Free-surface flow simulations with a Lagrangian and an Arbitrary Lagrangian–Eulerian methods

Martin Ferrand

École des Ponts ParisTech

THÈSE DE DOCTORAT
de l'École des Ponts ParisTech

Simulations d'écoulements à surface libre avec des méthodes lagrangienne et arbitrairement lagrangienne–eulérienne

Free-surface flow simulations with a Lagrangian and an arbitrary Lagrangian–Eulerian methods

École doctorale N° 531, Sciences Ingénierie et Environnement (SIE)

Spécialité : Mécanique des fluides

Thèse préparée au sein du laboratoire N°2840, CEREA,
et du département Mécanique des Fluides Energie et Environnement d'EDF R&D

Thèse soutenue le 19/01/2022, par

Martin FERRAND

Composition du jury :

Mario, RICCHIUTO Directeur de recherche, INRIA	*Président*
Pierre, SAGAUT Professeur, Université d'Aix Marseille	*Rapporteur*
David, LE TOUZÉ Professeur, École Centrale de Nantes	*Rapporteur*
Raphaèle, HERBIN Professeur, Université d'Aix Marseille	*Examinatrice*
Mario, RICCHIUTO Directeur de recherche, INRIA	*Examinateur*
Damien, VIOLEAU Professeur, Ecole des Ponts ParisTech	*Directeur de thèse*
Benedict, ROGERS Professeur, University of Manchester	*Invité*

Contents

List of Figures

List of Tables

Margot, j'ai hâte de pouvoir discuter de cela avec toi, et que tu m'apprennes des choses à ton tour.

Remerciements

Il n'est pas simple d'écrire ces lignes de remerciement et de n'oublier personne. Je tiens tout d'abord à remercier les membres du jury, d'avoir accepté d'évaluer mes travaux. Merci à Damien Violeau, c'est lui qui m'a donné l'envie, avec les autres professeurs de mécanique des fluides de l'École des Ponts ParisTech (Sofiane Benhamadouche, Michel Benoît et Bertrand Carissimo lorsque j'étais étudiant) de rejoindre la R&D du groupe EDF. Damien a aussi été un encadrant bienveillant, motivant, et exigeant. Je lui dois beaucoup. Mes remerciements vont aussi à l'endroit de Benedict Rogers et de Dominique Laurence, qui m'ont encadré et formé lorsque je finissais mes études d'ingénieur à Manchester.

Je remercie ensuite Frédéric Archambeau de m'avoir recruté dans l'équipe de développement de code_saturne ainsi que le chef de département de Mécanique des Fluides, Énergie et Environnement de l'époque, Pascal Mialon. Les mots de ma fiche de poste ("avec pour objectif de devenir un référant de code_saturne" ndrl) sont restés gravés, autant que Frédéric, dans mes pensées.

Ces années de développement logiciel m'ont permis d'apprendre aussi bien en modélisation, en mathématiques appliquées et en algorithmique, au côté de mes collègues, devenus amis, de l'équipe code_saturne (par ordre d'arrivée) : merci à Yvan Fournier pour son soutien indéfectible et son aide au débogage irremplaçable, merci à David Monfort de m'avoir initié au développement dans code_saturne, merci à Jérôme Bonelle pour les échanges au sujet des schémas, merci à Bruno Audebert de m'avoir montré l'exigence du rôle de chef de projet, merci à Jacques Fontaine pour les lignes de codes écrites ensembles, et enfin merci à Erwan Le Coupanec pour sa rigueur et son implication quotidienne (tu as aussi eu la dure tâche de partager mon bureau toutes ces années). Il a été pour moi un honneur de diriger cette équipe, qui n'avait pas tellement besoin d'être motivée. Je n'oublie pas les nouveaux venus, Jean-François Wald, Chaï Koren, Thomas Fonty et Vincent Stobiac, c'est un plaisir d'échanger avec vous au quotidien.

Un remerciement aussi pour ceux qui développent le solveur diphasique neptune_cfd, avec qui j'ai eu plaisir à échanger (par ordre alphabétique) : Mathieu Guingo, Jérôme Laviéville, Namane Méchitoua et Nicolas Mérigoux.

Ce manuscrit comprend des travaux sur la méthode SPH que j'ai eu le plaisir de partager avec des amis avec qui il était agréable de développer lors des soirée *Geek-Party* (par ordre d'arrivée) : Christophe Kassiotis, François-Xavier Morel, Antoine Joly, Arno Mayhoffer et Agnes Leroy. Je suis déjà nostalgique de ces soirées.

Je tiens aussi à remercier les personnes avec qui j'ai eu le plaisir d'encadrer des élèves stagiaires, chronologiquement : Bertrand Sapa, Fatiha Nmira, Christophe Péniguel, Charles Demay, Jérôme Bonelle, Erwan Le Coupanec, Jean-Marc Hérard, Jean-François Wald, Jean-Pierre Minier, Sofiane Benhamadouche et Bertrand Carissimo ; ainsi que ces élèves pour qui il est toujours formateur d'essayer de transmettre. Je remercie aussi les personnes avec qui j'ai écrits des articles (par ordre alphabétique) : Ophélie Angelini, Sofiane Benhamadouche, Clément Colas, Cédric Flageul, Jacques Fontaine, Thomas Fonty, Alex Ghaïtanellis, Jeffrey Harris, Jean-Marc Hérard, Antoine Joly, Christophe Kassiotis, Jean-Claude Latché, Dominique Laurence, Erwan Le Coupanec, Agnès Leroy, Arno Mayrhoffer, François-Xavier Morel, Benedict Rogers. Plus largement, je remercie mes nombreux collègues, j'apprécie travailler avec vous.

EDF R&D a la chance d'avoir en son sein des experts qui m'ont énormément appris (sans ordre) : merci Jean-Marc Hérard de m'avoir tant appris et pour ta bienveillante motivation, sans toi le département MFEE aurait infiniment moins d'exigence scientifique ; merci Sofiane Benhamadouche pour ta rigueur et les nombreuses discussions concernant la turbulence, les pertes de charges notamment et les schémas de discrétisation, merci Jean-Pierre Minier pour les cours et formations sur les méthodes hybrides moments/PDF, ta rigueur et tes intuitions, tes explications et tes idées font de nos échanges un plaisir ; merci à nouveau à Jérôme Bonelle pour les heures

passées à échanger sur les schémas de discrétisation, tes travaux sont très inspirants ; Merci à Bertrand Carissimo, tu as la patience d'expliquer les fondamentaux de la (micro)-météorologie et leurs impacts sur la dispersion, j'apprécie nos échanges. Enfin, Yvan, je ne compte plus les heures que tu m'as fait gagner en résolvant des bogues et en m'initiant à l'architecture logicielle, j'espère que ce manuscrit saura te donner des idées pour que tu puisses avoir la reconnaissance que tu mérites.

