the formation of the filaments. Therefore, an increase of the circularity is observed after $t = 2$, except for the LS-CSF method which produces many non-physical fragments.

The center of mass of the bubble is mainly determined by the bulk shape of the bubble, which is correctly reproduced by nearly all methods. The results are therefore not shown here[12].

The L2-norm of the error between the velocity profile and that of the reference solution is plotted on Fig. 4.33. Although it is difficult to make objective conclusions, due to the absence of a true reference solution, it seems that the tested methods are at best first-order accurate. None of the methods stick out in performance, but the VOF methods have at least the benefit of showing a first-order convergence for the mean rising velocity and the center of mass, while the LS methods show a rather erratic behavior.

[10] The mass errors at the end of the simulation are reported on Table B.5.
[11] The maximum velocity and its time of occurrence are reported on Table B.5.
[12] Results are shown on Fig. B.2 and Table B.5.

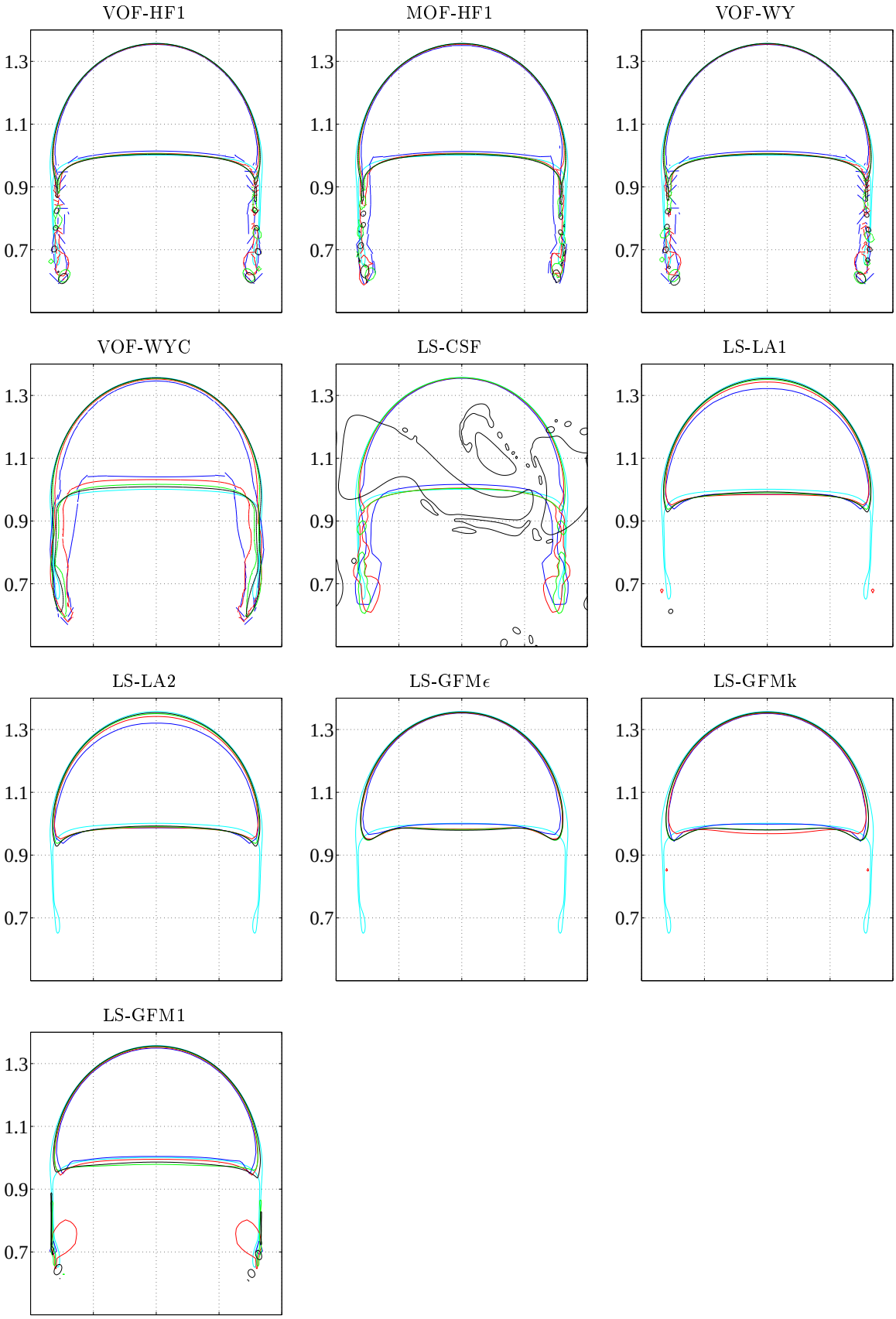

Figure 4.29: Rising bubble benchmark 2: Bubble shape at the final time $t = 3.0$. Blue (32×64), green (64×128), red (128×256), black (256×512) and cyan (reference solution).

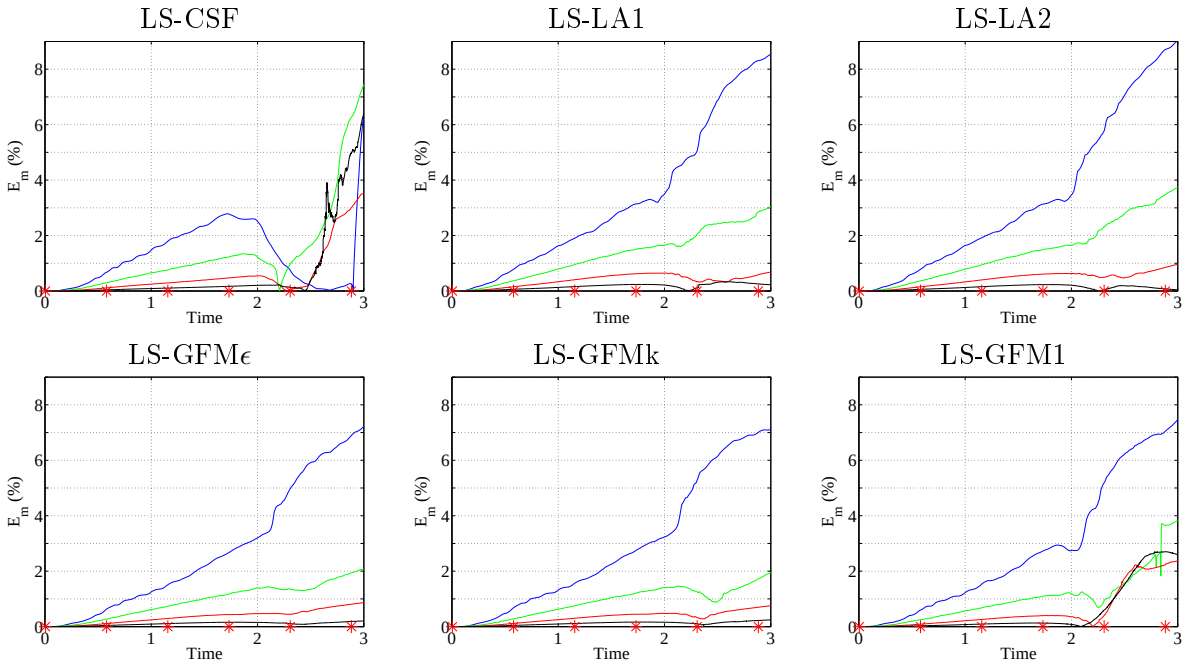

Figure 4.30: Rising bubble benchmark 2: Relative mass error as a function of time on different grid sizes. Blue (32×64), green (64×128), red (128×256), black (256×512) and green stars (reference solution).

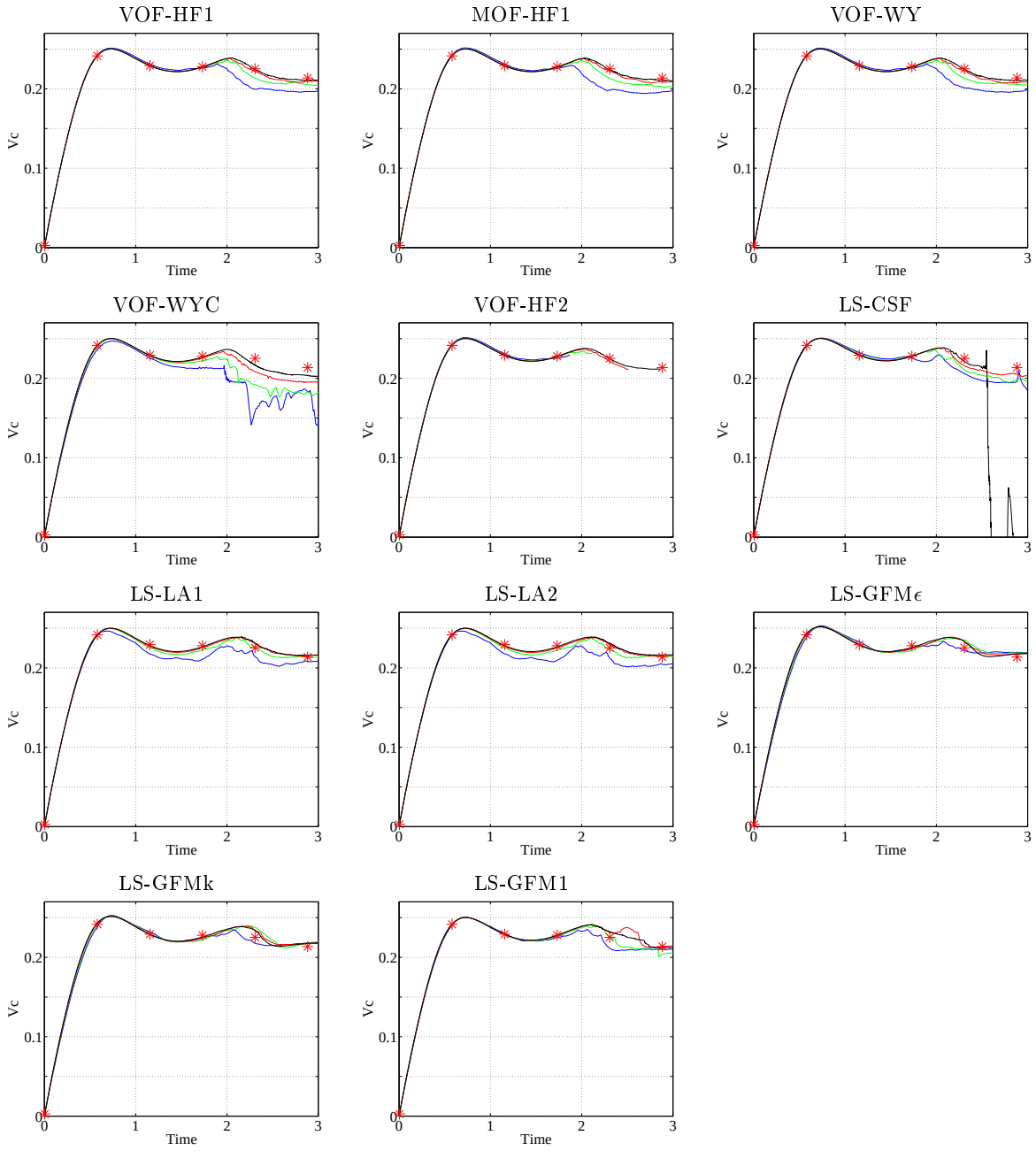

Figure 4.31: Rising bubble benchmark 2: Mean rising velocity as a function of time. Blue (32×64), green (64×128), red (128×256), black (256×512) and red stars (reference solution).

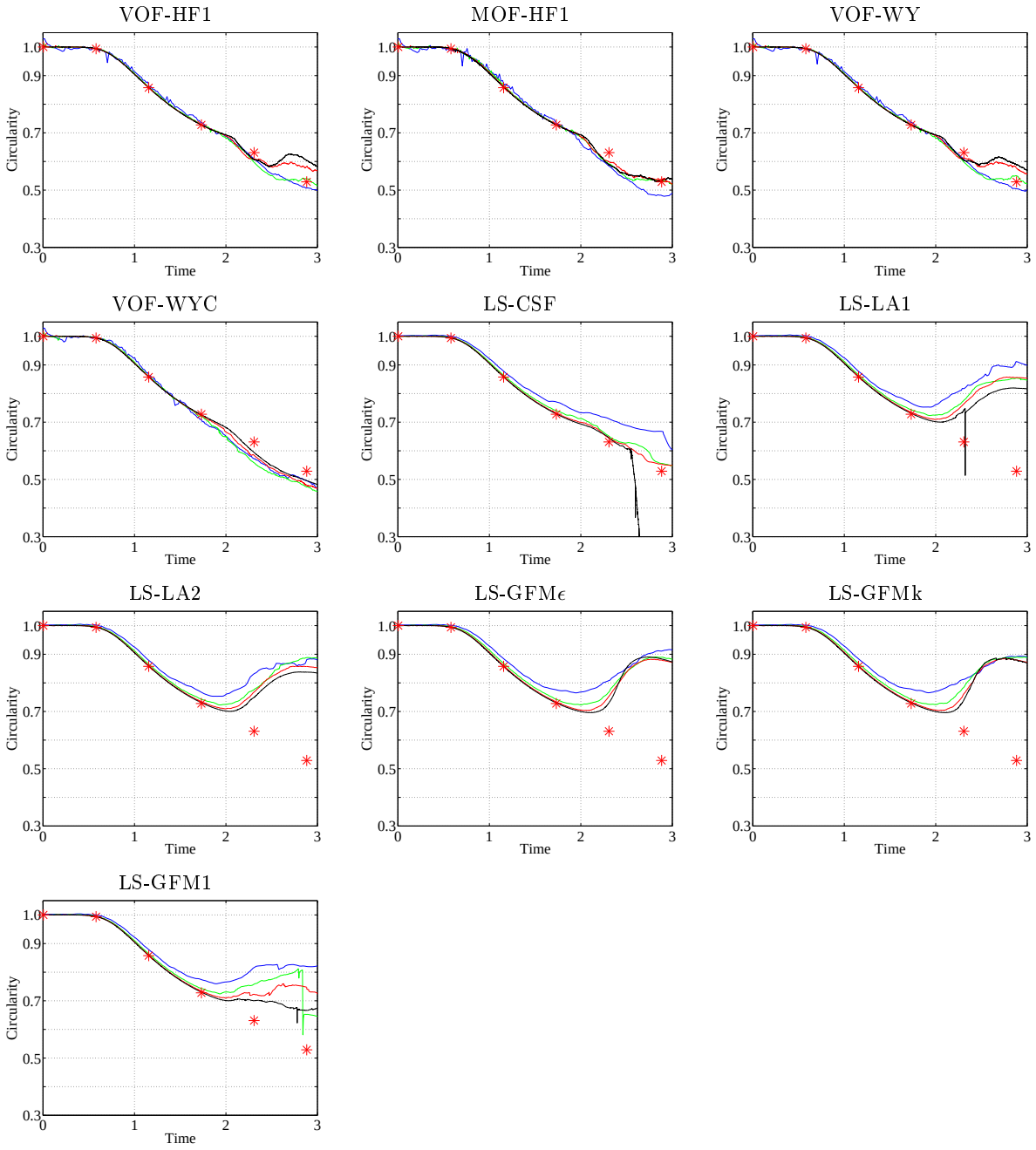

Figure 4.32: Rising bubble benchmark 2: Bubble circularity on different grid sizes. Blue (32×64), green (64×128), red (128×256), black (256×512) and red stars (reference solution).

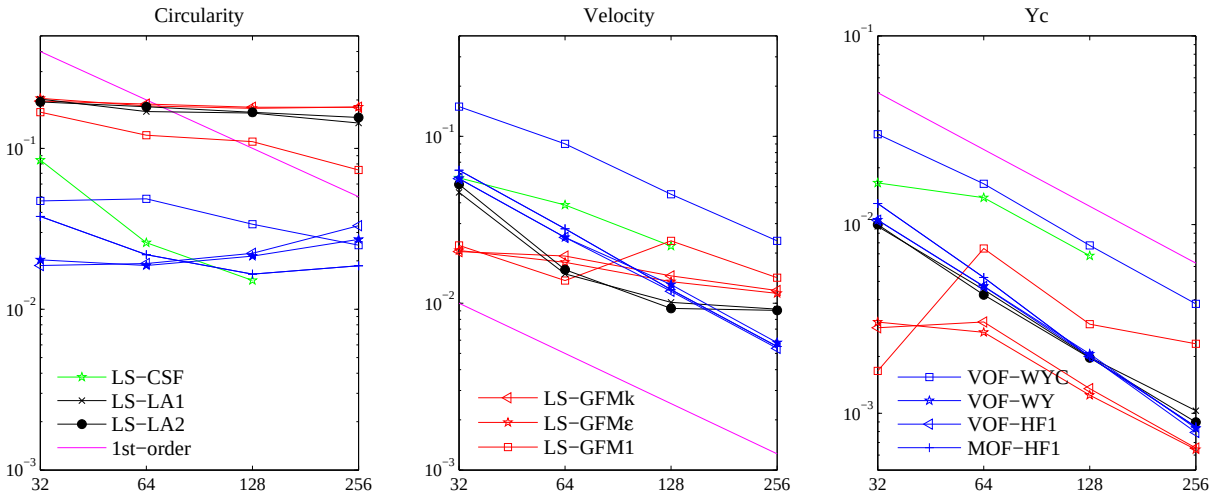

Figure 4.33: Rising bubble benchmark 2: L2-norm of the error on the evolution of the circularity, the rising velocity and the centroid position as a function of the grid size in the x-direction.