Je remercie Isabelle François, de m'avoir encouragé à l'écriture de cette thèse, et d'avoir su me libérer un peu de temps via l'initiative locale MFEE. Je remercie aussi Jean-Paul Chabard et Françoise Préteux, pour la confiance qu'ils m'ont accordée. Je remercie mes collègues du CEREA avec qui je suis heureux d'apprendre quotidiennement.

Je remercie pour leur nombreuses corrections (par ordre du nombre de corrections) : Thomas Fonty, Cédric Flageul, Romain Camy et Damien Violeau.

Je remercie toutes les personnes qui m'ont porté assistance (par ordre alphabétique) : Dalida Chartrel, Youngseob Kim, Lydie Periac et Anne Peure.

Je remercie mes amis (sans ordre), de science, d'enfance ou de football.

Enfin, je finis ces remerciements par ma famille (par ordre d'arrivée) : parents, sœur et frère, Lorène, merci de votre soutien. J'espère rendre mes grands-parents fiers de moi là où ils sont.

Simulations d'écoulements à surface libre avec des méthodes lagrangienne et arbitrairement lagrangienne–eulérienne

Abstract Free-surface flows have various natures in the environmental and industrial contexts. It may be a smoothly-varying regular surface with waves from which one may want to extract renewable energy, or an also smooth surface of an evaporating pool in case of recirculating pumps deficiency, but also a flow down a spillway with a really disturbed free-surface.

Lagrange and/or Euler approaches can be used to solve the discretised Navier–Stokes equations with a free-surface.

Among the Lagrangian methods, smoothed particle hydrodynamics (SPH) is a mesh-less numerical method ideal for simulating potentially violent free-surface phenomena such as a wave breaking, or a dam-break for which many Eulerian methods can be difficult to apply.

Gentle free-surface flows can also be tackled with the arbitrary Lagrangian Eulerian (ALE) mesh-based finite volume method, where the free-surface faces of the mesh move so that the kinematic boundary condition is fulfilled.

The first Chapter is made to introduce SPH and ALE finite volume and to draw links between the two methods.

For SPH, dealing with boundary conditions (walls and open boundaries) is one of the most challenging parts as it is declared as one of the *Grand Challenges* of the international organisation representing the community of researchers and industrial users of smoothed particle hydrodynamics (SPHERIC). Concerning walls, the proposed methodology introduced in the second Chapter relies on the semi-analytical approach which consists in renormalising the density field near a solid wall with respect to the missing kernel support area.The SPH part of this work extends this semi-analytical methodology, where intrinsic *gradient* and *divergence* operators that ensure conservation properties are employed. The accuracy of the physical field such as the pressure next to walls is considerably improved, and the consistent manner developed to wall-correct operators allows us to perform simulations with turbulence models.

The new model is demonstrated for cases including hydrostatic conditions for a channel flow, still water in a tank of complex geometry and a dam-break over triangular bed profile with sharp angle where a significantly-improved behaviour is obtained in comparison with the conventional boundary techniques. Simulation of the benchmark test case of a square object moving in an enclosed tank shows good agreement with the reference solution and no voids are formed within the fluid domain. The performance of the model for a 2-D turbulent flow in a channel is demonstrated as the profiles of velocity, turbulent kinetic energy and its dissipation are in agreement with the theoretical ones. Finally, the performance of the model is demonstrated for flow in a fish-pass where velocity field and turbulent viscosity field are satisfactorily reproduced compared to mesh-based codes. This work has been published in Ferrand et al. (2013b).

An axisymmetric formulation with a unified renormalisation factor taking both radial correction and wall renormalisation is proposed as an extension of this work in Appendix D.5.

The third Chapter deals with open-boundaries for the SPH approach with the resolution of a Riemann problem associated to the hyperbolic compressible SPH framework used. The discretisation of the boundary in surface elements (segments in 2-D) and vertices is adequate to make particles enter progressively so that no pressure wave are created by the release of new fluid particles. This work has been published in Ferrand et al. (2017). Some details or how to integrate the geometrical renormalisation factor used in the SPH boundary conditions is presented in Appendix. This work has been published in Mayrhofer et al. (2015).

The fourth Chapter presents the ALE finite volume algorithm developed in the massively parallel open-source code code_saturne (see Team (2021)). An original mixing of cell-based numerical scheme used to get conservation of mass and momentum on each cell control volume and a vertex-based scheme based on the compatible discrete operators (CDO) approach

is presented with a particular care on the free-surface condition both on fluid and mesh-displacement. Various verification and validation test cases are presented. This work has been published in Ferrand and Harris (2021). Space discretisation of a Poisson equation, used for the mass correction step in the ALE finite volume approach, is presented in Appendix A and has been published in Ferrand et al. (2014).

Résumé Les écoulements à surface libre peuvent présenter des aspects très différentes dans l'environnement et l'industrie. Il peut s'agir d'une surface tranquille, régulière mathématiquement parlant, avec des vagues dont on peut vouloir extraire de l'énergie renouvelable, ou encore d'une surface, lisse également, d'une piscine s'évaporant suite à une défaillance des pompes de recirculation, ou enfin d'un écoulement dans un déversoir de barrage avec une surface libre vraiment perturbée et complexe dans sa forme et sa topologie. Les approches lagrangiennes et / ou eulériennes peuvent être utilisées pour résoudre les équations discrétisées de Navier–Stokes avec une surface libre.

Parmi les méthodes lagrangiennes, l'hydrodynamique des particules lissées (en anglais "*smoothed particule hydrodynamics*" SPH) est une méthode numérique sans maillage idéale pour simuler des phénomènes potentiellement violents de surface libre très déformée tels qu'une vague déferlante ou un déversoir de barrage pour lequel de nombreuses méthodes eulériennes peuvent être difficiles à appliquer.

Les écoulements à surface libre réguliers peuvent également être traités avec la méthode des volumes finis sur maillage déformable avec une approche arbitrairement lagrangienne eulérienne (ALE), où les faces de surface libre du maillage se déplacent de sorte que la condition aux limites cinématique soit vérifiée.

Le premier chapitre a pour objectif de présenter les méthodes volumes finis ALE et SPH et den établir des liens.

Pour SPH, la gestion des conditions aux limites (murs et frontières ouvertes) est l'une des parties les plus difficiles car elle est déclarée comme l'un des grands défis de l'organisation internationale représentant la communauté des chercheurs et des utilisateurs industriels de l'hydrodynamique des particules lissées (SPHERIC). Concernant les murs, nos travaux suivent lapproche semi-analytique, introduite au chapitre 2, qui consiste à renormaliser la masse volumique près d'un mur plein par rapport à la zone de support de noyau manquante. La partie SPH de ce travail étend cette méthodologie semi-analytique, où des opérateurs de gradient et de divergence intrinsèques qui garantissent des propriétés de conservation sont employés. La précision du champ physique telle que la pression à côté des murs est considérablement améliorée, et la manière cohérente de prise en compte des parois développée pour les opérateurs nous permet d'effectuer des simulations avec des modèles de turbulence.

Une formulation axisymétrique avec un facteur de renormalisation unifié prenant à la fois la correction radiale et la renormalisation de la paroi est proposée en annexe.

Le chapitre 3 traite des frontières ouvertes pour l'approche SPH avec la résolution d'un problème de Riemann associé au cadre SPH compressible hyperbolique utilisé. La discrétisation de la frontière en éléments de surface (segments en 2-D) et en sommets est adéquate pour faire entrer les particules progressivement afin qu'aucune onde de pression ne soit créée par la libération de nouvelles particules fluides. Quelques autres aspects sur SPH dont lintégration du facteur de renormalisation sont présentés en annexe.

Le chapitre 4 présente l'algorithme volumes finis ALE développé dans le code open-source massivement parallèle code_saturne. Un mélange original de schéma numérique basé sur les cellules utilisé pour obtenir la conservation de la masse et de la quantité de mouvement sur chaque volume de contrôle de cellule et un schéma basé sur les sommets basé sur

l'approche des opérateurs discrets compatibles (CDO) est présenté avec un soin particulier sur la condition à la limite de la surface libre aussi bien du point de vue du fluide que du déplacement du maillage. Différents cas de test de vérification et de validation sont présentés. La discrétisation spatiale d'une équation de Poisson, utilisée pour l'étape de correction de masse dans l'approche des volumes finis ALE, est présentée en annexe.

Contexte et organisation du manuscrit

Ce manuscrit rassemble une partie de mes travaux réalisés dans la modélisation et la simulation numérique des écoulements depuis mon master de recherche (Mphil) réalisé à l'université de Manchester en 2010 (Ferrand (2011)). Durant ma carrière d'ingénieur chercheur à EDF – R&D (2010 – maintenant) et de pilote opérationnel du projet Plateforme ThermoHydraulique Locale (PTHL, 2014 – 2019)[1], puis maintenant au laboratoire Centre d'Enseignement et de Recherche en Environnement Atmosphérique (CEREA), j'ai eu l'opportunité de mettre en œuvre des modèles et des algorithmes pour apporter des réponses à des enjeux variés pour EDF (conditions d'entrée libre pour les simulations d'incendies, algorithmes dilatables, modèles avec phase glissante pour la simulation de la combustion charbon et biomasse, mise en place d'un module de simulation de la ségrégation carbone dans les gros forgés, mise en place de modélisation permettant la simulation de colmatage de tambour filtrant en cas d'arrivée massive de colmatants, mise en place de modèles pour l'étude des piscines de stockage du bâtiment combustible en situation incidentelle, étude du comportement des colis bitumineux de stockage des déchets radioactif sous sollicitation thermique due à un incendie,...). J'ai eu l'occasion de participer au développement d'outils de calcul scientifique, principalement code_saturne, code volumes finis, développé depuis la fin du 20^e siècle, open-source depuis 2006 ; mais aussi dans les codes particulaires SPARTACUS-2D et Sphynx. Il m'a été permis d'encadrer une dizaine de stages, et de participer au suivi de 5 thèses. Enfin, j'ai la chance d'enseigner la mécanique des fluides et la mécanique des fluides numérique à l'École des Ingénieurs de la Ville de Paris (vacataire de 2010 à 2020, responsable du module de mécanique des fluides de 2019 à 2020) et à l'École des Ponts ParisTech (vacataire remplaçant de 2011 à 2015, puis vacataire depuis 2016 sur le cours de mécanique des fluide, enseignant vacataire et responsable du cours de mécanique des fluides numérique depuis 2015).

La sélection des travaux présentée dans ce manuscrit est faite en suivant le fil rouge du point de vue arbitrairement lagrangien-eulérien (ALE) pour les écoulements à surface libre.

Les écoulements à surface libre représentent un large spectre d'applications pour une entreprise comme EDF : nous pouvons penser en premier lieu aux barrages hydro-électriques, aux vagues comme menaces de constructions côtières ou d'éoliennes *off-shore*. Les surfaces libres sont aussi à considérer lorsque l'on s'intéresse au comportement de piscine de stockage de combustible nucléaire usé, notamment en prenant en compte l'expérience post-Fukushima ; mais encore lorsque l'on s'intéresse au comportement d'un colis bitumeux sous sollicitation thermique intense lors d'un incendie hypothétique d'un stockage profond : le bitume se comporte d'abord comme un milieu continu visco-plastique puis comme un fluide newtonien, sa surface montant dans le colis du fait de la dilatation thermique.

La première méthode développée, *smoothed particle hydrodynamics* (SPH) repose sur une discrétisation en particules, et est principalement lagrangienne (mis à part les particules discrétisant le bord du domaine étudié).

La seconde méthode repose sur l'approche des volumes finis en version ALE pour la

[1] Projet qui développe le logiciel open-source code_saturne d'EDF, ses méthodes numériques, ses modélisations physiques, ses algorithmes pour le calcul haute performance, sa vérification et validation, ainsi que son intégration dans la plateforme Salome.

simulation des écoulements à surface libre.

Les travaux réalisés ont fait l'objet d'une attention particulière sur le traitement des conditions aux limites, pour ce qui est du traitement des parois (Chapitre 2) ou des frontières ouvertes (Chapitre 3) pour la méthode SPH (il est à noter que le traitement des conditions aux limites constitue un des grands challenges de la communauté SPH), ou pour le traitement de la surface libre pour la méthode des volumes finis ALE (Chapitre 4). Dans ce dernier point, l'usage de schémas numériques sur le maillage dual, associé aux nœuds du maillage a permis d'améliorer la robustesse globale du schéma.

Les deux méthodes développées et présentées dans ce manuscrit ont fait l'objet d'articles publiés dans Ferrand et al. (2013b), Ferrand et al. (2017) et Ferrand and Harris (2021).

1. Introduction

1.1 System of equations, Lagrangian and Eulerian approaches and free-surface

1.1.1 Fluid motion equations

Fluid motion, when studied with the continuum hypothesis in an open space (usually bounded) Ω, is described by conservative laws – mass, momentum and energy conservation – supplemented with equations of state (EOS) or constitutive laws.