4.5 Conclusions

In this chapter, we have described different existing methods for the coupling between the Navier-Stokes solver with an interface method (mainly, the level-set and the volume-of-fluid) in two dimensional cartesien grids. The considered level-set and volume-of-fluid methods have been already tested in configurations with imposed velocity field in Chapter 3.

These coupling techniques are inspired by some well-known works in two-phase flows and are implemented in our in-house code on 2D cartesian (MAC) grids. A cell-centered implementation of the flow solver coupled with the volume-of-fluid method, similar to that of the free CFD software Basilisk (Popinet et al.) has been also implemented. This latter served in the comparison between cell-centered and MAC grids.

The different coupling manners of the level-set method with the flow solver include: sharp interface representation (using the GFM method), smooth interface approach (using the smoothed Heaviside H_ϵ) and an intermediate approach which consists of using the GFM method with locally-averaged fluid properties. Other variants which consider a sharp variation of density with a smooth viscosity are also implemented and tested. The volume-of-fluid methods are based on the CSF approach for the surface tension. They differ either in the interface advection method or in the curvature estimation.

The comparison between these methods is performed through a set of well-known test cases. Table 4.10 gives an overview of the performances of each method. It is clear that the volume-of-fluid methods in the exception of the VOF-HF2 method have a good overall behavior. Indeed, the VOF-HF2 method does not reach the end of some simulations due to a failure in curvature computation. The smoothing kernel makes the results more accurate in some cases, but it make it worse in others. It is seen that the VOF-HF1 and VOF-WY methods, which differ only in the volume-of-fluid advection scheme, give very similar results for all the simulations. We can conclude that the influence of the advection schemes (Weymouth and Yue (2010) and Scardovelli and Zaleski (2003)) is negligible here.

For the particular problem of spurious currents, the amplitude of the spurious currents decreases in time with the VOF methods as well as with the LS methods when the level-set field is not reinitialized. However, when we reinitialize the level-set method after each advection step, the amplitude of these spurious currents is much higher.

In the level-set approach, the LS-GFMϵ and LS-LA2 methods seem to be the most accurate. These two methods perform well for most of the simulations, and both have performances comparable to those of the VOF methods. On the other hand, the LS-LA1 method gives poor results for many flow problems.

Table 4.10: Summary of the results with the different methods. The density and viscosity ratios are denoted as $\rho_r = \rho_l/\rho_g$ and $\mu_r = \mu_l/\mu_g$. SpuC-$\kappa \equiv$ Spurious currents due to inaccurate curvature computation. SpuC $\equiv$ spurious currents. OD $\equiv$ Oscillating drop. CapW $\equiv$ Capillary wave. RTI $\equiv$ Rayleigh-Taylor instability. RB1 $\equiv$ Rising Bubble benchmark 1. RB2 $\equiv$ Rising Bubble benchmark 2.
Excellent: +++. Good: ++ ; Fair: + ; Poor: - ; ?: not performed.

Group	Zero viscosity			Uniform viscosity			Non-uniform viscosity				
Test case	SpuC-κ	OD	RTI	CapW	RTI	RTI	SpuC	OD	CapW	RB1	RB2
	$\mu = 0$ $\rho_r : 1, 10^3, 10^5$ $\sigma = 73$ $g = 0$	$\mu = 0$ $\rho_r : 10^3$ $\sigma = 1$ $g = 0$	$\mu = 0$ $\rho_r : 4$ $\sigma = 1$ $g = 1$	$\mu_r : 1$ $\rho_r : 1$ $\sigma = 1$ $g = 0$	$\mu_r : 1$ $\rho_r : 7.23$ $\sigma = 0$ $g = 9.81$	$\mu_r : 1$ $\rho_r : 7.23$ $\sigma = 0.1337$ $g = 9.81$	$\mu_r : 1 - 1000$ $\rho_r : 1 - 1000$ $\sigma = 1$ $g = 0$	$\mu_r : 200$ $\rho_r : 100$ $\sigma = 1$ $g = 0$	$\mu_r : 55.7$ $\rho_r : 850$ $\sigma = 1$ $g = 0$	$\mu_r : 10$ $\rho_r : 10$ $\sigma = 24.5$ $g = 0.98$	$\mu_r : 100$ $\rho_r : 1000$ $\sigma = 1.96$ $g = 0.98$
VOF-HF1	++	++	+	+++	++	++	+++	++	++	+++	++
VOF-WY	++	++	+	+++	++	++	+++	++	++	+++	++
VOF-HF2	+	++	−	+++	++	++	+++	+++	++	++	−
VOF-WYC	++	+[a]/++[b]	++	++	+++	++	+++	−	++	+++	+++
MOF-HF1	−	−	+	+++	++	++	+++[c] / −[d]	++	−	+++	++
LS-CSF	−	−[e] / +++[f]	++	++	++	++	−	+	++	+++	−
LS-LA1	−	−	−	++	−	++	+++[c] / −[d]	+	+	+++	+
LS-LA2	+++	+++	−	++	++	+++	+++	+++	+	+++	+
LS-GFMk	++	+	++	++	++	+	−	++	−	++	+
LS-GFMϵ	+++	+	++	++	++	+	+++	++	−	++	+
LS-GFM1	+	+	++	++	++	+	−	++	+++	+++	+
Gerris	?	+++	?	++	+++	+	?	?	++	?	?

[a] without smoothing kernel
[b] with smoothing kernel
[c] for high viscosity
[d] for low viscosity
[e] with reinitialization after each advection
[f] with CFLreinit = 0.2

Conclusions and perspectives

The aim of this work was to develop a numerical method to simulate two-phase incompressible flows with non-miscible fluids. It is a complex numerical problem that implies the knowledge of the theoretical background of fluid mechanics, mathematics and numerical analysis. This thesis focused on the analysis and the comparison of several numerical methods, especially those dealing with the interface capturing and their coupling with the flow solver. This work is split into four parts. The first three parts deal with the interface transport in an imposed velocity field, and the last one with the coupling with the flow solver.

The first part is devoted to the analysis and selection of the robust numerical tools for the level-set method. In particular, the reinitialization step and the advection schemes were rigorously investigated including a wide range of numerical schemes. The reinitialization step keeps the signed-distance property of the level-set field, which is needed to accurately determine interface properties such as the curvature and the normal to the interface. However, this step tends to move the interface and thereby causes the change of the reference fluid mass. We implemented two enhancements of the reinitialization step: the sub-cell fix proposed by Russo and Smereka (2000) and the volume constraint by Sussman et al. (1999). The well-known test problems of the Zalesak disk, shearing vortex and time-reversed single vortex are used to verify the accuracy of these methods. The volume constraint is much more robust especially with conservative schemes, in particular the fifth-order WENO scheme. In addition, it is less sensitive to the CFL number.

The second part is devoted to the volume-of-fluid methods, for which it is much easier to conserve mass. The directional-splitting conservative (with respect to the mass of the reference fluid) and consistent (which keeps the volume fraction C between 0 and 1) geometrical resolution (Scardovelli and Zaleski, 2003; Tryggvason et al., 2011) for the volume fraction advection is studied. We included in our comparison the alternative directional-splitting conservative and consistent scheme developped by Weymouth and Yue (2010), which can be easily extended to 3D cartesian grids. In both methods, the piecewise linear interface reconstruction is performed using the ELVIRA algorithm which combines accuracy and less CPU resources. The results of these two VOF methods are almost identical in passive advection tests. In contrast to the level-set method which tends to thicken thin interface filaments, the VOF methods break it into fragments in the case where more than one interface pass through the reconstruction blocks of 3×3 cells. The moment-of-fluid method (Dyadechko and Shashkov, 2005, 2006, 2008) reconstructs the interface in each mixed cell using the center of mass of the reference fluid in that cell. Thus, these centroids are also advected using a similar technique as for the volume fractions. The MOF method is more accurate than the VOF methods in the passive advection tests and is a promising method since it does not break the filament. However, some numerical difficulties

have been encountered especially for small CFL number, but this may be avoided by a careful implementation of this method.

The third part deals with coupled level-set and volume-of-fluid methods. These coupled techniques are supposed to combine the advantages of both constituting methods, namely a good interface representation and the mass conservation. Some coupled methods were easy to implement such as the VOSET method (Sun and Tao, 2010). In fact, this latter needs only a supplementary routine for determining the signed-distance to the interface. The accuracy of the VOSET methods depends first on the accuracy of the used VOF method, then on that of the algorithm computing the level-set field. Other methods are more complex such as the CLSVOF (Sussman and Puckett, 2000; Son and Hur, 2002) and CLSMOF (Jemison et al., 2013) methods. In our implementation of coupled approaches, the VOF fractions are advected using the conservative and consistent scheme. As a consequence, algorithms for the volume fraction redistribution (for cases where $C > 1$ or $C < 0$) are not required.
We investigated the VOSET, MCLS (van der Pijl et al., 2005), CLSVOF and CLSMOF (which couples LS with MOF) methods. Some of these coupled methods use alternative approaches to overcome fastidious algorithms for the VOF method. In general, the behavior of the coupled methods is very close to that of the VOF methods. For example, the coupled techniques break thin filaments. Only the degree of regularity and length of the non-fractured vortex filament differ from a method to another. In the MCLS coupling, the use of the standard reinitialization equation with 1st- or 2nd-order advection schemes provide more accurate results than with enhanced reinitialization methods and high-order advection schemes. The CLSVOF methods produce similar results as those of the VOF method, and the CLSMOF provides the most accurate results.

The last part concerns the coupling between the incompressible Navier-Stokes solver and the interface method. The flow solver is based on a 2D uniform staggered (MAC) grid. The convective terms are discretized in space using the 2nd-order ENO scheme. The diffusion terms are treated implicitly and discretized in space using the 2nd-order central-difference scheme. The pressure-velocity decoupling is realized through the projection method. We have implemented six different methods based on the level-set approach and five on the VOF approach, inspired by some well-known works from the literature. In the level-set based flow solver, the treatment of the interface is realized either through a

- Smoothed pressure jump (CSF, Brackbill et al. (1992); Chang et al. (1996)) with smoothed fluid properties
- Sharp pressure jump (GFM) with jump in fluid properties (Son and Dhir, 2007)
- Sharp pressure jump (GFM) with jump in density, and smoothed viscosity (Desjardins et al., 2008)
- Sharp pressure jump (GFM) with locally averaged fluid properties (Vigneaux, 2008)

In the VOF-based flow solver, the fluid properties are computed using the volume-of-fluid weighted-average. The main difference is the interface method itself (VOF (Scardovelli and Zaleski, 2003), VOF (Weymouth and Yue, 2010) or MOF). Thus we have three VOF methods for which we add one coupled to a Navier-Stokes solver with a collocated grid setting. All of these methods use the Height Function by Popinet (2009) except one which uses the Height Function implementation of Hernández et al. (2008).
These 11 methods are compared in different flow configurations: spurious currents in a stationary circular bubble, oscillating drops, standing capillary wave, Rayleigh-Taylor instability and

rising bubbles. The obtained results showed that the volume-of-fluid methods using the Height Function method by Popinet (2009) give very good results for most simulations. Concerning the level-set based solvers, we found that two methods generally give very good results and are comparable to those obtained with VOF methods. The LS-GFMϵ uses the GFM approach with a jump in density and a smoothed viscosity, while the interface curvature computed according to Chang et al. (1996). The LS-LA2 method uses also the GFM approach with locally-averaged fluid properties (Vigneaux, 2008), while the curvature is computed using a different method (Sussman et al., 1999). A very similar method to the LS-LA2, named LS-LA1, computes the interface curvature according to Vigneaux (2008) fails in many simulations.

Future work includes the validation of the flow solver coupling with coupled volume-of-fluid and level-set methods in 2D, then the axisymmetric flow solver with level-set and VOF methods. This will allow to simulate, for example, bubbles, droplets and two-phase flows inside circular channels. Eventually, the extension of the code to fully three dimensional grids should be considered in order to be able to treat a wide range of configurations.

Including the energy equation and accurate interface conditions (to take into account the change in the mass of the reference fluid due to the phase change) to the actual code will allow to simulate bubble growth during phase change, which is of a major interest to our laboratory. Other phenomena such as film boiling could be then predicted and studied only flat solids. In order to take into account the phase change on irregular solid surfaces, accurate velocity boundary conditions should be applied. This can be performed using the immersed boundary method.

From a computational time point-of-view, it would be interesting either to extend the current sequential implementation to parallel computers or to use the local grid refinement. In this context, the use of open-source CFD codes such as Gerris (Popinet) and Basilisk (Popinet et al.) is interesting. This will allow to study more realistic problems and help in the understanding of the physical phenomena behind phase change.

Appendices

Appendix A

Interface Methods

A.1 Mass and positioning error computation

The mass loss/gain error is given by:

$$E_M = \frac{\text{mass} - \text{Exact mass}}{\text{Exact mass}} = \frac{\int_\Omega H(\phi_C)d\Omega - \text{Exact mass}}{\text{Exact mass}} \tag{1.59}$$

The positioning error E_1 is expressed as:

$$E_1 = \frac{\int_\Omega |H(\phi_E) - H(\phi_C)| \, d\Omega}{\int_\Omega H(\phi_E)d\Omega} \tag{1.60}$$

The Heaviside integral in equations (1.59) and (1.60) can be computed by different methods. However, only few authors tell how they evaluate it. Sussman and Puckett (2000) proposed the following procedure:

For each staggered cell $[x_i, x_{i+1}] \times [y_j, y_{j+1}]$, check:

1. if $\max[\phi(i : i+1, j : j+1)] > 0$, then the cell is entirely inside the reference fluid. and so $\int_{\Omega_{ij}} H(\phi_{i,j})d\Omega = \Delta x \Delta y$.

2. if $\min[\phi(i : i+1, j : j+1)] < 0$, then the cell is entirely outside the reference. Hence, $\int_{\Omega_{ij}} H(\phi_{i,j})d\Omega = 0$.

3. Elsewhere, the cell contains both fluids. Subdivide the cell into $m \times m$ sub-cells (see Fig. A.1). Interpolate the level-set values to the center of each sub-cell and get $\overline{\phi(k,l)}$, $(k,l) \in [1,m]^2$. Then, the area occupied by the reference fluid will be the sum of filled sub-cell areas which is estimated from:

$$\int_{\Omega_{ij}} H(\phi_{i,j})d\Omega = \frac{\Delta x \Delta y}{m^2} \sum_{k=1}^{m} \sum_{l=1}^{m} H(\overline{\phi(k,l)})$$

where

$$H(\overline{\phi(k,l)}) = \begin{cases} 1 & \text{if } \overline{\phi(k,l)} \geq 0 \\ 0 & \text{elsewhere} \end{cases} \tag{A.1}$$

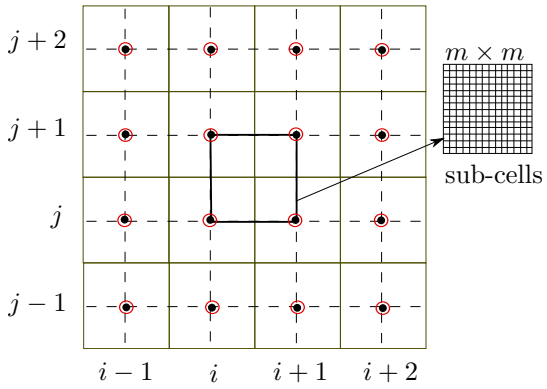

Figure A.1: Staggered grid for mass estimation. Interpolation is performed within the dashed cells through which interface passes. The exact and the computed level-set are defined at the vertices of the dashed cells represented by red circles.

The total area of the reference fluid is the sum of its areas in each computational cell of the domain.

The mass error due to the approximation of the Heaviside Eq. (A.1) for a given grid and sub-cells number is bounded by:

$$L\Delta x \geq \left|\text{Exact Mass} - \int_{\Omega} H(\phi_E)d\Omega\right| \geq \frac{L\Delta x}{m^2}$$

where L is interface length.

Sussman and Puckett (2000) have used the bilinear interpolation which is 2nd-order. We have also implemented bicubic interpolation based on fourth-order Lagrange polynomials.