Conservation of mass and momentum leads to Cauchy equations:

$$\begin{aligned} &\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \boldsymbol{u}) = \dot{m}, \\ &\frac{\partial}{\partial t}(\rho \boldsymbol{u}) + \mathbf{div}(\boldsymbol{u} \otimes \rho\, \boldsymbol{u}) = \mathbf{div}(\boldsymbol{\sigma}) + \boldsymbol{f} + \dot{m}\boldsymbol{u}^{in}, \end{aligned} \tag{1.1}$$

where $\boldsymbol{u}$ represents the fluid velocity, ρ is the density of the fluid, $\boldsymbol{\sigma}$ is the stress field, $\boldsymbol{f}$ is a field of volume force (e.g., gravity force, Coriolis force, additional head losses force), $\dot{m}$ is a possible mass source, and $\boldsymbol{u}^{in}$ the value of the corresponding injected velocity. For the sake of simplicity, $\dot{m}$ will be set to zero hereafter.

The stress tensor $\boldsymbol{\sigma}$ is decomposed into an isotropic component, called the pressure p, and the deviatoric component, $\boldsymbol{\tau}$:

$$\boldsymbol{\sigma} = -p\mathbf{1} + \boldsymbol{\tau}, \tag{1.2}$$

and thus the Cauchy equations are:

$$\begin{aligned} &\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \boldsymbol{u}) = 0, \\ &\frac{\partial}{\partial t}(\rho \boldsymbol{u}) + \mathbf{div}(\boldsymbol{u} \otimes \rho\, \boldsymbol{u}) = \boldsymbol{f} - \boldsymbol{\nabla} p + \mathbf{div}\,(\boldsymbol{\tau}). \end{aligned} \tag{1.3}$$

For a Newtonian fluid, the constitutive law connecting the stress tensor to the deviatoric part of the deformation rate tensor, given as $\boldsymbol{S} := \frac{1}{2}\left(\boldsymbol{\nabla}\boldsymbol{u} + \boldsymbol{\nabla}\boldsymbol{u}^T\right)$, is determined as:

$$\boldsymbol{\tau}(\boldsymbol{u}) = \mu\left(\boldsymbol{\nabla}\boldsymbol{u} + \boldsymbol{\nabla}\boldsymbol{u}^T\right) + \left(\kappa - \frac{2}{3}\mu\right)\operatorname{tr}(\boldsymbol{\nabla}\boldsymbol{u})\,\mathbf{1}, \tag{1.4}$$

where κ is the volume viscosity, usually neglected and therefore omitted hereafter. Note that at the continuous level, $\mathrm{tr}\left(\boldsymbol{\nabla}\,\boldsymbol{u}\right) = \mathrm{div}\left(\boldsymbol{u}\right)$.

The combination of terms in the Cauchy momentum Equations (1.1) and a Newtonian fluid, Eq. (1.4) gives the Navier–Stokes equations:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \mathrm{div}\left(\rho \boldsymbol{u}\right) &= 0, \\ \frac{\partial}{\partial t}(\rho \boldsymbol{u}) + \mathbf{div}\left(\boldsymbol{u} \otimes \rho\, \boldsymbol{u}\right) &= \boldsymbol{f} - \boldsymbol{\nabla} p + \mathbf{div}\left[\mu\left(\boldsymbol{\nabla}\,\boldsymbol{u} + \boldsymbol{\nabla}\,\boldsymbol{u}^T\right) - \tfrac{2}{3}\mu \mathrm{tr}\left(\boldsymbol{\nabla}\,\boldsymbol{u}\right)\mathbf{1}\right]. \end{aligned} \tag{1.5}$$

For compressible flows, an EOS linking the pressure to the density and the total energy field is added to Equation (1.5). Making the assumption of *barotropic* flow, the model for the EOS reads:

$$p = \mathcal{P}\left(\rho\right). \tag{1.6}$$

Tait's EOS (see e.g. Hayward (1967)), also known as stiffened gas EOS, is commonly used to model liquids:

$$\mathcal{P}\left(\rho\right) = p_0 + \frac{\rho_0 c_0^2}{\gamma}\left(\frac{\rho}{\rho_0} - 1\right)^{\gamma}, \tag{1.7}$$

where ρ_0 is the reference density, and p_0 and c_0 are its corresponding reference pressure and reference speed of sound, and γ is the Laplace coefficient.

For incompressible flows, the fluid flow is assumed to have a zero divergence, and the density is either deduced from the energy equation or directly supposed to be constant over space and time.

Finally, Equations (1.5) are completed by boundary conditions in time and space (i.e. initial condition and boundary conditions when the space domain is bounded).

1.1.2 Lagrangian and Eulerian approaches

Equations (1.5) are written in a *conservative* form meaning that all terms are written as time or space divergence, and thus give boundary terms when integrating over a time×space element. This form is mandatory when dealing with non-regular solutions with shocks. For regular solutions, Equations (1.5) are equivalent to the *non-conservative* form:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \boldsymbol{\nabla}\rho \cdot \boldsymbol{u} &= -\rho \mathrm{div}\left(\boldsymbol{u}\right), \\ \rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla}\boldsymbol{u} \cdot \boldsymbol{u}\right) &= \boldsymbol{f} - \boldsymbol{\nabla} p + \mathbf{div}\left(\mu\left(\boldsymbol{\nabla}\,\boldsymbol{u} + \boldsymbol{\nabla}\,\boldsymbol{u}^T\right) - \tfrac{2}{3}\mu \mathrm{tr}\left(\boldsymbol{\nabla}\,\boldsymbol{u}\right)\mathbf{1}\right). \end{aligned} \tag{1.8}$$

When studying the time evolution of a transported variable $\psi\left(\boldsymbol{x},\, t\right)$, one can adopt an Eulerian point of view, that is to look at $\frac{\partial}{\partial t}\psi\left(\boldsymbol{x},\, t\right)$, the space variable $\boldsymbol{x}$ being fixed.

As for the Lagrangian point of view, one looks at the time evolution of variable ψ following the fluid particles, i.e. along the particle path $\boldsymbol{x}(t)$ defined by $\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t) = \boldsymbol{u}\left(\boldsymbol{x}(t), t\right)$ and with an initial condition. Thus:

$$\frac{\mathrm{d}}{\mathrm{d}t}\psi\left(\boldsymbol{x}(t),\, t\right) = \frac{\partial}{\partial t}\psi\left(\boldsymbol{x}(t),\, t\right) + \boldsymbol{\nabla}\psi \cdot \boldsymbol{u}, \tag{1.9}$$

is called the Lagrangian derivative or again the particle derivative. In the sequel, for the sake of simplicity, we write $\frac{\mathrm{d}}{\mathrm{d}t}\psi\left(\boldsymbol{x}(t),\, t\right)$ as $\frac{\mathrm{d}\psi}{\mathrm{d}t}$.

The *non-conservative* form of Navier–Stokes Equations (1.8) therefore have a Lagrangian writing:

$$\begin{aligned} \frac{\mathrm{d}\rho}{\mathrm{d}t} &= -\rho \mathrm{div}\,(\boldsymbol{u}), \\ \rho\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} &= \boldsymbol{f} - \boldsymbol{\nabla} p + \mathbf{div}\,\left[\mu\left(\boldsymbol{\nabla}\,\boldsymbol{u} + \boldsymbol{\nabla}\,\boldsymbol{u}^T\right) - \tfrac{2}{3}\mu \mathrm{tr}\,(\boldsymbol{\nabla}\,\boldsymbol{u})\,\mathbf{1}\right]. \end{aligned} \tag{1.10}$$

1.1.3 Free-surface and some examples

A free-surface ∂F_s is a material surface *free* to move with the fluid. It is thus a Lagrangian surface. Mathematically speaking, the surface at time t can be defined as the kernel of a function $\phi_s\,(\boldsymbol{x}, t)$:

$$\boldsymbol{x} \in \partial F_s \text{ i.f.f. } \phi_s\,(\boldsymbol{x}, t) = 0. \tag{1.11}$$

For instance, for a regular free surface whose elevation Z_s can be determined only with x and y components we can define $\phi_s\,(\boldsymbol{x}, t) = z - Z_s\,(x, y, t)$. Then $\boldsymbol{n}_s := \boldsymbol{\nabla}\phi_s$ is a vector field normal to the free surface. Therefore, taking a point $\boldsymbol{x}_s(t)$ at the free surface, moving at the fluid velocity (which is also the surface velocity), we get the kinematic condition:

$$\frac{\mathrm{d}\phi_s\,(\boldsymbol{x}_s(t), t)}{\mathrm{d}t} = 0 = \frac{\partial \phi_s}{\partial t} + \boldsymbol{n}_s \cdot \boldsymbol{u}_s. \tag{1.12}$$

For a regular free surface, we get $\frac{\partial Z_s}{\partial t} = \boldsymbol{n}_s \cdot \boldsymbol{u}_s$. The evolution of the surface is therefore to be tracked if one wants to study the fluid below the free-surface only. This is straightforward for a purely Lagrangian method, and can also be done with an arbitrary Lagrangian–Eulerian approach introduced in §1.2.

Various applications can benefit from numerical simulation of free-surface flows, and an exhaustive list is out of the scope of this document. We illustrate it here with some examples we encountered.

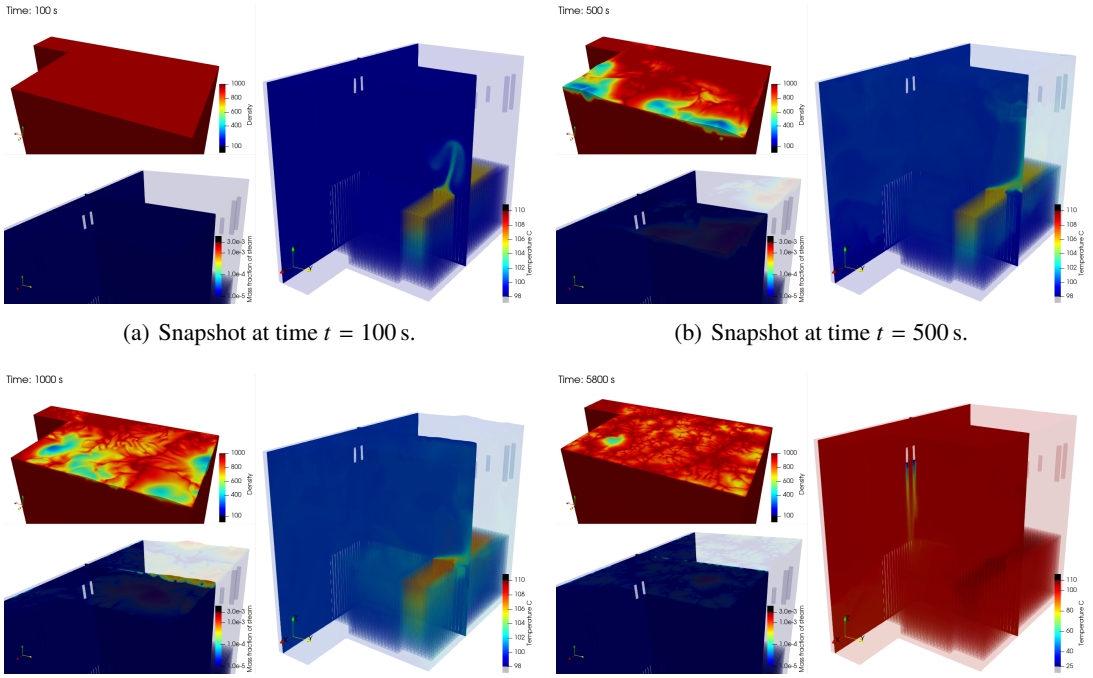
The most common free-surface is seen between water and air, with wave propagation. Waves may be a concern for engineers to design dikes, offshore platforms, but also an opportunity to retrieve some energy from them.

Free surface in pools are also of interest. For the swimmer at the Olympic Games, the interaction of wakes with the free surface plays a key role in the athletes' performance. In another context, pools are used to store fuel assemblies after having used them in nuclear generators. The behaviour of these pools in the event that pumps are out of service have been studied in details since the Fukushima accident as shown in Figure 1.1.

When studying long term nuclear waste storage, one can use bitumen package. One important question for the safety is therefore how does the package behave under fire conditions. Heated, the bitumen of the package melts and behaves as a free-surface flow as displayed on Figure 1.2.

Another smooth free-surface can be observed in the Berre lagoon, and its modelling is of importance to study the evolution of the brackish water ecosystem in this area connected to the Mediterranean Sea as illustrated in Figure 1.3. For this type of turbulent flows, momentum exchanges and turbulent kinetic energy next to the free-surface are key challenging quantities to model (see Ferrand, 2009). The modelling of turbulence is out of the scope of this document, and is briefly discussed in Chapter 2 to present how to impose standard wall functions in the SPH method.