If the staggered cell delimited by the grid points $(i,j),(i,j+1),(i+1,j)$ and $(i+1,j+1)$ contains both fluids, we subdivide it into $m \times m$ cells and the bilinear interpolation yields the values of the level-set on the center of sub-cells (x_k, y_l):

$$\overline{\phi(k,l)} = \frac{1}{\Delta x \Delta y}\Big(\phi_{i,j}(x_{i+1}-x_k)(y_{j+1}-y_l) + \phi_{i+1,j}(x_k-x_i)(y_{j+1}-y_l) + \phi_{i,j+1}(x_{i+1}-x_k)(y_l-y_j) + \phi_{i+1,j+1}(x_k-x_i)(y_l-y_j)\Big)$$

where $x_k = x_i + \frac{k-\frac{1}{2}}{m}\Delta x$, $y_l = y_i + \frac{l-\frac{1}{2}}{m}\Delta y$ and $(k,l) \in [1,m]^2$. $\overline{\phi(k,l)}$ refers to interpolated level-set into the position (x_k, y_l).

Fourth-order Lagrange polynomials are based on a stencil of 4×4 points. We interpolate the level-set between i and $i+1$ first in the x-direction. For the j-th column for example, the interpolation formula reads:

$$\widetilde{\phi(k,j)} = \frac{1}{6(\Delta x)^3}\Big(- \phi_{i-1,j}(x_k-x_i)(x_k-x_{i+1})(x_k-x_{i+2}) + 3\phi_{i,j}(x_k-x_{i-1})(x_k-x_{i+1})(x_k-x_{i+2}) - 3\phi_{i+1,j}(x_k-x_{i-1})(x_k-x_i)(x_k-x_{i+2}) + \phi_{i+2,j}(x_k-x_{i-1})(x_k-x_i)(x_k-x_{i+1})\Big)$$

where $x_k = x_i + \dfrac{k-\frac{1}{2}}{m}\Delta x$ and $k \in [1, m]$.

Then, we interpolate these level-set values between j and $j+1$ following the y-direction on positions x_k with $k \in [1, m]$ as follows:

$$\begin{aligned}\overline{\phi(k,l)} = \frac{1}{6(\Delta y)^3}\Big(&-\widetilde{\phi_{k,j-1}}(y_l - y_j)(y_l - y_{j+1})(y_l - y_{j+2}) + \\ &3\widetilde{\phi_{k,j}}(y_l - y_{j-1})(y_l - y_{j+1})(y_l - y_{j+2}) - \\ &3\widetilde{\phi_{k,j+1}}(y_l - y_{j-1})(y_l - y_j)(y_l - y_{j+2}) + \\ &\widetilde{\phi_{k,j+2}}(y_l - y_{j-1})(y_l - y_j)(y_l - y_{j+1})\Big)\end{aligned}$$

The knowledge of the reference field and the computed level-set field on each computational staggered cell is needed in the positioning error E_1 estimation. We apply then the same treatment as for the determination of Heaviside integral.

$$E_1 = \frac{\int_\Omega |H(\phi_E) - H(\phi_C)|\, d\Omega}{\int_\Omega H(\phi_E) d\Omega} = \frac{\sum_{(i,j)\in\Omega} \int_{\Omega_{ij}} |H(\phi_E(i,j)) - H(\phi_C(i,j))|\, d\Omega}{\text{Exact mass}}$$

where $\displaystyle\int_{\Omega_{ij}} H(\phi_E(i,j)) d\Omega$ is the Heaviside integral on the cell (i,j).

The procedure we used to compute the numerator for each staggered cell is described below with the use of the following notation:

$$\begin{cases} \max\limits_E = \max\left[\phi_E(i:i+1, j:j+1)\right] \\ \min\limits_E = \min\left[\phi_E(i:i+1, j:j+1)\right] \\ \max\limits_C = \max\left[\phi_C(i:i+1, j:j+1)\right] \\ \min\limits_C = \min\left[\phi_C(i:i+1, j:j+1)\right] \end{cases}$$

- If $\max\limits_E \times \max\limits_C > 0$ or $\min\limits_E \times \min\limits_C > 0$, then the cell is filled or empty of reference fluid in both solutions. Hence,

$$\int_{\Omega_{ij}} \big|H(\phi_E(i,j)) - H(\phi_C(i,j))\big| d\Omega = 0$$

- If $\min\limits_E \times \max\limits_E < 0$ and $\min\limits_C \times \max\limits_C < 0$, then the cell is mixed in both cases. Hence, we subdivide it into $m \times m$ sub-cells, and

$$\int_{\Omega_{ij}} \big|H(\phi_E(i,j)) - H(\phi_C(i,j))\big| d\Omega = \frac{\Delta x \Delta y}{m^2} \sum_{k=1}^{m}\sum_{l=1}^{m} \big|H(\overline{\phi_E(k,l)}) - H(\overline{\phi_C(k,l)})\big|$$

- Else if $\min\limits_E \times \max\limits_E < 0$, then the cell contains both fluids in the exact solution and only one fluid in the computed one. Hence, we interpolate the exact level-set at the centers of the $m \times m$ sub-cells, and

$$\int_{\Omega_{ij}} \big|H(\phi_E(i,j)) - H(\phi_C(i,j))\big| d\Omega = \left| \frac{\Delta x \Delta y}{m^2} \sum_{k=1}^{m}\sum_{l=1}^{m} H(\overline{\phi_E(k,l)}) - \int_{\Omega_{ij}} H(\phi_C(i,j)) d\Omega \right|$$

- Else if $\min\limits_C \times \max\limits_C < 0$, then the cell contains both fluids in the computed solution and one

fluid in the exact one. Hence, we interpolate the computed level-set at the centers of the $m \times m$ sub-cells, and

$$\int_{\Omega_{ij}} \left|H(\phi_E(i,j)) - H(\phi_C(i,j))\right| d\Omega = \left| \int_{\Omega_{ij}} H(\phi_E(i,j)) d\Omega - \frac{\Delta x \Delta y}{m^2} \sum_{k=1}^{m} \sum_{l=1}^{m} H(\overline{\phi_C(k,l)}) \right|$$

In order to evaluate the accuracy of this approach, we compute the exact signed-distance function (ϕ_E) to a circle of radius $r = 0.15$ centered at $(0.25, 0.5)$ in a 1×1 domain (see Fig. 1.5). Its exact area is πr^2. Grid sizes are varied in the range of 1/8 up to 1/1024 and the number of sub-cells from 1 (no sub-cell) to 2000^2. Fig. A.2 shows the absolute value of the mass error E_M using the bilinear (left) and the bicubic (right) interpolations of the level-set function within mixed staggered cells. For the bilinear interpolation, the accuracy of mass estimation is

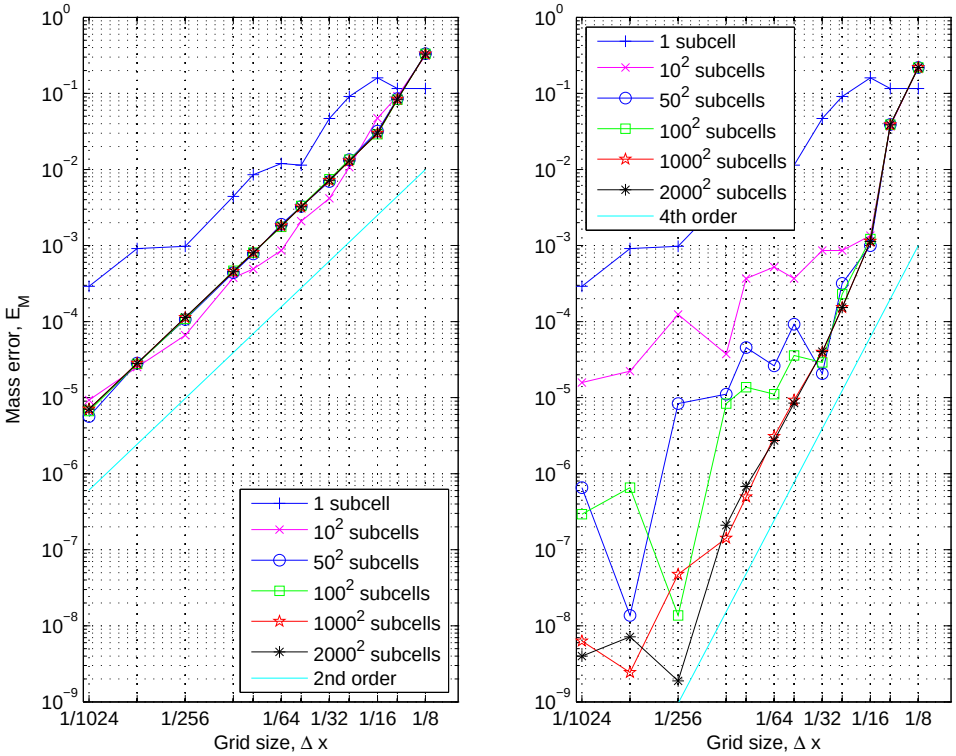

Figure A.2: Mass error E_M of the circle of radius 0.15 defined by the exact signed-distance function, as a function of grid and sub-grid sizes. left: Bilinear interpolation; right: Bicubic interpolation.

second-order and the error is almost independent from the number of sub-cells except for $m = 1$. For the bicubic interpolation, error decreases by increasing the number of sub-cells. We see also that for a given sub-cell number, the error is fluctuating depending on the grid; this is due to the position of the interface on the grid. For $m \geq 1000$, the bicubic interpolation is 4th-order accurate. We remark also that the 4th-order Lagrange polynomials provides a better measure of mass than the bilinear interpolation.

Now, we consider a more complicated interface with sharp angles which is the Zalesak notched disk (see §1.5.2). Its exact area is given by $(\pi R^2 - w(\sqrt{R^2 - (w/2)^2} + 3R - 2h)/2)$ where w is the notch width, $(R + \sqrt{R^2 - (w/2)^2} - h)$ is the notch depth and R the radius. Fig. A.3 shows the relative errors for grid sizes in the range of 1/16 to 1/1024 and for sub-cells number m^2 going from 50^2 up to 2000^2.

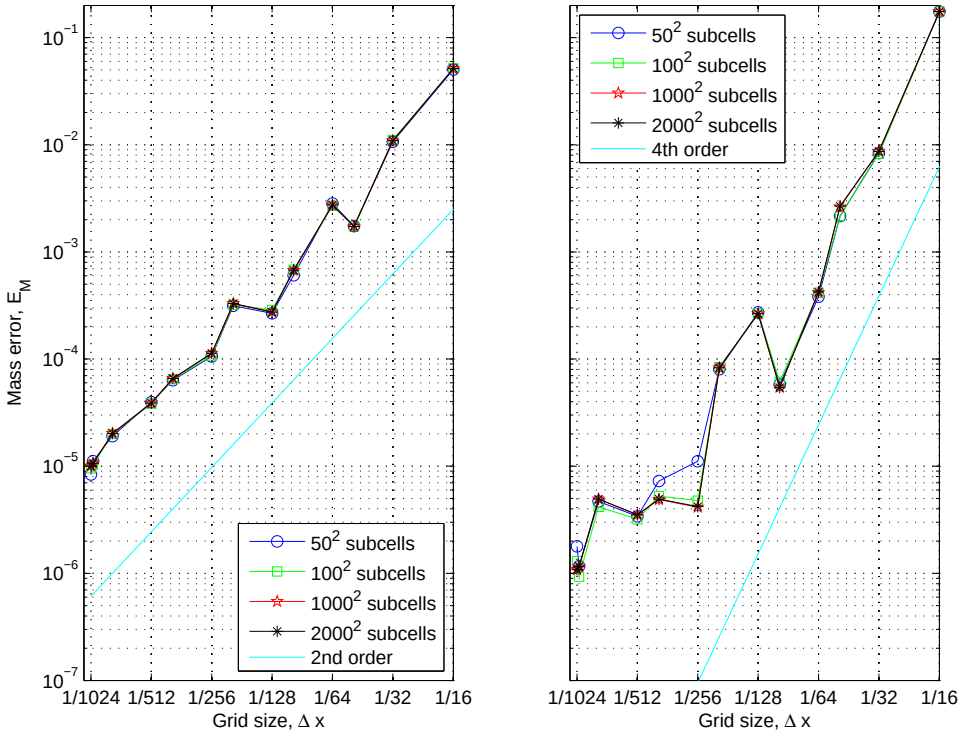

Figure A.3: Mass error E_M of the Zalesak disk defined by the exact signed-distance function on different grid and sub-grid sizes. left: Bilinear interpolation; right: Bicubic interpolation.

The order of the mass error, E_M, is the same as the order ot the used interpolation. However, we can remark that the error depends mainly on the position of the angles and the notch on the grid. We remark also that the order of convergence is approximately identical to that of the used interpolation and that the bicubic interpolation remains more precise than the bilinear one.

As the bicubic interpolation provides a better estimation of mass, we use it to estimate the positioning error E_1 and the mass error E_M, with a number of sub-cells $m^2 = 1000^2$ unless otherwise specified.

A.2 Determination of the reference solution of the single vortex

Shearing Vortex

The knowledge of the exact position of the interface is compulsory since it is needed to compute the positioning error E_1. However, in this test case, interface position is not given by an analytical formula; only the initial and final positions are exactly known. The exact interface is determined at any time of the simulation as follows:

1. At $t = 0$, seed the interface with equidistant massless markers (10^5 markers)
2. Advect these markers by solving the equation of motion for a particle

$$\vec{u} = \frac{d\vec{x}}{dt}$$

using the fourth-order Runge-Kutta scheme

$$\vec{x}^{(n+1)} = \vec{x}^{(n)} + \frac{\Delta t}{6}\left(\vec{u}^{(1)} + 2\vec{u}^{(2)} + 2\vec{u}^{(3)} + \vec{u}^{(4)}\right)$$

where

$$\begin{cases} \vec{u}^{(1)} = \vec{u}(\vec{x}) \\ \vec{u}^{(2)} = \vec{u}\left(\vec{x} + \dfrac{\Delta t}{2}\vec{u}^{(1)}\right) \\ \vec{u}^{(3)} = \vec{u}\left(\vec{x} + \dfrac{\Delta t}{2}\vec{u}^{(2)}\right) \\ \vec{u}^{(4)} = \vec{u}\left(\vec{x} + \Delta t\vec{u}^{(3)}\right) \end{cases}$$

The time-step used for the advection of markers is set to $\Delta t = \min(h, 10^{-2})$ where h is the time-step of the interface capturing method. Such value of Δt ensures that between the two times $t = 0$ and $t = T = 6$, the maximum displacement of the markers is in the range of 10^{-8}. By linking the set of markers at a given time, we obtain the exact interface location.

3. Reconstruct the signed-distance function, which is the distance from the center of each computational cell to the closest marker. We consider it as the reference solution. Both, the reference solution and the level-set are defined at the same grid points; the positioning error E_1 is therefore computed using the bicubic interpolation with 1000^2 sub-cells (see §1.5.1).

The mass error E_M of the reference solution is also computed using the bicubic interpolation with 1000^2 sub-cells, and is plotted on Fig. A.4 for grid sizes going from $1/50^2$ downto $1/800^2$ at time $t = T/2 = 3$. The positioning error of the reference solution is of the same range as the mass error E_M. These errors are too small comparing to that of the transported level-set, it represents approximately 1%.

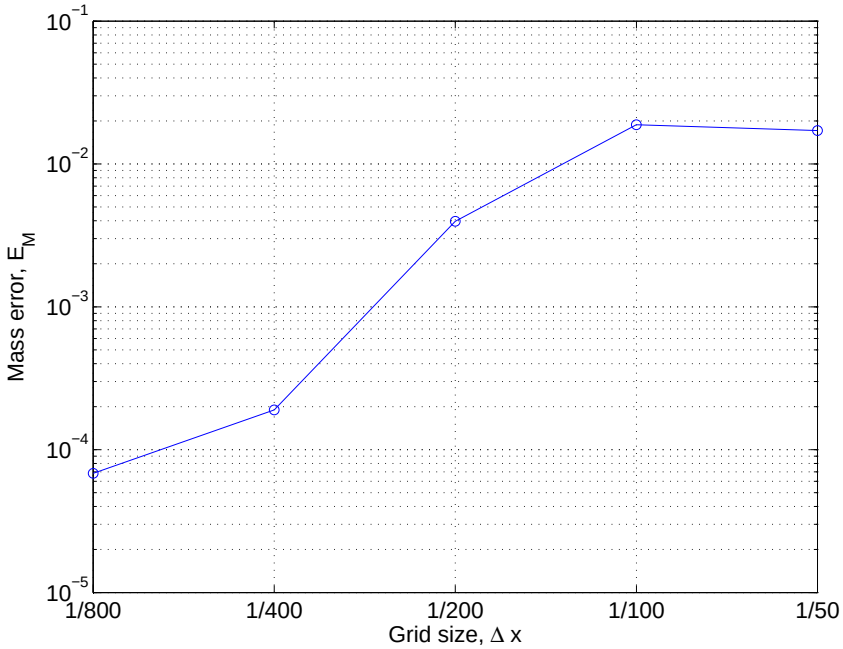

Figure A.4: Mass error E_M due to the bicubic interpolation of the reference solution defining the vortex interface at time $t = T/2 = 3$.

Time-reversed single vortex

The time-reversed single vortex problem is defined on §1.5.4. The reference solution at any time, except at the start and the end of the simulation, is determined using the Lagrangian method

detailed in the previous section.