Open boundary conditions when studying atmospheric flows is a key challenge when modelling at CFD scale the flow around an industrial site. Namely, the boundary condition on the top of the simulation domain can lead to non-physical behaviour if we impose a zero flux



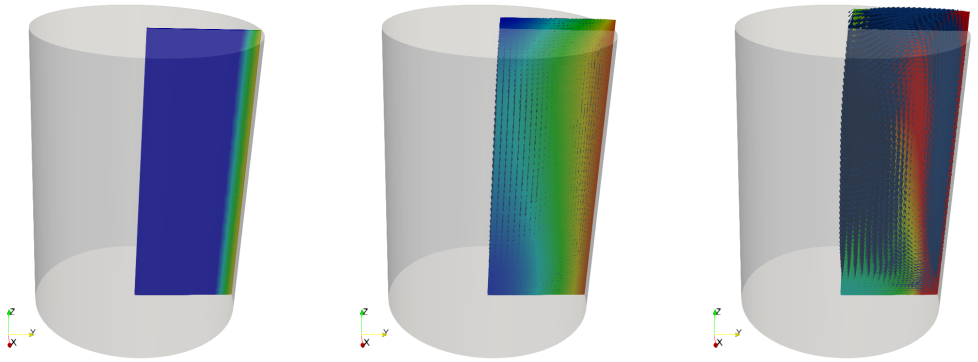
(a) Snapshot at time $t = 100\,\mathrm{s}$.

(b) Snapshot at time $t = 500\,\mathrm{s}$.

(c) Snapshot at time $t = 1000\,\mathrm{s}$.

(d) Snapshot at time $t = 2000\,\mathrm{s}$.

Figure 1.1: Simulation of a fuel assembly storage pool free-surface evolution performed by Romain Camy (EDF DT) using code_saturne. When cooling pumps stop working, heat is released with natural convection only. Boiling may occur and there is a risk of pump failure.

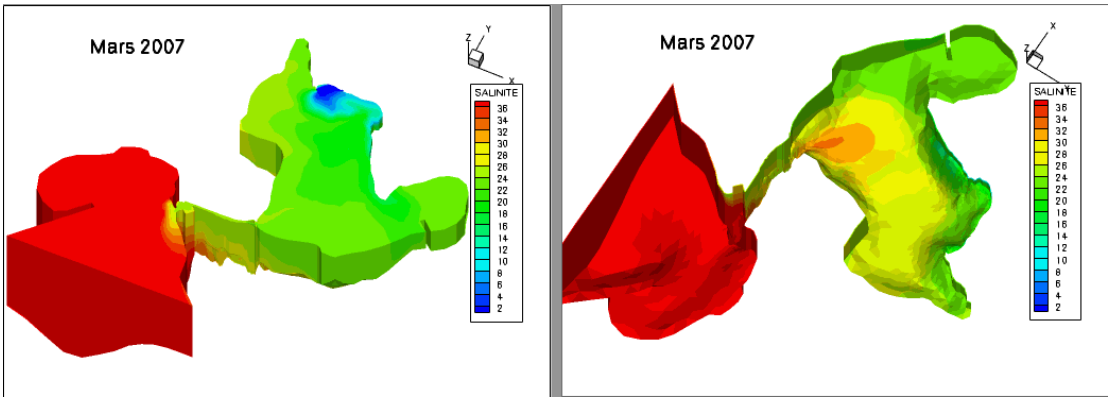


(a) Snapshot at time $t = 900\,\mathrm{s}$.

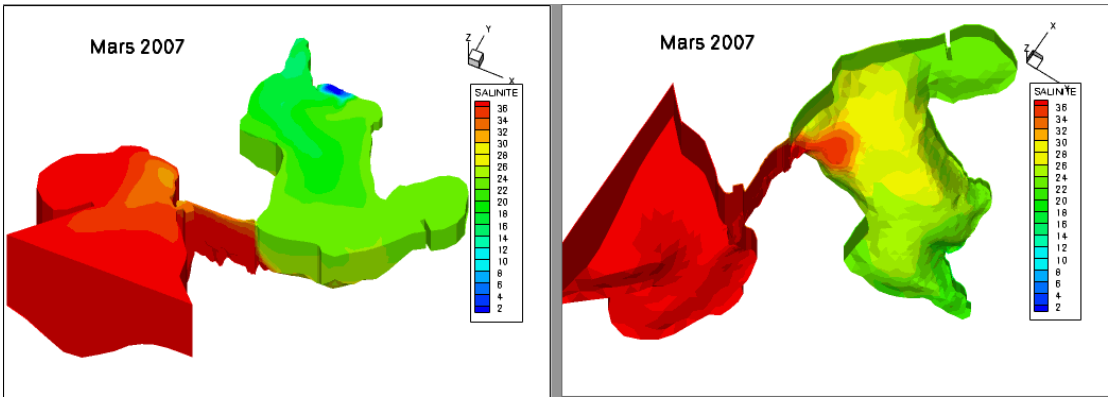
(b) Snapshot at time $t = 5400\,\mathrm{s}$.

(c) Snapshot at time $t = 7200\,\mathrm{s}$.

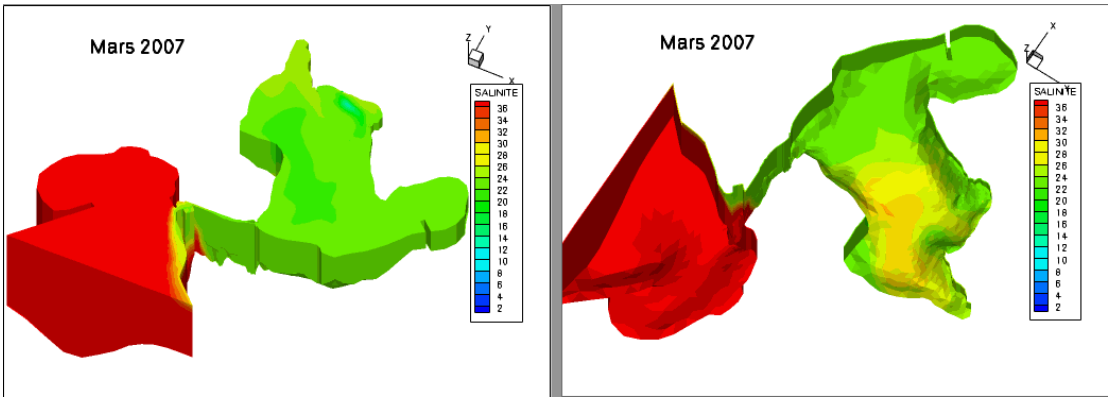
Figure 1.2: Simulation of a slice of a 2 kg bitumen pot which is heated from the side using code_saturne. This numerical model aims at reproducing the experimental temperature behaviour of bituminous sludge. At room temperature, the transfers are very limited and heat exchange takes place by conduction through the material. The density change is significant from a threshold temperature rise. The presence of a density gradient is the establishment condition of a convection loop. This case consists in studying a fluid flow, which is pseudo-Newtonian through a Bingham law, inside a cylindrical bitumen package using ALE to catch the upper moving boundary interface. The initial cylindrical domain is represented in transparent grey to show the swelling of the sludge.