Fig. A.5 shows E_M of the reference solution for grid sizes in the range $[1/512, 1/32]$ at time $t = T/2 = 1$. The positioning error of the interpolated signed-distance function is of the same range as the mass error, and this error is too small comparing to the mass error and the positioning error of the computed level-set, it represents approximately 1% of the level-set error.

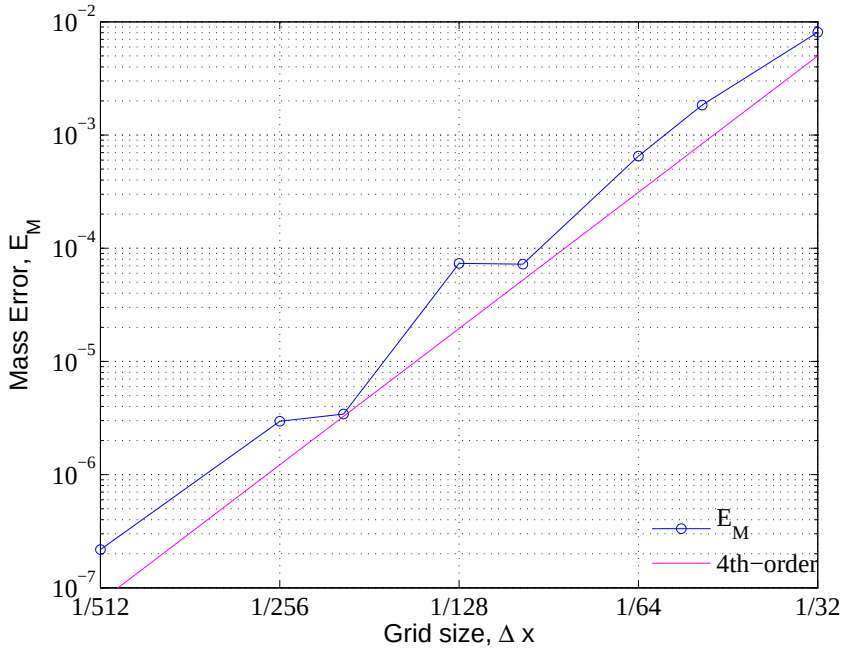

Figure A.5: Mass error E_M due to the bicubic interpolation of the reference solution defining the time-reversed single vortex interface at time $t = T/2 = 1$.

A.3 VOF: Distance and fraction computation

The objective here is to compute the fraction C inside a computational cell (2D or 2D axisymmetric) given the line equation

$$\vec{x} \cdot \vec{n} + d = 0 \tag{A.2}$$

where $\vec{n}$ is the normal and d the distance to the interface. We also consider the inverse problem in which the fraction C and the normal $\vec{n}$ to the interface are known and we want to deduce the distance d. These two procedures are needed to reconstruct the interface and to compute the mass fluxes through the cell faces. This problem was already treated in Li (1995); Gueyffier et al. (1999); Scardovelli and Zaleski (2000) for 2D and 3D configurations, while Son and Hur (2002) have detailed the algorithm for 2D cartesian and axisymmetric coordinates. Other methods use an iterative procedures to find the distance to the cell center Rider and Kothe (1998). The main challenge when designing robust algorithms is to take into consideration round-off errors in limiting cases, which tend to produce non-physical results. Examples are when the interface is nearly horizontal or vertical, and/or when the fraction is close to zero or one.

Distance between a line and a point

We consider the 2D cell plotted on Fig. A.6 with size $\Delta x \times \Delta y$. The origin O is defined here at the center of the cell. As an example, suppose we want to compute the distance between the red line, passing through the lower-left corner of the cell with coordinates $(\frac{-\Delta x}{2}, -\frac{\Delta y}{2})$, and the cell center O. Since the normal components are normalized, we can write them as a function of the

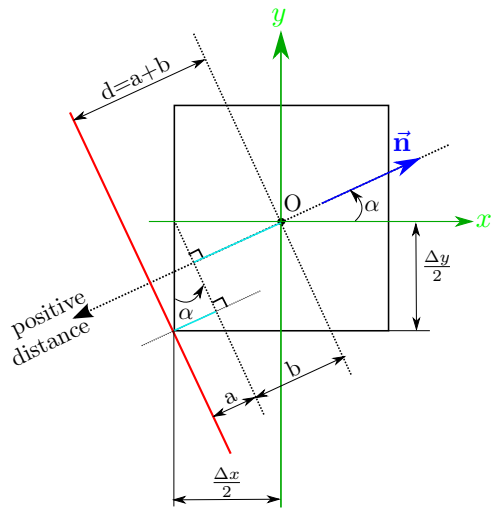

Figure A.6: Distance between a line and a point.

angle α, with $\vec{n} = (\cos(\alpha), \sin(\alpha))$. This (minimal) distance d between the cell center and the red line is equal to the sum of the two distances a and b, where $b = \frac{\Delta x}{2}\cos(\alpha)$ and $a = \frac{\Delta y}{2}\sin(\alpha)$, so that

$$d = \frac{\Delta x}{2} n_x + \frac{\Delta y}{2} n_y$$

Of course, this can also be obtained directly from Eq. (A.2) by setting $\vec{x} = (\frac{-\Delta x}{2}, -\frac{\Delta y}{2})$.

A.3.1 Fraction computation

We want to compute the fraction of the reference fluid inside a 2D computational cell given a linear interface with a unit normal vector $\vec{n} = (n_x, n_y)$ and a distance d to the cell center O. The

line equation is then $n_x x + n_y y + d = 0$. Several cases need to be considered depending on the normal orientation and the distance to the cell center. In order to reduce the number of possible cases, and without loss of generality, we rotate the interface in a way that the normal always points to the positive x- and y-directions.

The convention for the distance from the cell center to the interface is as follows:

- The distance is zero if the linear interface passes through the cell center, in which case the fraction is equal to 1/2 in 2D/3D cartesian setting.
- If the cell center is inside the reference fluid as in Fig. A.7, the distance is defined negative, in which case the fraction is greater than 1/2. The cell center is then below the line.
- Finally, if the cell center is outside the reference phase then the distance is positive and therefore the fraction is smaller than 1/2. The cell center is then above the line.

This way, the normal vector is pointing in the outward direction of the reference fluid.

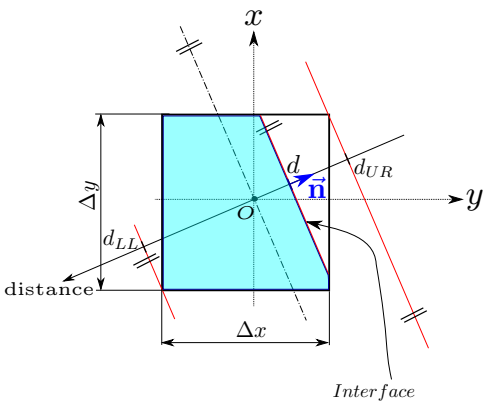

Figure A.7: Linear interface segment and the characteristic distances.

Two extreme distances could then be defined as shown on Fig. A.7. The first one corresponds to the distance from the cell center to the linear interface with normal (n_x, n_y) passing through the upper-right corner, denoted as d_{UR}. The second one is the distance from the cell center to the line with the normal (n_x, n_y) passing through the lower-left corner denoted as d_{LL} with $(d_{\mathrm{LL}} = -d_{\mathrm{UR}})$.

According to the adopted sign of the distance, d_{LL} is positive and d_{UR} is negative. The meaning of these two distances is that if $d > d_{LL}$ then the cell is empty, while the cell is full if $d < d_{UL}$.

The eight possible cases for the linear interface disposition are depicted on Fig. A.8, while Fig. A.9 plots the fraction as a function of the distance for a linear interface which is not horizontal neither vertical. The line is considered horizontal if $|n_y| \leq \epsilon$ and vertical if $|n_x| \leq \epsilon$, where ϵ is a small parameter set to 10^{-8}.

The fraction C inside the cell is computed following Algorithm A.1. Round-off errors that may appear if X or Y are much greater than unity or close to zero are avoided in the current implementation. Note that in the case where $\Delta x = \Delta y$, the algorithm could be made more compact since the number of cases is reduced to 5.

Algorithm A.1 Fraction computation in 2D cartesian grid system (for the different cases, refer to Fig. A.8).

Input: $\vec{n} = (n_x, n_y)$, d, Δx, Δy

$d_{LL} \leftarrow (|n_x|\,\Delta x + |n_y|\,\Delta y)\,/2$

if $(d \geq d_{LL})$ **then** ▷ the cell is empty (case 1)

$C \leftarrow 0$

else if $(d \leq -d_{LL})$ **then** ▷ the cell is full (case 2)

$C \leftarrow 1$

else if $(|n_y| < 10^{-8})$ **then** ▷ the line is horizontal (case 3)

$C \leftarrow \dfrac{1}{2} - \dfrac{d}{\Delta x}$

else if $(|n_x| < 10^{-8})$ **then** ▷ the line is vertical (case 4)

$C \leftarrow \dfrac{1}{2} - \dfrac{d}{\Delta y}$

else

$X \leftarrow \dfrac{d_{LL} - d}{|n_x|\Delta x}$ Intersection point with the $x-$axis

$Y \leftarrow \dfrac{d_{LL} - d}{|n_y|\Delta y}$ Intersection point with the $y-$axis

if $(X \leq 1$ & $Y \leq 1)$ **then** ▷ (case 5)

$C \leftarrow XY/2$

else if $(X \geq 1$ & $Y \geq 1)$ **then** ▷ (case 6)

$X_2 \leftarrow \dfrac{d_{LL} + d}{|n_x|\Delta x}$ Intersection point with the $x-$axis

$Y_2 \leftarrow \dfrac{d_{LL} + d}{|n_y|\Delta y}$ Intersection point with the $y-$axis

$C \leftarrow 1 - X_2 Y_2/2$

else if $(X \leq 1)$ **then** ▷ (case 7)

$X_1 \leftarrow \dfrac{\frac{1}{2}\left(|n_x|\Delta x - |n_y|\Delta y\right) - d}{|n_x|\Delta x}$

$C \leftarrow (X + X_1)/2$

else ▷ $Y \leq 1$, (case 8)

$Y_1 \leftarrow \dfrac{\frac{1}{2}\left(-|n_x|\Delta x + |n_y|\Delta y\right) - d}{|n_y|\Delta y}$

$C \leftarrow (Y + Y_1)/2$

end if

end if

$C \leftarrow \max(0, \min(C, 1))$ ▷ Enforce fraction to be in the range $[0, 1]$

return C

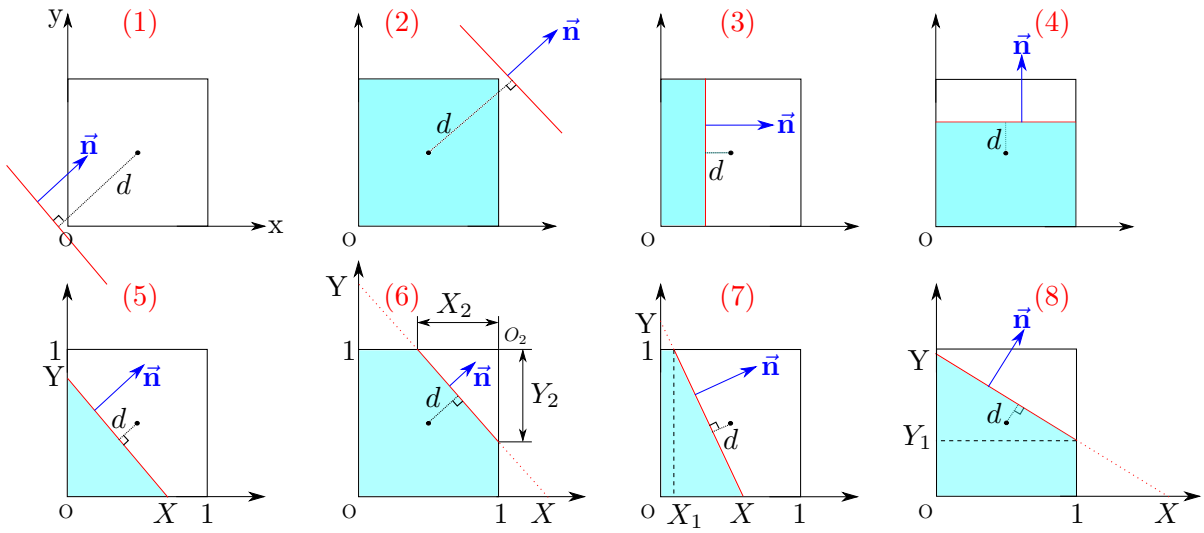

Figure A.8: Possible cases of the linear interface in 2D cartesian grid setting.

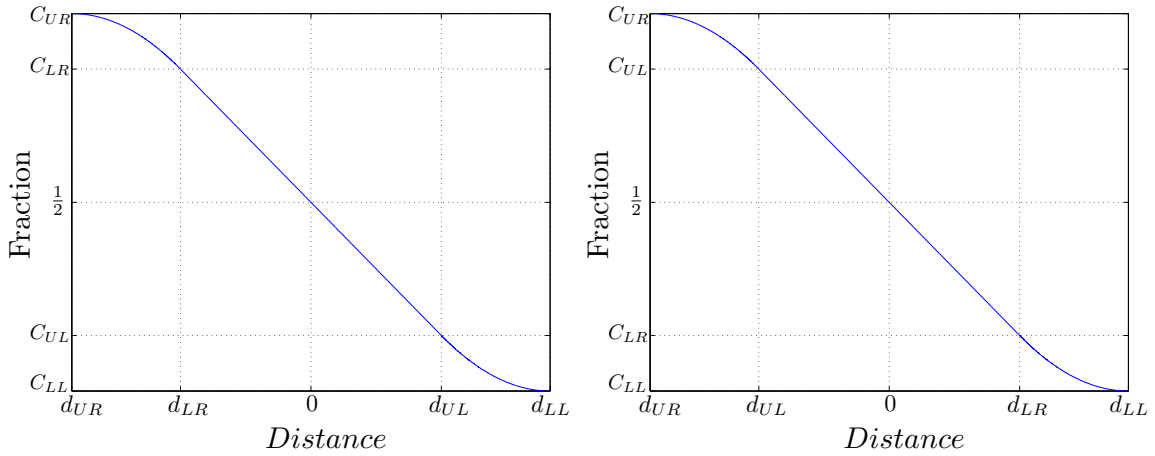

Figure A.9: Fraction of the reference fluid as a function of the distance for a given normal to the interface. The left figure corresponds to $|n_x| > |n_y|$, and the right one to $|n_x| < |n_y|$.

A.3.2 Distance computation

Now, we suppose that the volume fraction C inside the cell is known, the normalized components of the normal to the interface $\vec{n} = (n_x, n_y)$ are also known. We want to find the distance from this interface to the cell center that satisfies the fraction C inside this cell. This procedure is the inverse of what we have already seen on §A.3.1. Again, in order to reduce the number of possible cases, we rotate the interface in a way that the normal points out in the positive $x-$ and $y-$directions.

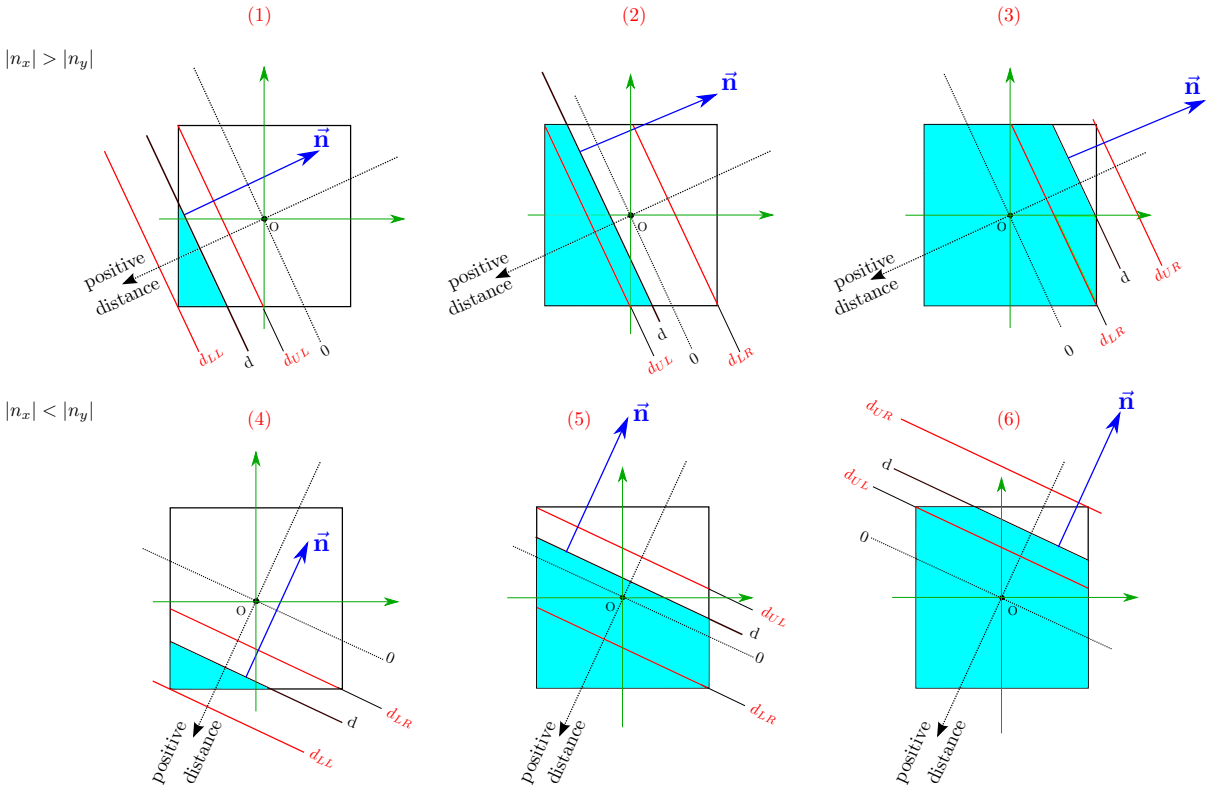

Figure A.10: Possible cases of the linear interface and the reference distances in 2D cartesian grid system.

The distance from the line to the lower-left, upper-left, upper-right and lower-right corners of the cell are, in 2D cartesian coordinates, given by:

$$\begin{aligned} d_{LL} &= |n_x| \frac{\Delta x}{2} + |n_y| \frac{\Delta y}{2} \\ d_{UL} &= |n_x| \frac{\Delta x}{2} - |n_y| \frac{\Delta y}{2} \\ d_{UR} &= -d_{LL} \\ d_{LR} &= -d_{UL} \end{aligned} \tag{A.3}$$

and the corresponding fractions to the two intermediate distances is computed using the Algorithm A.1:

$$\begin{aligned} C_{UL} &= \text{fraction below the line defined by } (|n_x|, |n_y|, d = d_{UL}) \\ C_{LR} &= \text{fraction below the line defined by } (|n_x|, |n_y|, d = d_{LR}) \end{aligned} \tag{A.4}$$

If the target fraction is C and the normal to the interface is $\vec{n}$, then we compare this target fraction with the two fractions C_{UL} and C_{LR}. The possible cases are represented on fig. A.10, and

depending on the value of the target fraction, we determine the distance d satisfying this target fraction. The procedure of determining the distance in 2D cartesian grid systems is detailed on Algorithm A.2.

Algorithm A.2 Distance computation in 2D cartesian grid system (for the different cases, refer to Fig. A.10).

Input: $\vec{n} = (n_x, n_y)$, C, Δx, Δy

if $(|n_y| < 10^{-8})$ **then** ▷ The line is horizontal

$d \leftarrow \left(\frac{1}{2} - C\right)\Delta x$

else if $(|n_x| < 10^{-8})$ **then** ▷ The line is vertical

$d \leftarrow \left(\frac{1}{2} - C\right)\Delta y$

else ▷ Compute the reference distances Eq. A.3 and the corresponding fractions Eq. A.4

if $(C \leq C_{UL}$ & $C \leq C_{LR})$ **then**

if $(C_{UL} < C_{LR})$ **then** ▷ Case (1), $|n_x| > |n_y|$

$d \leftarrow d_{LL} - \sqrt{(d_{UL} - d_{LL})^2 \frac{C}{C_{UL}}}$

else ▷ Case (4), $|n_y| > |n_x|$

$d \leftarrow d_{LL} - \sqrt{(d_{LR} - d_{LL})^2 \frac{C}{C_{LR}}}$

end if

else if $(C \geq C_{UL}$ & $C \geq C_{LR})$ **then**

if $(C_{UL} < C_{LR})$ **then** ▷ Case (3), $|n_x| > |n_y|$

$d \leftarrow d_{UR} + \sqrt{(d_{LR} - d_{UR})^2 \frac{1 - C}{1 - C_{LR}}}$

else ▷ Case (6), $|n_y| > |n_x|$

$d \leftarrow d_{UR} + \sqrt{(d_{UL} - d_{UR})^2 \frac{1 - C}{1 - C_{UL}}}$

end if

else

if $(|C_{LR} - C_{UL}| < 10^{-12})$ **then** ▷ The line passes through the cell center

$d \leftarrow \frac{d_{UL} + d_{LR}}{2}$

else ▷ Linearly interpolate the distance, case (2) or (5)

$\theta \leftarrow (C - C_{UL})/(C_{LR} - C_{UL})$

$d \leftarrow \theta d_{LR} + (1 - \theta) d_{UL}$

end if

end if

end if

return d

Appendix B

2D two-phase flow solver

B.1 Collocated Navier-Stokes solver

One of the major advantages of the collocated arrangement, where all variables including the velocity are defined at the cell center, is that it greatly facilitates the implementation of adaptive grid refinement (Almgren et al., 1998; Popinet, 2003). In order to avoid the well-known checker-board problem associated with the discretization of the pressure equation on collocated grids, Puckett et al. (1997) developed an approximate projection method for two-phase flows (coupled to the VOF method), based on the initial version by Almgren et al. (1996) for low mach number flows. Many variants of this approach have been used since, including non-incremental projection methods (Popinet, 2009), and incremental pressure correction (Francois et al., 2006) in which the pressure term of the previous time step is included in the computation of the provisional velocity.

B.1.1 Time integration

For the decoupling of pressure and velocity, we choose the approximate incremental projection method which consists of calculating first a provisional velocity using the previous pressure gradient, followed by a correction step in which the velocity is projected onto a divergence-free vector space. The overall time scheme is a corrector-predictor method.

In the method by Francois et al. (2006), the provisional velocity is first computed at the cell center according to [1]

$$\frac{\vec{u}_c^{*,0} - \vec{u}_c^n}{\Delta t} + [\vec{u} \cdot \nabla \vec{u}]_c^n = \left[\vec{g} + \vec{f}_\sigma - \frac{1}{\rho_f}\nabla p\right]_{f\to c}^{n-\frac{1}{2}} + \frac{1}{\rho_c^{n+\frac{1}{2}}}\left[\nabla \cdot \mu^{n+\frac{1}{2}}\left[(\nabla \vec{u}^{*,0}) + (\nabla \vec{u}^n)^T\right]\right]_c$$

where the superscript 0 refers to the corrector step of the time scheme, and the subscripts c and f denote respectively cell-centered and face-centered quantities. The pressure gradient and the surface tension term are computed at the cell faces (as in the staggered grid arrangement). These terms are kept together and computed with the same gradient operator, such as described by the balanced-force approach (Francois et al., 2006). The transfer from face-centered to cell-centered vector quantities is noted by $[.]_{f\to c}$, which is done by simple averaging in the direction of the

[1] Note that in the method by Francois et al. (2006), the conservative formulation of the momentum equation is used. As in Popinet (2009), and in agreement with many other studies on two-phase interface flows, we have chosen to use the non-conservative formulation.

respective components[2]. Next, the pressure and surface tension terms are substracted from the provisional velocity, which is then interpolated at the cell faces:

$$\vec{u}_c^{**} = \vec{u}_c^{*} - \Delta t \left[\vec{g} + \vec{f}_\sigma - \frac{1}{\rho_f} \nabla p \right]_{f \to c}^{n-\frac{1}{2}}$$
$$\vec{u}_f^{**} = \left[\vec{u}_c^{**} \right]_{c \to f}$$

The projection step is then defined by:

$$\nabla \cdot \left(\frac{1}{\rho_f^{n+\frac{1}{2}}} \nabla p^{n+\frac{1}{2},0} \right) = \nabla \cdot \left(\frac{\vec{u}_f^{**,0}}{\Delta t} + \vec{g} + \vec{f}_\sigma^{n+\frac{1}{2}} \right)$$
$$\vec{u}_c^{n+\frac{1}{2},0} = \vec{u}_c^{**,0} + \Delta t \left[\vec{g} + \vec{f}_\sigma^{n+\frac{1}{2}} - \frac{1}{\rho_f^{n+\frac{1}{2}}} \nabla p^{n+\frac{1}{2},0} \right]_{f \to c}$$
$$\vec{u}_f^{n+\frac{1}{2},0} = \vec{u}_f^{**,0} + \Delta t \left[\vec{g} + \vec{f}_\sigma^{n+\frac{1}{2}} - \frac{1}{\rho_f^{n+\frac{1}{2}}} \nabla p^{n+\frac{1}{2},0} \right]_{f}$$

This approach is called an approximate projection method, since the collocated velocity field $u_c^{n+1/2,0}$ is not exactly divergence free. However, the divergence of the staggered velocity field $u_f^{n+1/2,0}$ is forced to zero at each time step, and this keeps the continuity error to a minimum.

The corrector step of the overall time scheme consists of repeating the above steps, except that the updated pressure gradient is used and the explicit terms are replaced by a second-order approximation. The provisional velocity step is now:

$$\frac{\vec{u}_c^{*,1} - \vec{u}_c^{n}}{\Delta t} + \frac{1}{2} \left(\left[\vec{u} \cdot \nabla \vec{u} \right]_c^{n+1,0} + \left[\vec{u} \cdot \nabla \vec{u} \right]_c^{n} \right) = \left[\vec{g} + \vec{f}_\sigma^{n+\frac{1}{2}} - \frac{1}{\rho_f^{n+\frac{1}{2}}} \nabla p^{n+\frac{1}{2},0} \right]_{f \to c}$$
$$+ \frac{1}{\rho_c^{n+\frac{1}{2}}} \left[\nabla \cdot \mu^{n+\frac{1}{2}} \left[(\nabla \vec{u}^{*,1}) + \tfrac{1}{2} (\nabla \vec{u}^{n+1,0})^T + \tfrac{1}{2} (\nabla \vec{u}^{n})^T \right] \right]_c$$
$$\vec{u}_c^{**,1} = \vec{u}_c^{*,1} - \Delta t \left[\vec{g} + \vec{f}_\sigma^{n+\frac{1}{2}} - \frac{1}{\rho_f^{n+\frac{1}{2}}} \nabla p^{n+\frac{1}{2},0} \right]_{f \to c}$$
$$\vec{u}_f^{**,1} = \left[\vec{u}_c^{**,1} \right]_{c \to f}$$

[2]For example, for a variable q defined at the cell faces, $[q]_{f \to c}$ is defined by $q_{i,j} = (q_{i+1/2,j} + q_{i-1/2,j})/2$ or $q_{i,j} = (q_{i,j+1/2} + q_{i,j-1/2})/2$. Likewise, the transfer from cell-centered to face-centered quantities is noted by $[q]_{c \to f}$, we use $q_{i+1/2,j} = (q_{i+1,j} + q_{i,j})/2$ or $q_{i,j+1/2} = (q_{i,j+1} + q_{i,j})/2$.

The final projection step is:

$$\nabla \cdot \left(\frac{1}{\rho_f^{n+\frac{1}{2}}} \nabla p^{n+\frac{1}{2}} \right) = \nabla \cdot \left(\frac{\vec{u}_f^{**,1}}{\Delta t} + \vec{g} + \vec{f}_\sigma^{n+\frac{1}{2}} \right)$$

$$\vec{u}_c^{n+\frac{1}{2}} = \vec{u}_c^{**,1} + \Delta t \left[\vec{g} + \vec{f}_\sigma^{n+\frac{1}{2}} - \frac{1}{\rho_f^{n+\frac{1}{2}}} \nabla p^{n+\frac{1}{2}} \right]_{f \to c}$$

$$\vec{u}_f^{n+\frac{1}{2}} = \vec{u}_f^{**,1} + \Delta t \left[\vec{g} + \vec{f}_\sigma^{n+\frac{1}{2}} - \frac{1}{\rho_f^{n+\frac{1}{2}}} \nabla p^{n+\frac{1}{2}} \right]_{f}$$

The interface is defined at intermediate time levels, $t^{n+\frac{1}{2}}$, such that fluid properties and the interface curvature can be expressed directly at this instant. The volume fraction C is advanced first from $C^{n-\frac{1}{2}}$ to $C^{n+\frac{1}{2}}$ using the velocity field $\vec{u}_f^n$. It is important at this point to use the divergence-free cell-face velocity, in order to guarantee conservation of the volume fractions. The interface advection step is done only once each time step. The density, viscosity and surface tension term at $t^{n+\frac{1}{2}}$ is therefore the same for both the predictor and corrector step.

The pressure gradient and velocity divergence are computed on the staggered grid. Therefore, the same standard discrete gradient and divergence operators can be used as those given in §4.3.1. Also, the surface tension term is needed at the cell face, as is the case for the staggered flow solver. For a description of this term, the reader is referred to §4.3.6.

B.1.2 Advection terms

The advection terms in the momentum equations are discretized using the second-order ENO scheme:

$$[\vec{u} \cdot \nabla u]_{i,j}^u = u_{i,j} \left[\frac{\partial u}{\partial x} \right]_{i,j}^{\text{ENO}} + v_{i,j} \left[\frac{\partial u}{\partial y} \right]_{i,j}^{\text{ENO}}$$

$$[\vec{u} \cdot \nabla v]_{i,j}^v = u_{i,j} \left[\frac{\partial v}{\partial x} \right]_{i,j}^{\text{ENO}} + v_{i,j} \left[\frac{\partial v}{\partial y} \right]_{i,j}^{\text{ENO}}$$

The advection terms are upwind approximations determined from :

$$u \left[\frac{\partial \phi}{\partial x} \right]^{\text{ENO}} = \begin{cases} u\phi_x^- & \text{if } u \geq 0 \\ u\phi_x^+ & \text{if } u < 0 \end{cases}$$

The spatial derivatives ϕ_x^- and ϕ_x^+ are calculated using the second-order ENO scheme. Defining,

$$\begin{aligned} D\phi_i^- &= \phi_i - \phi_{i-1} \\ D\phi_i^+ &= \phi_{i+1} - \phi_i \\ D^2\phi_i^- &= \phi_i - 2\phi_{i-1} + \phi_{i-2} \\ D^2\phi_i^0 &= \phi_{i+1} - 2\phi_i + \phi_{i-1} \\ D^2\phi_i^+ &= \phi_{i+2} - 2\phi_{i+1} + \phi_i \end{aligned}$$

the ENO scheme is formulated as

$$\begin{aligned}\phi_x^- &= \frac{1}{h}D\phi^- + \frac{1}{2h}\text{minmod}\left(D^2\phi^0, D^2\phi^-\right)\\ \phi_x^+ &= \frac{1}{h}D\phi^+ - \frac{1}{2h}\text{minmod}\left(D^2\phi^0, D^2\phi^+\right)\end{aligned}$$

where

$$\text{minmod}(a,b) = \begin{array}{ll} a & \text{if } |a|<|b| \text{ and } ab>0\\ b & \text{if } |b|<|a| \text{ and } ab>0\\ 0 & \text{otherwise}\end{array}$$

B.1.3 Viscous terms

The implicit viscous terms are discretized as:

$$\begin{aligned}\frac{1}{\rho_{i,j}}\left[\nabla\cdot\mu(\nabla u)\right]_{i,j} &= \frac{1}{\rho_{i,j}\Delta x}\left[\mu_{i+\frac{1}{2},j}\frac{u_{i+1,j}-u_{i,j}}{\Delta x} - \mu_{i-\frac{1}{2},j}\frac{u_{i,j}-u_{i-1,j}}{\Delta x}\right]\\ &+ \frac{1}{\rho_{i,j}\Delta y}\left[\mu_{i,j+\frac{1}{2}}\frac{u_{i,j+1}-u_{i,j}}{\Delta y} - \mu_{i,j-\frac{1}{2}}\frac{u_{i,j}-u_{i,j-1}}{\Delta y}\right]\\ \frac{1}{\rho_{i,j}}\left[\nabla\cdot\mu(\nabla v)\right]_{i,j} &= \frac{1}{\rho_{i,j}\Delta x}\left[\mu_{i+\frac{1}{2},j}\frac{v_{i+1,j}-v_{i,j}}{\Delta x} - \mu_{i-\frac{1}{2},j}\frac{v_{i,j}-v_{i-1,j}}{\Delta x}\right]\\ &+ \frac{1}{\rho_{i,j}\Delta y}\left[\mu_{i,j+\frac{1}{2}}\frac{v_{i,j+1}-v_{i,j}}{\Delta y} - \mu_{i,j-\frac{1}{2}}\frac{v_{i,j}-v_{i,j-1}}{\Delta y}\right]\end{aligned}$$