(a) Snapshot when fresh water is realesed in the lagoon.



(b) Snapshot when the Mediterranean Sea is entering the lagoon.



(c) Snapshot when the channel is stratified.

Figure 1.3: Simulation of the exchange of salinity between the Berre lagoon and the Mediterranean Sea through the channel between them, using TELEMAC-3D. Salinity may stratify due to buoyant effect in the lagoon or mix when wind shear stress is sufficient. View from the top on the left-hand-side, view from the bottom on the right-hand-side.

such as in a wind tunnel experiment. Physically, the horizontal flow over a cliff can accelerate, but vertical velocity will also be generated, making velocity boundary conditions at the top of the domain complex to specify. One can therefore consider the top of the domain as a free surface of imposed pressure.

All the previous examples can be modelled using arbitrary Eulerian-Lagrangian approach introduced in section 1.2 or using fully Lagrangian approach such as SPH introduced in section 1.3.

1.2 Finite volume and arbitrary Lagrangian-Eulerian approach

1.2.1 Basic finite volume formulations for Newtonian fluids

The finite volume method has been introduced relatively recently (see e.g. Patankar (1980)) compared to the finite elements methods; a complete and valuable overview of the method can be found in Leveque (1992), Eymard et al. (2000) and Toro (2013). When dealing with conservative Equations (1.5), finite volume is a natural method that brings space and time integration over a subset $V_c \times [t^n,\, t^{n+1}]$ into boundary terms[1]:

$$\begin{aligned}
\int_{V_c} \rho\left(\boldsymbol{x},\, t^{n+1}\right) - \rho\left(\boldsymbol{x},\, t^{n}\right) \mathrm{d}\Omega + \int_{\partial V_c} \int_{t^n}^{t^{n+1}} \rho \boldsymbol{u} \mathrm{d}t \cdot \mathrm{d}\boldsymbol{S} = 0, \\
\int_{V_c} \rho \boldsymbol{u}\left(\boldsymbol{x},\, t^{n+1}\right) - \rho \boldsymbol{u}\left(\boldsymbol{x},\, t^{n}\right) \mathrm{d}\Omega + \int_{\partial V_c} \int_{t^n}^{t^{n+1}} \boldsymbol{u} \otimes \rho\, \boldsymbol{u} \mathrm{d}t \cdot \mathrm{d}\boldsymbol{S} = \int_{\partial V_c} \int_{t^n}^{t^{n+1}} \boldsymbol{\sigma} \mathrm{d}t \cdot \mathrm{d}\boldsymbol{S} \\
+ \int_{V_c} \int_{t^n}^{t^{n+1}} \boldsymbol{f}\, \mathrm{d}t \mathrm{d}\Omega.
\end{aligned} \tag{1.13}$$

Mean space values over cell V_c are denoted by:

$$(.)_c := \frac{1}{V_c} \int_{V_c} (.) \mathrm{d}\Omega, \tag{1.14}$$

with V_c being the cell measure. Similarly, mean time values over time interval $[t^n,\, t^{n+1}]$ are denoted by:

$$(.)|_n^{n+1} := \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} (.) \mathrm{d}t, \tag{1.15}$$

with Δt being the interval measure. Moreover, time values at time t^n (respectively t^{n+1}) are written with the superscript $(.)^n$ (respectively $(.)^{n+1}$). The time interval Δt is $t^{n+1} - t^n$. Using these notations (1.13) reads:

$$\begin{aligned}
V_c \left(\rho_c^{n+1} - \rho_c^n\right) + \Delta t \int_{\partial V_c} \rho \boldsymbol{u}|_n^{n+1} \cdot \mathrm{d}\boldsymbol{S} = 0, \\
V_c \left((\rho \boldsymbol{u})_c^{n+1} - (\rho \boldsymbol{u})_c^n\right) + \Delta t \int_{\partial V_c} \boldsymbol{u} \otimes \rho\, \boldsymbol{u}|_n^{n+1} \cdot \mathrm{d}\boldsymbol{S} = \Delta t \left[V_c\, \boldsymbol{f}_c|_n^{n+1} + \int_{\partial V_c} \boldsymbol{\sigma}|_n^{n+1} \cdot \mathrm{d}\boldsymbol{S}\right].
\end{aligned} \tag{1.16}$$

Therefore, space Ω is partitioned into a finite number of cells, supposed in the following to be polyhedra: $\Omega = \bigcup_c \Omega_c$. Geometrical quantities of interest, such as faces and face centres, nodes, cell centres are displayed on Figure 1.4.

[1] Note that the time integration of the time derivative using the fundamental theorem of calculus in 1-D is analogous to the space integration of the divergence using the divergence theorem in 3-D.

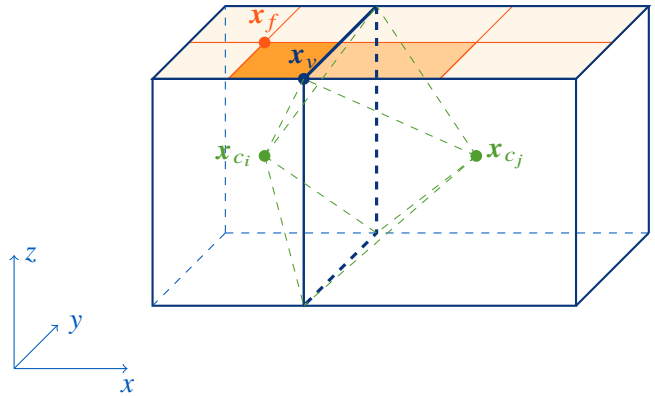


Figure 1.4: Labelling of information used for a mesh in code_saturne. In green, $\boldsymbol{x}_{c_i}$ is the barycentre of one cell c_i whereas $\boldsymbol{x}_{c_j}$ is the one of c_j. Each face f is split into portions of faces associated to a vertex. The portion of face f associated to a vertex v is drawn in solid orange, and is defined by $\boldsymbol{x}_f$, the barycentre of f, the middle of the two edges of f passing through vertex v, and $\boldsymbol{x}_v$ itself.