The explicit terms in the equations for u and v are calculated from respectively:

$$\begin{aligned}\frac{1}{\rho_{i,j}}\left[\nabla\cdot\mu(\nabla\vec{u})^T\right]^u_{i,j} &= \frac{1}{\rho_{i,j}\Delta x}\left[\mu_{i+\frac{1}{2},j}\frac{u_{i+1,j}-u_{i,j}}{\Delta x} - \mu_{i-\frac{1}{2},j}\frac{u_{i,j}-u_{i-1,j}}{\Delta y}\right]\\ &+ \frac{1}{\rho_{i,j}\Delta y}\left[\mu_{i,j+\frac{1}{2}}\frac{\overline{v}_{i+1,j+\frac{1}{2}}-\overline{v}_{i-1,j+\frac{1}{2}}}{2\Delta x} - \mu_{i,j-\frac{1}{2}}\frac{\overline{v}_{i+1,j-\frac{1}{2}}-\overline{v}_{i-1,j-\frac{1}{2}}}{2\Delta x}\right]\\ \frac{1}{\rho_{i,j}}\left[\nabla\cdot\mu(\nabla\vec{u})^T\right]^v_{i,j} &= \frac{1}{\rho_{i,j}\Delta x}\left[\mu_{i+\frac{1}{2},j}\frac{\overline{u}_{i+\frac{1}{2},j+1}-\overline{u}_{i+\frac{1}{2},j-1}}{2\Delta y} - \mu_{i-\frac{1}{2},j}\frac{\overline{u}_{i-\frac{1}{2},j+1}-\overline{u}_{i-\frac{1}{2},j-1}}{2\Delta y}\right]\\ &+ \frac{1}{\rho_{i,j}\Delta y}\left[\mu_{i,j+\frac{1}{2}}\frac{v_{i,j+1}-v_{i,j}}{\Delta y} - \mu_{i,j-\frac{1}{2}}\frac{v_{i,j}-v_{i,j-1}}{\Delta y}\right]\end{aligned}$$

where

$$\begin{aligned}\overline{u}_{i+\frac{1}{2},j} &= \frac{1}{2}\left(u_{i+1,j}+u_{i,j}\right)\\ \overline{v}_{i,j+\frac{1}{2}} &= \frac{1}{2}\left(v_{i,j+1}+v_{i,j}\right)\end{aligned}$$

The viscosity at the cell faces is defined as:

$$\begin{aligned}\mu_{i+1/2,j} &= \mu_g + (\mu_l-\mu_g)C_{i+1/2,j}\\ \mu_{i,j+1/2} &= \mu_g + (\mu_l-\mu_g)C_{i,j+1/2}\end{aligned}$$

where

$$\begin{aligned}C_{i+1/2,j} &= (C_{i,j}+C_{i+1,j})/2\\ C_{i,j+1/2} &= (C_{i,j}+C_{i,j+1})/2\end{aligned}$$

The density at the cell faces is defined in a similar manner:

$$\rho_{i+1/2,j} = \rho_g + (\rho_l - \rho_g)C_{i+1/2,j}$$
$$\rho_{i,j+1/2} = \rho_g + (\rho_l - \rho_g)C_{i,j+1/2}$$

B.2 Capillary wave

B.2.1 Uniform viscosity and density

Table B.1: Capillary standing wave for fluids with uniform density and viscosity: RMS error of the temporal evolution of the interface height as a function of grid size. $\mathrm{CFL}_\sigma = \mathrm{CFL} = 0.5$. The level-set is reinitialized after each advection step.

Grid	8	16	32	64	128
LS-LA1	0.2251	0.0966	0.0242	0.0058	0.0022
LS-LA2	0.2179	0.0838	0.0207	0.0049	0.0018
LS-GFM1	0.2307	0.1095	0.0287	0.0070	0.0023
LS-CSF	0.2447	0.1295	0.0391	0.0101	0.0030
LS-GFMk	0.2326	0.1098	0.0288	0.0070	0.0023
LS-GFMϵ	0.2171	0.0835	0.0206	0.0049	0.0018
VOF-WYC	0.1483	0.0295	0.0090	0.0040	0.0019
VOF-HF1	0.0549	0.0246	0.0068	0.0016	0.0011
VOF-WY	0.0549	0.0246	0.0068	0.0016	0.0011
MOF-HF1	0.0509	0.0282	0.0090	0.0025	0.0013
VOF-HF2	0.0541	0.0253	0.0069	0.0016	0.0011
Gerris	0.1591	0.0316	0.0077	0.0022	0.0005

B.2.2 Air-water capillary wave

Table B.2: Capillary standing wave for water-air: RMS error on the interface height evolution as a function of grid size. $\mathrm{CFL}_\sigma = \mathrm{CFL} = 0.5$. The level-set is reinitialized after each advection step. No smoothing kernel is used with the VOF methods.

Grid	8	16	32	64	128
VOF-HF1	0.1798	0.0531	0.0160	0.0060	0.0033
VOF-WY	0.1798	0.0531	0.0160	0.0060	0.0033
VOF-HF2	0.1827	0.0632	0.0285	0.0172	0.0105
VOF-WYC	0.2062	0.1101	0.0375	0.0132	0.0058
MOF-HF1	21.8143	0.0526	1.4594	0.0957	0.0932
LS-CSF	0.1570	0.0800	0.0451	0.0174	0.0039
LS-LA1	0.2521	0.0865	0.0232	0.0104	0.0056
LS-LA2	0.2472	0.0655	0.0219	0.0112	0.0061
LS-GFMk	0.3090	0.1978	0.1125	0.0627	0.0315
LS-GFMϵ	0.3037	0.1811	0.1035	0.0598	0.0308
LS-GFM1	0.3002	0.1768	0.0600	0.0194	0.0049
Gerris	0.2041	0.0795	0.0191	0.0069	0.00313

B.2.2.1 Influence of the smoothing kernel

The MOF-HF1 method does not work very well for this test case since the density ratio is large, and the smoothing kernel does not enhance the results significantly. For the VOF-HF1 and VOF-WY methods, it enhances the results for coarse grids only there is a net enhancement with the VOF-HF2 and VOF-WYC methods. The Vertex-Avg smoothing kernel seems to be the one yielding the best results. In the Gerris software (Popinet), the smoothing of the density and viscosity is slightly more accurate on fine grids.

Table B.3: Capillary standing wave for water-air: influence of the smoothing kernel on the VOF methods. Measure of the RMS error on the interface height evolution as a function of grid size with $\mathrm{CFL}_\sigma = \mathrm{CFL} = 0.5$.

method	kernel /Grid	8	16	32	64	128
VOF-HF2	none	0.1827	0.0632	0.0285	0.0172	0.0105
	K_8	0.1674	0.0491	0.0271	0.0169	0.0098
	δ_ϵ	0.1751	0.0586	0.0283	0.0171	0.0102
	Vertex-Avg	0.1502	0.0271	0.0241	0.0163	0.0091
MOF-HF1	none	21.8143	0.0526	1.4594	0.0957	0.0932
	K_8	0.8650	0.0426	0.0160	0.3167	0.2928
	δ_ϵ	3.1895	0.0501	0.0166	0.1972	0.1065
	Vertex-Avg	0.1391	0.0262	0.3697	0.4991	0.3467
VOF-WYC	none	0.2062	0.1101	0.0375	0.0132	0.0058
	K_8	0.2049	0.0826	0.0238	0.0091	0.0045
	δ_ϵ	0.2027	0.0870	0.0264	0.0099	0.0048
	Vertex-Avg	0.2028	0.0690	0.0213	0.0088	0.0044
Gerris	none	0.1997	0.1191	0.0330	0.0095	0.0070
	Vertex-Avg	0.2035	0.0791	0.0233	0.0074	0.0033
	Vertex-Avg (ρ & μ)	0.2041	0.0795	0.0191	0.0069	0.0035

B.3 Rising bubble benchmark

B.3.1 Benchmark case 1

Table B.4: Rising bubble benchmark 1: minimum circularity $\mathbf{c}_{\min}$, maximum velocity $\mathbf{v}_{\max}$ and their time of occurrence. Final centroid position $\mathbf{Y_c}$ and relative mass error $\mathbf{E_m}$.

Method	Grid	$\mathbf{c}_{\min}$	$\mathbf{t}\vert_{\mathbf{c}=\mathbf{c}_{\min}}$	$\mathbf{v}_{\max}$	$\mathbf{t}\vert_{\mathbf{v}=\mathbf{v}_{\max}}$	Yc	$\mathbf{E_m}(\%)$
VOF-HF1	32	0.8906	1.8430	0.2412	0.9659	1.0836	0
	64	0.8980	1.8588	0.2416	0.9266	1.0816	0
	128	0.9002	1.8685	0.2417	0.9248	1.0815	0
	256	0.9006	1.8840	0.2417	0.9230	1.0816	0
MOF-HF1	32	0.8775	2.1041	0.2413	0.9607	1.0835	0
	64	0.8909	2.2686	0.2417	0.9266	1.0816	0
	128	0.8932	2.1387	0.2417	0.9248	1.0815	0
	256	0.9000	1.9082	0.2417	0.9225	1.0816	0
VOF-WY	32	0.8910	1.8430	0.2412	0.9659	1.0836	0
	64	0.8981	1.8588	0.2416	0.9266	1.0816	0
	128	0.9002	1.8685	0.2417	0.9248	1.0815	0
	256	0.9006	1.8840	0.2417	0.9232	1.0816	0
VOF-WYC	32	0.8711	2.0988	0.2407	0.9659	1.0781	0
	64	0.8975	1.9585	0.2415	0.9340	1.0802	0
	128	0.8999	1.8724	0.2417	0.9261	1.0813	0
	256	0.9006	1.8849	0.2417	0.9230	1.0817	0
VOF-HF2	32	0.8780	1.9318	0.2411	0.9659	–	0
	64	0.8977	1.9013	0.2414	0.9285	1.0804	0
	128	0.9003	1.8698	0.2416	0.9248	1.0806	0
	256	0.9006	1.8847	0.2416	0.9223	1.0809	0
LS-CSF	32	0.9214	1.8900	0.2391	0.9450	1.0901	4.34
	64	0.9075	1.8404	0.2411	0.9285	1.0834	1.80
	128	0.9030	1.8828	0.2416	0.9235	1.0818	0.62
	256	0.9016	1.9100	0.2417	0.9220	1.0817	0.25
LS-LA1	32	0.9256	1.7386	0.2382	0.9241	1.0953	4.49
	64	0.9115	1.8367	0.2410	0.9230	1.0860	1.95
	128	0.9056	1.8730	0.2416	0.9215	1.0830	0.77
	256	0.9031	1.8881	0.2417	0.9202	1.0823	0.31
LS-LA2	32	0.9257	1.7386	0.2385	1.0129	1.0963	4.90
	64	0.9113	1.8385	0.2409	0.9119	1.0863	1.99
	128	0.9053	1.8743	0.2415	0.9235	1.0827	0.75
	256	0.9030	1.8886	0.2416	0.9181	1.0820	0.30
LS-GFMϵ	32	0.9194	1.7647	0.2387	0.9346	1.0879	4.51
	64	0.9067	1.8385	0.2415	0.9211	1.0822	1.83
	128	0.9026	1.8841	0.2422	0.9176	1.0813	0.61
	256	0.9011	1.8692	0.2424	0.9176	1.0817	0.27
LS-GFMk	32	0.9187	1.7751	0.2390	0.9346	1.0931	4.86
	64	0.9069	1.8330	0.2417	0.9211	1.0832	2.01
	128	0.9026	1.8835	0.2423	0.9176	1.0812	0.71
	256	0.9011	1.8514	0.2424	0.9172	1.0814	0.31
LS-GFM1	32	0.9250	1.7386	0.2387	0.9398	1.0978	4.94
	64	0.9098	1.8404	0.2410	0.9285	1.0866	1.87
	128	0.9044	1.8704	0.2416	0.9202	1.0835	0.66
	256	0.9023	1.8690	0.2417	0.9218	1.0827	0.25
	ref	0.9013	1.8999	0.2417	0.9239	1.0819	0

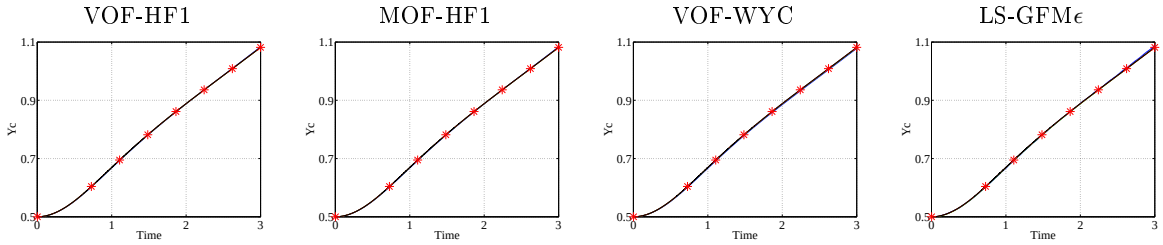

Figure B.1: Rising bubble benchmark 1: Position of the bubble centroid as a function of time on different grid sizes. Blue (32×64), green (64×128), red (128×256), black (256×512) and red stars (reference solution).

B.3.2 Benchmark case 2

Table B.5: Rising bubble benchmark 2: minimum circularity $\mathbf{c}_{\min}$, maximum velocity $\mathbf{v}_{\max}$ and their time of occurrence, final centroid position $\mathbf{Y_c}$ and relative mass error $\mathbf{E_m}$.

Method	Grid	$\mathbf{c}_{\min}$	$\mathbf{t}\|_{\mathbf{c}=\mathbf{c}_{\min}}$	$\mathbf{v}_{\max}$	$\mathbf{t}\|_{\mathbf{v}=\mathbf{v}_{\max}}$	Yc	$\mathbf{E_m}(\%)$
LS-CSF	32	0.6029	3.0000	0.2507	0.7396	1.0941	6.42
	64	0.5479	3.0000	0.2505	0.7373	1.1007	7.45
	128	0.5482	2.9777	0.2503	0.7300	1.1196	3.50
	256	0.0837	2.6359	0.2503	0.7300	0.9445	6.41
LS-GFMk	32	0.7655	1.9194	0.2528	0.7396	1.1429	7.12
	64	0.7242	1.9971	0.2514	0.7520	1.1357	1.95
	128	0.7036	2.0398	0.2518	0.7300	1.1378	0.76
	256	0.6959	2.1030	0.2518	0.7266	1.1383	0.24
LS-GFMϵ	32	0.7652	1.9546	0.2530	0.7396	1.1439	7.22
	64	0.7231	1.9971	0.2515	0.7568	1.1374	2.08
	128	0.7034	2.0398	0.2518	0.7300	1.1381	0.87
	256	0.6959	2.1033	0.2518	0.7263	1.1384	0.20
LS-GFM1	32	0.7596	1.8841	0.2506	0.7396	1.1408	7.47
	64	0.5812	2.8413	0.2502	0.7324	1.1112	3.82
	128	0.7096	1.9958	0.2502	0.7288	1.1309	2.37
	256	0.6226	2.7789	0.2502	0.7285	1.1325	2.60
LS-LA1	32	0.7525	1.9370	0.2463	0.6867	1.1261	8.55
	64	0.7234	1.9385	0.2494	0.7129	1.1322	3.03
	128	0.7095	1.9910	0.2498	0.7202	1.1360	0.68
	256	0.5139	2.3224	0.2500	0.7263	1.1371	0.22
LS-LA2	32	0.7532	1.9017	0.2463	0.6867	1.1248	9.04
	64	0.7229	1.9385	0.2494	0.7129	1.1334	3.74
	128	0.7092	1.9910	0.2499	0.7202	1.1361	0.97
	256	0.7003	2.0511	0.2500	0.7266	1.1376	0.04
VOF-WYC	32	0.4700	3.0000	0.2470	0.7396	1.0793	0
	64	0.4581	3.0000	0.2497	0.7324	1.1001	0
	128	0.4687	3.0000	0.2502	0.7324	1.1189	0
	256	0.4823	3.0000	0.2502	0.7306	1.1280	0
VOF-WY	32	0.4956	2.9799	0.2513	0.7220	1.1148	0
	64	0.5219	3.0000	0.2505	0.7324	1.1270	0
	128	0.5555	2.9944	0.2503	0.7275	1.1324	0
	256	0.5688	2.9999	0.2502	0.7288	1.1353	0
VOF-HF1	32	0.4985	2.9976	0.2514	0.7220	1.1148	0
	64	0.5142	2.9980	0.2505	0.7324	1.1270	0
	128	0.5650	2.9517	0.2503	0.7275	1.1327	0
	256	0.5818	2.4725	0.2502	0.7285	1.1355	0
MOF-HF1	32	0.4785	2.9115	0.2516	0.7220	1.1104	0
	64	0.5205	2.9932	0.2506	0.7324	1.1255	0
	128	0.5207	2.9980	0.2503	0.7275	1.1326	0
	256	0.5286	2.8394	0.2502	0.7285	1.1353	0
	ref	0.5144	2.9994	0.2502	0.7316	1.1376	0

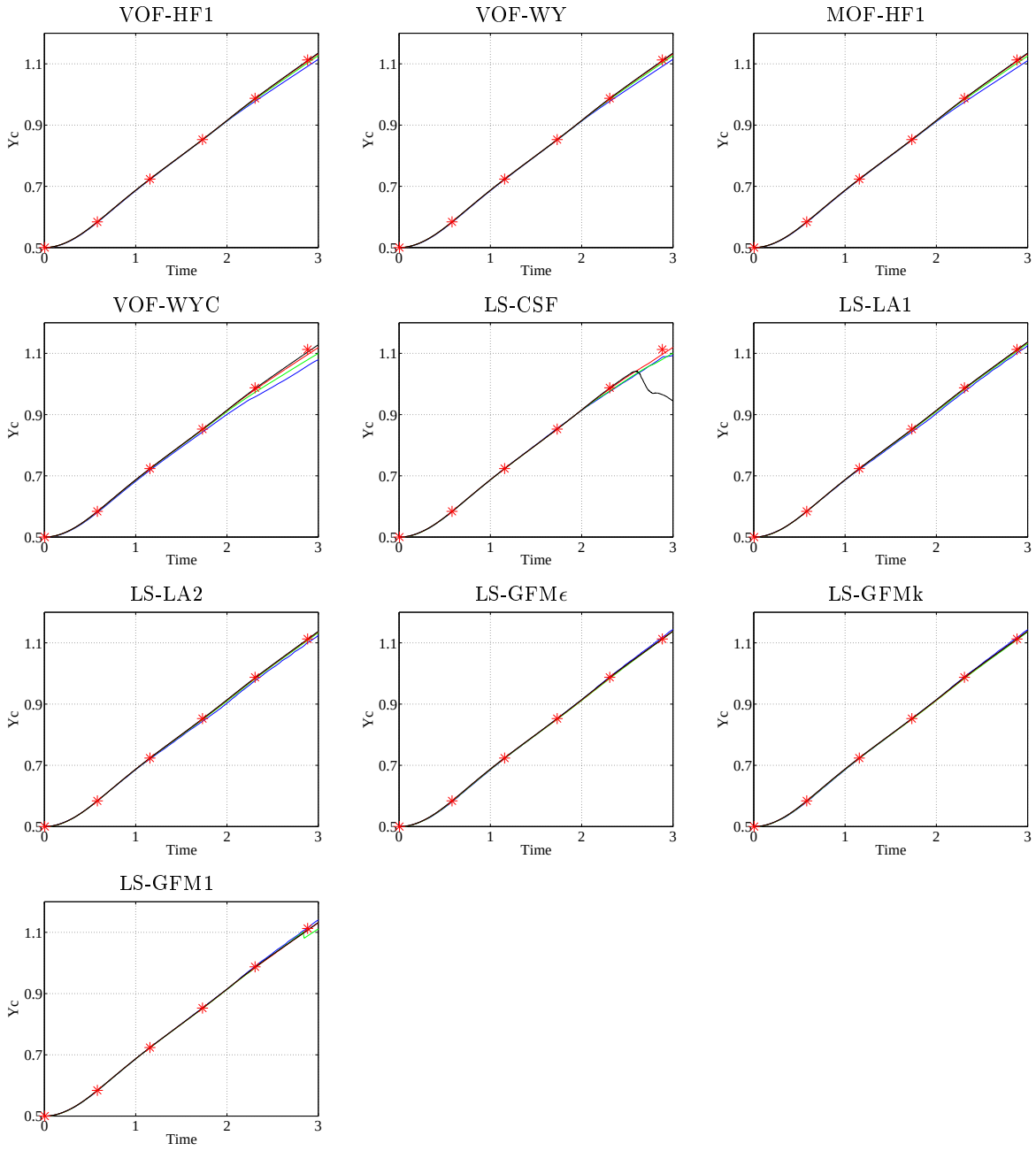

Figure B.2: Rising bubble benchmark 2: Position of the bubble centroid as a function of time on different grid sizes. Blue (32×64), green (64×128), red (128×256), black (256×512) and red stars (reference solution).

Bibliography

S. Aland and A. Voigt. Benchmark computations of diffuse interface models for two-dimensional bubble dynamics. *Int. J. Numer. Meth. Fluids*, 69(3):747–761, 2012.

A. Albadawi, D. B. Donoghue, A.J. Robinson, D. B. Murray, and Y. M. C. Delauré. Influence of surface tension implementation in volume of fluid and coupled volume of fluid with level set methods for bubble growth and detachment. *Int. J. Multiphase Flow*, 53:11–28, 2013.

A. S. Almgren, J. B. Bell, and W. G. Szymczak. A numerical method for the incompressible navier-stokes equations based on an approximate projection. *SIAM J. Sci. Comput.*, 17(2), 1996.

A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome. A conservative adaptive projection method for the variable density incompressible navier-stokes equations. *J. Comput. Phys.*, 142:1–46, 1998.

E. Aulisa, S. Manservisi, and R. Scardovelli. A novel representation of the surface tension force for two-phase flow with reduced spurious currents. *Computer Methods in Applied Mechanics and Engineering*, 195(44):6239–6257, 2006.

E. Aulisa, S. Manservisi, R. Scardovelli, and S. Zaleski. Interface reconstruction with least-squares fit and split advection in three-dimensional cartesian geometry. *J. Comput. Phys.*, 225 (2):2301–2319, 2007.

J. B. Bell and D. L. Marcus. A second-order projection method for variable-density flows. *J. Comput. Phys.*, 101(2):334–348, 1992.

J. B. Bell, P. Colella, and H. M. Glaz. A second-order projection method for the incompressible navier-stokes equations. *J. Comput. Phys.*, 85(2):257–283, 1989.

R. E. Bellman and R. H. Pennington. Effects of surface tension and viscosity on taylor instability. 1953.

G. Bornia, A. Cervone, S. Manservisi, R. Scardovelli, and S. Zaleski. On the properties and limitations of the height function method in two-dimensional cartesian geometry. *J. Comput. Phys.*, 230(4):851–862, 2011.

A. Bourlioux. A coupled level-set volume-of-fluid algorithm for tracking material interfaces. In *Proceedings of the 6th International Symposium on Computational Fluid Dynamics, Lake Tahoe, CA*, volume 15, 1995.

J. U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *J. Comput. Phys.*, 100(2):335–354, 1992.

V. V. Buwa, D. Gerlach, F. Durst, and E. Schlücker. Numerical simulations of bubble formation on submerged orifices: Period-1 and period-2 bubbling regimes. *Chemical Engineering Science*, 62(24):7119–7132, 2007.

I. Chakraborty, G. Biswas, and P. S. Ghoshdastidar. Bubble generation in quiescent and co-flowing liquids. *Int. J. Heat & Mass Transfer*, 54(21):4673–4688, 2011.

I. Chakraborty, G. Biswas, and P. S. Ghoshdastidar. A coupled level-set and volume-of-fluid method for the buoyant rise of gas bubbles in liquids. *Int. J. Heat & Mass Transfer*, 58(1): 240–259, 2013.

Y. C. Chang, T. Y. Hou, B. Merriman, and S. Osher. A level set formulation of Eulerian interface capturing methods for incompressible fluid flows. *J. Comput. Phys.*, 124(2):449–464, 1996.

D. L. Chopp. Some improvements of the fast marching method. *SIAM J. Sci. Comput.*, 23(1): 230–244, 2001.

A.J. Chorin. Numerical solution of the navier–stokes equations. *Math. Comput.*, 22:745–762, 1986.

L. Cortelezzi and A. Prosperetti. Small-ampltitude waves on the surface of a layer of a viscous-liquid. *Quarterly of Applied Mathematics*, 38(4):375–389, 1981.

J.M. Cubos-Ramírez, J. Ramírez-Cruz, M. Salinas-Vázquez, W. Vicente-Rodríguez, E. Martinez-Espinosa, and C. Lagarza-Cortes. Efficient two-phase mass-conserving level set method for simulation of incompressible turbulent free surface flows with large density ratio. *Computers & Fluids*, 2016.

S. J. Cummins, M. M. Francois, and D. B. Kothe. Estimating curvature from volume fractions. *Computers & structures*, 83(6):425–434, 2005.

B. J. Daly. Numerical study of the effect of surface tension on interface instability. *Phys. Fluids*, 12(7):1340–1354, 1969.

F. Denner and B. G.-M. van Wachem. On the convolution of fluid properties and surface force for interface capturing methods. *Int. J. Multiphase Flow*, 54:61–64, 2013.

F. Denner and B. G.-M. van Wachem. Numerical time-step restrictions as a result of capillary waves. *J. Comput. Phys.*, 285:24–40, 2015.

F. Denner, D. R. van der Heul, G. T. Oud, M. M. Villar, A. da Silveira Neto, and B. G.-M. van Wachem. Comparative study of mass-conserving interface capturing frameworks for two-phase flows with surface tension. *Int. J. Multiphase Flow*, 61:37–47, 2014b.

O. Desjardins and H. Pitsch. A spectrally refined interface approach for simulating multiphase flows. *J. Comput. Phys.*, 228(5):1658–1677, 2009.

O. Desjardins, V. Moureau, and H. Pitsch. An accurate conservative level set/ghost fluid method for simulating turbulent atomization. *J. Comput. Phys.*, 227(18):8395–8416, 2008.

V. Dyadechko and M. Shashkov. Moment-of-fluid interface reconstruction. *Los Alamos report LA-UR-05-7571*, 2005.

V. Dyadechko and M. Shashkov. Multi-material interface reconstruction from the moment data. *Los Alamos National Laboratory, Report LA-UR-06-5846*, 2006.

V. Dyadechko and M. Shashkov. Reconstruction of multi-material interfaces from moment data. *J. Comput. Phys.*, 227(11):5361–5384, 2008.

D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. *J. Comput. Phys.*, 183(1):83–116, November 2002.

J.-L. Estivalezes. Une formulation conservative du schéma weno pour améliorer la résolution de la méthode level-set. Technical report, ONERA Toulouse, 2002.

Featflow. Data are available at http://www.featflow.de/en/benchmarks/cfdbenchmarking/bubble/bubble_reference.html - july 2016.

R.P. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost of fluid method). *J. Comput. Phys.*, 152(2):457–492, 1999.

R.P. Fedkiw, B. Merriman, and S. Osher. Simplified discretization of systems of hyperbolic conservation laws containing advection equations. *J. Comput. Phys.*, 157(1):302–326, January 2000.

M.M. Francois, S.J. Cummins, E.D. Dendy, D.B. Kothe, J.M. Sicilian, and M.W. Williams. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *J. Comput. Phys.*, 213(1):141–173, 2006.

D. E. Fyfe, E. S. Oran, and M. J. Fritts. Surface tension and viscosity with lagrangian hydrodynamics on a triangular mesh. *J. Comput. Phys.*, 76(2):349–384, 1988.

D. Gerlach, G. Tomar, G. Biswas, and F. Durst. Comparison of volume-of-fluid methods for surface tension-dominant two-phase flows. *Int. J. Heat & Mass Transfer*, 49(3-4):740–754, 2006.

D. Gerlach, N. Alleborn, V. Buwa, and F. Durst. Numerical simulation of periodic bubble formation at a submerged orifice with constant gas flow rate. *Chemical Engineering Science*, 62(7):2109–2125, 2007.

F. Gibou, R.P. Fedkiw, L.-T. Cheng, and M. Kang. A second-order-accurate symmetric discretization of the Poisson equation on irregular domains. *J. Comput. Phys.*, 176(1):205–227, 2002.

G. Golub and C. Loan. *Matrix computations*. The John Hopkins University Press, 1989.

P. Gómez, J. Hernandez, and J. López. On the reinitialization procedure in a narrow-band locally refined level set method for interfacial flows. *Int. J. Numer. Meth. Eng.*, 63(10):1478–1512, 2005.

D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, and S. Zaleski. Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *J. Comput. Phys.*, 152(2):423–456, 1999.

D.Z. Guo, D.L. Sun, Z.Y. Li, and W.Q. Tao. Phase change heat transfer simulation for boiling bubbles arising from a vapor film by the voset method. *Numerical Heat Transfer, Part A: Applications*, 59(11):857–881, 2011.

A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, iii. *J. Comput. Phys.*, 71(2):231–303, 1987.

J. Helmsen, P. Colella, and E. G. Puckett. Non-convex profile evolution in two dimensions using volume of fluids. *Lawrence Berkeley National Laboratory report LBNL-40693*, 1997.

J. Hernández, J. López, P. Gómez, C. Zanzi, and F. Faura. A new volume of fluid method in three dimensions — part i: Multidimensional advection method with face-matched flux polyhedra. *Int. J. Numer. Meth. Fluids*, 58(8):897–921, 2008.

M. Herrmann. A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids. *J. Comput. Phys.*, 227(4):2674–2706, 2008.

C. W. Hirt and B. D. Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *J. Comput. Phys.*, 39(1):201–225, January 1981.

J. Hua and J. Lou. Numerical simulation of bubble rising in viscous liquid. *J. Comput. Phys.*, 222(2):769–795, 2007.

J. Hua, J. F. Stene, and P. Lin. Numerical simulation of 3d bubbles rising in viscous liquids using a front tracking method. *J. Comput. Phys.*, 227(6):3358–3382, 2008.

S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, and L. Tobiska. Quantitative benchmark computations of two-dimensional bubble dynamics. *Int. J. Numer. Meth. Fluids*, 60(11):1259–1288, 2009.

M. Jemison, E. Loch, M. Sussman, M. Shashkov, M. Arienti, M. Ohta, and Y. Wang. A coupled level set-moment of fluid method for incompressible two-phase flows. *J. Scientific Computing*, 54(2-3):454–491, 2013.

G.-S. Jiang and D. Peng. Weighted ENO schemes for Hamilton-Jacobi equations. *SIAM J. Sci. Comput.*, 21:2126–2143, 2000.

G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted eno schemes. Technical report, DTIC Document, 1995.

G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted eno schemes. *J. Comput. Phys.*, 126(1):202–228, 1996.

M. Kang, R.P. Fedkiw, and X.-D. Liu. A boundary condition capturing method for multiphase incompressible flow. *J. Scientific Computing*, 15(3):323–360, September 2000.

R. Knikker. Study of a staggered fourth-order compact scheme for unsteady incompressible viscous flows. *Int. J. Numer. Methods Fluids*, 59(10):1063–1092, 2009.

M. Kwakkel, W.-P. Breugem, and B. J. Boersma. Extension of a clsvof method for droplet-laden flows with a coalescence/breakup model. *J. Comput. Phys.*, 253:166–188, 2013.

B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti. Modelling merging and fragmentation in multiphase flows with surfer. *J. Comput. Phys.*, 113(1):134–147, 1994.

J. Li. Calcul d'interface affine par morceaux. *Comptes rendus de l'Académie des sciences. Série II, Mécanique, physique, chimie, astronomie*, 320(8):391–396, 1995.

Q. Li, J. Ouyang, B. Yang, and X. Li. Numerical simulation of gas-assisted injection molding using clsvof method. *Applied Mathematical Modelling*, 36(5):2262–2274, 2012.

K. Ling, Zeng-Yao Li, and Wen-Quan Tao. A direct numerical simulation for nucleate boiling by the voset method. *Numerical Heat Transfer, Part A: Applications*, 65(10):949–971, 2014.

K. Ling, Z.-H. Li, D.-L. Sun, Y.-L. He, and W.-Q. Tao. A three-dimensional volume of fluid & level set (voset) method for incompressible two-phase flow. *Computers & Fluids*, 118:293–304, 2015.

X.-D. Liu, R. P. Fedkiw, and M. Kang. A boundary condition capturing method for poisson's equation on irregular domains. *J. Comput. Phys.*, 160(1):151–178, May 2000.

Xu-Dong Liu, Stanley Osher, and Tony Chan. Weighted essentially non-oscillatory schemes. *J. Comput. Phys.*, 115(1):200–212, 1994.

J. López and J. Hernández. On reducing interface curvature computation errors in the height function technique. *J. Comput. Phys.*, 229(13):4855–4868, 2010.

J. López, J. Hernández, P. Gómez, and F. Faura. A volume of fluid method based on multidimensional advection and spline interface reconstruction. *J. Comput. Phys.*, 195(2):718–742, 2004.

J. López, C. Zanzi, P. Gómez, R. Zamora, F. Faura, and J. Hernández. An improved height function technique for computing interface curvature from volume fractions. *Computer Methods in Applied Mechanics and Engineering*, 198(33):2555–2564, 2009.

E. Marchandise, P. Geuzaine, N. Chevaugeon, and J.-F. Remacle. A stabilized finite element method using a discontinuous level set approach for the computation of bubble dynamics. *J. Comput. Phys.*, 225(1):949–974, 2007.

T. Menard, S. Tanguy, and A. Berlemont. Coupling level set/vof/ghost fluid methods: Validation and application to 3d simulation of the primary break-up of a liquid jet. *Int. J. Multiphase Flow*, 33(5):510–524, May 2007.

A. Mukherjee and S. G. Kandlikar. Numerical study of single bubbles with dynamic contact angle during nucleate pool boiling. *Int. J. Heat & Mass Transfer*, 50(1):127–138, 2007.

Y. Ng, H. Chen, C. Min, and F. Gibou. Guidelines for Poisson solvers on irregular domains with Dirichlet boundary conditions using the Ghost Fluid Method. *J. Scientific Computing*, 41(2): 300–320, 2009.

B. M. Ningegowda and B. Premachandran. A coupled level set and volume of fluid method with multi-directional advection algorithms for two-phase flows with and without phase change. *Int. J. Heat & Mass Transfer*, 79:532–550, 2014.

W. F. Noh and P. Woodward. Slic (simple line interface calculation). In *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics June 28–July 2, 1976 Twente University, Enschede*, pages 330–340. Springer, 1976.

R. R. Nourgaliev and T. G. Theofanous. High-fidelity interface tracking in compressible flows: Unlimited anchored adaptive level set. *J. Comput. Phys.*, 224(2):836–866, 2007.

M. Ohta, T. Imura, Y. Yoshida, and M. Sussman. A computational study of the effect of initial bubble conditions on the motion of a gas bubble rising in viscous liquids. *Int. J. Multiphase flow*, 31(2):223–237, 2005.

Thomas C Oppe, Wayne D Joubert, and David R Kincaid. Nspcg user's guide, version 1.0: A package for solving large sparse linear systems by various iterative methods. Technical report, Tech. Rep. CNA–216, Center for Numerical Analysis, University of Texas at Austin, Austin, TX, 1988.

S. Osher and R. Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153. Springer, 2002.

S. Osher and R. Fedkiw. *Level set methods and dynamic implicit surfaces*. Springer-Verlag New York, Inc., 2003.

S. Osher and J. A Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.

S. Osher and C.-W. Shu. High order essentially non-oscillatory schemes for Hamilton-Jacobi equations. *SIAM J. Numer. Anal.*, 28:902–921, 1991.

D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A PDE-based fast local level set method. *J. Comput. Phys.*, 155(2):410–438, 1999.

C. S. Peskin. Numerical analysis of blood flow in the heart. *J. Comput. Phys.*, 25(3):220–252, 1977.

C. D. Pierce. *Progress-variable approach for large-eddy simulation of turbulent combustion*. PhD thesis, Stanford University, Stanford, CA, 2001.

J. E. Pilliod. *An analysis of piecewise linear interface reconstruction algorithms for volume-of-fluid methods*. Master's thesis, University of California, 1992.

J. E. Pilliod and E. G. Puckett. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *J. Comput. Phys.*, 199(2):465–502, 2004.

S. Popinet. The gerris flow solver, available at http://gfs.sourceforge.net. 2015.

S. Popinet. Gerris: a tree-based adaptive solver for the incompressible euler equations in complex geometries. *J. Comput. Phys.*, 190(2):572–600, 2003.

S. Popinet. An accurate adaptive solver for surface-tension-driven interfacial flows. *J. Comput. Phys.*, 228(16):5838–5866, 2009.

S. Popinet and S. Zaleski. A front-tracking algorithm for accurate representation of surface tension. *Int. J. Numer. Meth. Fluids*, 30(6):775–793, 1999.

S. Popinet et al. The basilisk flow solver is the successor of gerris. it is available at http://basilisk.fr. 2016.

A. Prosperetti. Motion of two superposed viscous fluids. *Phys. Fluids*, 24(7):1217–1223, 1981.

E. Puckett. On the second-order accuracy of volume-of-fluid interface reconstruction algorithms: convergence in the max norm. *Communications in Applied Mathematics and Computational Science*, 5(1):99–148, 2010.

E. G. Puckett. A volume-of-fluid interface tracking algorithm with applications to computing shock wave refraction. In *Proceedings of the Fourth International Symposium on Computational Fluid Dynamics*, pages 933–938, 1991.

E. G. Puckett, A. S. Almgren, J. B. Bell, D. L. Marcus, and W. J. Rider. A high-order projection method for tracking fluid interfaces in variable density incompressible flows. *J. Comput. Phys.*, 130(2):269–282, 1997.

F. Raees. *A Mass-Conserving hybrid interface capturing method for geometrically complicated domains*. PhD thesis, 2016.

Y. Renardy and M. Renardy. Prost: A parabolic reconstruction of surface tension for the volume-of-fluid method. *J. Comput. Phys.*, 183(2):400–421, 2002.

W. J. Rider. Approximate projection methods for incompressible flows: Implementation, variants and robustness. Technical report, Los Alamos National Laboratory (LA-UR-2000), 1995.

W. J. Rider and D. B. Kothe. Reconstructing volume tracking,. *J. Comput. Phys.*, 141(2): 112–152, April 1998.

G. Russo and P. Smereka. A remark on computing distance functions. *J. Comput. Phys.*, 163 (1):51–67, 2000.

R. Scardovelli and S. Zaleski. Analytical relations connecting linear interfaces and volume fractions in rectangular grids. *J. Comput. Phys.*, 164(1):228–237, 2000.

R. Scardovelli and S. Zaleski. Interface reconstruction with least-square fit and split eulerian–lagrangian advection. *Int. J. Numer. Meth. Fluids*, 41(3):251–274, 2003.

J. A. Sethian. Fast marching methods. *SIAM review*, 41(2):199–235, 1999.

D. Shi, Q. Bi, and R. Zhou. Numerical simulation of a falling ferrofluid droplet in a uniform magnetic field by the voset method. *Numerical Heat Transfer, Part A: Applications*, 66(2): 144–164, 2014.

S. Shin, S. I. Abdel-Khalik, V. Daru, and D. Juric. Accurate representation of surface tension using the level contour reconstruction method. *J. Comput. Phys.*, 203(2):493–516, March 2005.

C.-W. Shu. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. Technical report, NASA/CR-97-206253 ICASE Report No. 97-65, 1997.

C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes, ii. *J. Comput. Phys.*, 83(1):32–78, 1989.

G. Son. Efficient implementation of a coupled level-set and volume-of-fluid method for three-dimensional incompressible two-phase flows. *Numerical Heat Transfer: Part B: Fundamentals*, 43(6):549–565, 2003.

G. Son and V. K. Dhir. A level set method for analysis of film boiling on an immersed solid surface. *Numerical Heat Transfer, Part B: Fundamentals*, 52(2):153–177, 2007.

G. Son and N. Hur. A coupled level set and volume-of-fluid method for the buoyancy-driven motion of fluid particles. *Numerical Heat Transfer: Part B: Fundamentals*, 42(6):523–542, 2002.

D. L. Sun and W. Q. Tao. A coupled volume-of-fluid and level set (voset) method for computing incompressible two-phase flows. *Int. J. Heat & Mass Transfer*, 53(4):645–655, January 2010.

M. B. Sun, Z. G. Wang, and X. S. Bai. Assessment and modification of sub-cell-fix method for re-initialization of level-set distance function. *Int. J. Numer. Meth. Fluids*, 62:211–236, 2010.

M. Sussman. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *J. Comput. Phys.*, 187(1):110–136, 2003.

M. Sussman and E. Fatemi. An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM J. Sci. Comput.*, 20:1165–1191, 1999.

M. Sussman and E. G. Puckett. A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. *J. Comput. Phys.*, 162(2):301–337, 2000.

M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.*, 114(1):146–159, 1994.

M. Sussman, E. Fatemi, P. Smereka, and S. Osher. An improved level set method for incompressible two-phase flows. *Computers & Fluids*, 27(5-6):663–680, 1998.

M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome. An adaptive level set approach for incompressible two-phase flows. *J. Comput. Phys.*, 148(1):81–124, 1999.

M. Sussman, K. M Smith, M Y. Hussaini, M. Ohta, and R Zhi-Wei. A sharp interface method for incompressible two-phase flows. *J. Comput. Phys.*, 221(2):469–505, 2007.

S. Tanguy. Développement d'une méthode de suivi d'interface. application aux écoulements diphasiques. Master's thesis, Faculté des sciences de l'université de Rouen, 2004.

G. Tomar, G. Biswas, A. Sharma, and A. Agrawal. Numerical simulation of bubble growth in film boiling using a coupled level-set and volume-of-fluid method. *Phys. Fluids*, 17(11):112103–13, 2005.

G. Tomar, G. Biswas, A. Sharma, and S.W.J. Welch. Multimode analysis of bubble growth in saturated film boiling. *Phys. Fluids*, 20(9):092101, 2008.

A. Y. Tong and Z. Wang. A numerical method for capillarity-dominant free surface flows. *J. Comput. Phys.*, 221(2):506–523, 2007.

D. J. Torres and J. U. Brackbill. The point-set method: front-tracking without connectivity. *J. Comput. Phys.*, 165(2):620–644, 2000.

G. Tryggvason, R. Scardovelli, and S. Zaleski. *Direct numerical simulations of gas-liquid multiphase flows*. Cambridge University Press Cambridge, 2011.

O. Ubbink. *Numerical prediction of two fluid systems with sharp interfaces*. PhD thesis, University of London PhD Thesis, 1997.

O. Ubbink and R. I. Issa. A method for capturing sharp fluid interfaces on arbitrary meshes. *J. Comput. Phys.*, 153(1):26–50, 1999.

S. O. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *J. Comput. Phys.*, 100(1):25–37, 1992.

S. P. van der Pijl, A. Segal, C. Vuik, and P. Wesseling. A mass-conserving level-set method for modelling of multi-phase flows. *Int. J. Numer. Meth. Fluids*, 47(4):339–361, 2005.

S. P. van der Pijl, A. Segal, C. Vuik, and P. Wesseling. Computing three-dimensional two-phase flows with a mass-conserving level set method. *Comput. Vis. Sci.*, 11(4):221–235, September 2008.

M. Van Sint Annaland, N.G. Deen, and J.A.M. Kuipers. Numerical simulation of gas bubbles behaviour using a three-dimensional volume of fluid method. *Chemical Engineering Science*, 60(11):2999–3011, 2005.

P. Vigneaux. *Méthode level set pour des problèmes d'interfaces en microfluidique*. PhD thesis, Université de Bordeaux I, 2008.

T. G. Vignesh and S. Bakshi. Noniterative interface reconstruction algorithms for volume of fluid method. *Int. J. Numer. Meth. Fluids*, 73(1):1–18, 2013.

K. Voronetska. *Simulation numérique directe des écoulements à phases dispersées*. PhD thesis, UNIVERSITÉ BORDEAUX, 2012.

C. Wall, C. D. Pierce, and P. Moin. A semi-implicit method for resolution of acoustic waves in low mach number flows. *J. Comput. Phys.*, 181(2):545–563, 2002.

T. Wang, H. Li, Y. Feng, and D. Shi. A coupled volume-of-fluid and level set (voset) method on dynamically adaptive quadtree grids. *Int. J. Heat & Mass Transfer*, 67:70–73, 2013.

T. Wang, H. Li, Y. Zhang, W. Han, T. Sheng, and W. Zhang. Numerical simulation of two-phase flows using 3d-voset method on dynamically adaptive octree grids. In *2014 22nd International Conference on Nuclear Engineering*, pages V004T10A012–V004T10A012. American Society of Mechanical Engineers, 2014.

T. Wang, H. Li, Y. Zhang, and D. Shi. Numerical simulation of bubble dynamics in a uniform electric field by the adaptive 3d-voset method. *Numerical Heat Transfer, Part A: Applications*, 67(12):1352–1369, 2015.

T. Wang, H. Li, and J. Zhao. Three-dimensional numerical simulation of bubble dynamics in microgravity under the influence of nonuniform electric fields. *Microgravity Science and Technology*, pages 1–10, 2016.

Y. Wang, S. Simakhina, and M. Sussman. A hybrid level set-volume constraint method for incompressible two-phase flow. *J. Comput. Phys.*, 231(19):6438–6471, 2012a.

Z. Wang and A. Y Tong. Deformation and oscillations of a single gas bubble rising in a narrow vertical tube. In *ASME 4th International Conference on Nanochannels, Microchannels, and Minichannels*, pages 1195–1202. American Society of Mechanical Engineers, 2006.

Z. Wang and A. Y. Tong. A sharp surface tension modeling method for two-phase incompressible interfacial flows. *Int. J. Numer. Meth. Fluids*, 64(7):709–732, 2010.

Z. Wang, J. Yang, B. Koo, and F. Stern. A coupled level set and volume-of-fluid method for sharp interface simulation of plunging breaking waves. *Int. J. Multiphase Flow*, 35(3):227–246, 2009.

Z. Wang, J. Yang, and F. Stern. A new volume-of-fluid method with a constructed distance function on general structured grids. *J. Comput. Phys.*, 231(9):3703–3722, 2012b.

H. Weller. A new approach to vof-based interface capturing methods for incompressible and compressible flow, opencfd ltd. Technical report, Report TR/HGW/04, 2008.

H. G. Weller. The development of a new flame area combustion model using conditional averaging. *Thermo-fluids section report TF*, 9307, 1993.

G. D. Weymouth and D. K.-P. Yue. Conservative volume-of-fluid method for free-surface simulations on cartesian-grids. *J. Comput. Phys.*, 229(8):2853–2865, 2010.

M. Wörner. Numerical modeling of multiphase flows in microfluidics and micro process engineering: a review of methods and applications. *Microfluidics and nanofluidics*, 12(6):841–886, 2012.

Y. F. Yap, J. C. Chai, T. N. Wong, K. C. Toh, and H. Y. Zhang. A global mass correction scheme for the level-set method. *Numerical Heat Transfer, Part B: Fundamentals*, 50(5): 455–472, 2006.

K. Yokoi, D. Vadillo, J. Hinch, and I. Hutchings. Numerical studies of the influence of the dynamic contact angle on a droplet impacting on a dry surface. *Phys. Fluids*, 21(7):072102, 2009.

D. L. Youngs. Time-dependent multi-material flow with large fluid distortion. *Numerical methods for fluid dynamics*, 24(2):273–285, 1982.

S. T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.*, 31(3):335–362, 1979.

J. Zhang. Numerical methods for partial differential equations. *Numer. Methods Partial Different. Eqs.*, 14(2):263–280, 1998.

Y. Zhang, Q. Zou, and D. Greaves. Numerical simulation of free-surface flow using the level-set method with global mass correction. *Int. J. Numer. Meth. Fluids*, 63(6):651–680, 2010.

D. Zuzio. *Direct numerical simulation of two phase flows with adaptive mesh refinement.* PhD thesis, Université de Toulouse, 2010.

FOLIO ADMINISTRATIF

THESE DE L'UNIVERSITE DE LYON OPEREE AU SEIN DE L'INSA LYON

NOM : DJATI
Prénom : Nabil

DATE de SOUTENANCE : 22 Juin 2017

TITRE : Study of interface capturing methods for two-phase flows.

NATURE : Doctorat

Numéro d'ordre : 2017LYSEI052

Ecole doctorale : MEGA de Lyon (Mécanique, Energétique, Génie Civil, Acoustique)

Spécialité :Thermique Energétique

RESUME :
This thesis is devoted to the development and comparison of interface methods for incompressible two-phase flows. It focuses on the selection of robust interface capturing methods, then on the manner of their coupling with the Navier-stokes solver. The level-set method is first investigated, in particular the influence of the advection scheme and the reinitialization step on the accuracy of the interface capturing. It is shown that the volume constraint method for reinitialization is robust and accurate in combination with the conservative fifth-order WENO schemes for the advection. It is found that interface errors increase drastically when the CFL number is very small. As a remedy, reinitializing the level-set field less often reduces the amount of numerical diffusion and non-physical interface displacement. Mass conservation is, however, not guaranteed with the level-set methods. The volume-of-fluid (VOF) method is then investigated, which naturally conserves the mass of the reference fluid. A geometrical consistent and conservative scheme is adopted, then an alternative technique more easily extended to 3D. It is found that both methods give very similar results. The moment-of-fluid (MOF) method, which reconstructs the interface using the reference fluid centroid, is found to be more accurate than the VOF methods. Different coupled level-set and VOF methods are then investigated, namely: CLSVOF, MCLS, VOSET and CLSMOF. It is observed that the level-set method tends to thicken thin filaments, whereas the VOF and coupled methods break up thin structures in small fluid particles. Finally, we coupled the level-set and volume-of-fluid methods with the incompressible Navier-Stokes solver. We compared different manners (sharp and smoothed) of treating the interface jump conditions. It is shown that the VOF methods are more robust, and provide excellent results for almost all the performed simulations. Two level-set methods are also identified that give very good results, comparable to those obtained with the VOF methods.

MOTS-CLÉS : Two-phase incompressible flow simulation, interface capturing, level-set method, volume-of-fluid method, moment-of-fluid method, coupled level-set and volume-of-fluid, ghost fluid method, continuous surface force.

Laboratoire (s) de recherche : Centre d'Energétique et Thermique de Lyon - CETHIL

Directeur de thèse : Shihe XIN

Président de jury : Peter SPELT

Composition du jury :

SPELT, Peter	Professeur (EC Lyon)	Président
CHENIER, Eric	Maître de Conférences (Université Paris-Est Marne la vallée)	Rapporteur
JOSSERAND, Christophe	Directeur de Recherche CNRS (Ecole Polytechnique, Palaiseau)	Rapporteur
DARU, Virginie	Maître de Conférences (ENSAM-Paris)	Examinatrice
XIN, Shihe	Professeur (INSA-Lyon)	Directeur de thèse
KNIKKER, Ronnie	Maître de Conférences (INSA-Lyon)	Codirecteur de thèse