Equations (1.16) lead to the definition of the extensive quantities cell volume V_c, cell mass M_c, cell momentum $\boldsymbol{Q}_c$ and cell external force $\boldsymbol{F}_c$ (they are denoted by capital letters):

$$\begin{aligned} V_c &:= \int_{V_c} \mathrm{d}\Omega, & \boldsymbol{F}_c &:= \int_{V_c} \boldsymbol{f}\,\mathrm{d}\Omega, \\ M_c &:= \int_{V_c} \rho\,\mathrm{d}\Omega, & \boldsymbol{Q}_c &:= \int_{V_c} \rho\boldsymbol{u}\,\mathrm{d}\Omega. \end{aligned} \tag{1.17}$$

As for the intensive quantities, let ρ_c be considered the cell-averaged density, let $\boldsymbol{f}_c$ be the cell-averaged force and let $\boldsymbol{u}_c$ be the density-weighted cell-averaged velocity, defined as:

$$\begin{aligned} \rho_c &:= \frac{M_c}{V_c}, \\ \boldsymbol{f}_c &:= \frac{\boldsymbol{F}_c}{V_c}, \\ \boldsymbol{u}_c &:= \frac{\boldsymbol{Q}_c}{M_c}. \end{aligned} \tag{1.18}$$

Let us first start assuming the meshed domain is motionless, integrated mass equation of (1.13) rewrites:

$$M_c^{n+1} - M_c^n + \Delta t \sum_{f\in\mathcal{F}_c} (\rho\,\boldsymbol{u})_f\big|_n^{n+1} \cdot \boldsymbol{S}_{c>f} = 0, \tag{1.19}$$

where $\mathcal{F}_c$ is the ensemble of all the planar polygonal faces of cell c and $\boldsymbol{S}_{c>f}$ is the *outward* surface vector. The surface measure is denoted by $S_f := \left|\boldsymbol{S}_{c>f}\right|$. Note that the quantity $(.)_f$ coming from the integration of the divergence is defined as the average (in space) over the face of $(.)$:

$$(.)_f := \frac{1}{S_f}\int_f (.)\mathrm{d}S, \tag{1.20}$$

thus the mass flux over a face reads $\boldsymbol{q}_f\big|_n^{n+1} \cdot \boldsymbol{S}_{c>f} = (\rho \boldsymbol{u})_f\big|_n^{n+1} \cdot \boldsymbol{S}_{c>f}$, $\boldsymbol{q}$ being the local momentum. The expression of mean face values with respect to the cell values will be given by the space discretisation.

Performing the same computation for the momentum equation of (1.13) gives:

$$\boldsymbol{Q}_c^{n+1} - \boldsymbol{Q}_c^n + \Delta t \sum_{f \in \mathcal{F}_c} (\boldsymbol{u} \otimes \rho\, \boldsymbol{u} - \boldsymbol{\sigma})_f\big|_n^{n+1} \cdot \boldsymbol{S}_{c>f} = \Delta t\, \boldsymbol{F}_c\big|_n^{n+1} . \tag{1.21}$$

The finite volume method has been very popular for *computational fluid dynamics* (CFD) for both hyperbolic equations for compressible flows and incompressible flows. The goal of numerical modellers is to express face fluxes in terms of cell values.

Discrete version of the differential operators:

In the finite volume method the divergence operator plays a key role, as it gives exact space balance on **fixed** cells when applied on face-averaged field:

$$\mathrm{Div}_c \left\{ \boldsymbol{\Psi}_f \right\} := \frac{1}{V_c} \sum_{f \in \mathcal{F}_c} \boldsymbol{\Psi}_f \cdot \boldsymbol{S}_{c>f} = (\mathrm{div}\,(\boldsymbol{\Psi}))_c . \tag{1.22}$$

Discrete cell gradient of scalars ψ is then defined as divergence of a tensor $\psi \mathbf{1}$:

$$\mathbf{Grad}_c \left\{ \psi_f \right\} := \frac{1}{V_c} \sum_{f \in \mathcal{F}_c} \psi_f \boldsymbol{S}_{c>f} = (\mathbf{div}\,(\psi \mathbf{1}))_c . \tag{1.23}$$

Cell Laplacian operator applied to a scalar field ψ with a diffusivity field K writes:

$$\mathrm{Lap}_c\,(\{K\}\,,\,\{\psi\}) := (\mathrm{div}\,(K \boldsymbol{\nabla} \psi))_c = \frac{1}{V_c} \sum_{f \in \mathcal{F}_c} (K \boldsymbol{\nabla} \psi)_f \cdot \boldsymbol{S}_{c>f}. \tag{1.24}$$

In what follows, a Two-Point Flux Approximation (TPFA, see Eymard et al., 2014) with non-orthogonal reconstruction is used to discretise the Laplacian operator. See Ferrand et al. (2014) and appendix A for the extension to non-Cartesian grids in code_saturne for an anisotropic diffusion problem.

With these notations, the mass and momentum discrete balances over **fixed** cells are rewritten:

$$\frac{\rho_c^{n+1} - \rho_c^n}{\Delta t} + \mathrm{Div}_c \left\{ \boldsymbol{q}_f\big|_n^{n+1} \right\} = 0, \tag{1.25}$$

$$\frac{(\rho \boldsymbol{u})_c^{n+1} - (\rho \boldsymbol{u})_c^n}{\Delta t} + \mathbf{Div}_c \left\{ (\boldsymbol{u} \otimes \boldsymbol{q})_f\big|_n^{n+1} \right\} = \boldsymbol{f}_c\big|_n^{n+1} - \mathbf{Grad}_c \left\{ p_f\big|_n^{n+1} \right\} + \mathbf{Div}_c \left\{ \boldsymbol{\tau}_f\big|_n^{n+1} \right\} . \tag{1.26}$$

Time stepping and space discretisation:

The steps to solve the 3-D Navier–Stokes equations in the legacy algorithm of code_saturne are described in this paragraph. Following pioneer work of Chorin (1968) and Temam (1968), equations are solved in two steps: one prediction step for the velocities from the explicit pressure field, and a correction step, where the continuity equation is used to calculate a change of pressure, which adjusts the mass flux to ensure mass conservation and then update the velocity field. It is also possible to perform sub-iterations over these two steps, as for a Pressure-Implicit with Splitting of Operators (PISO) algorithm (for a description of PISO see Issa et al., 1986; Versteeg and Malalasekera, 2007, for a general presentation of pressure correction algorithms). We will call it PISO-like algorithm and sub-iterations will be denoted by the superscript k starting at 1.

For the prediction step, code_saturne uses by default an implicit Euler first-order scheme (extension to second-order scheme will be presented in the next paragraph), where the viscous shear stress is time-implicit[2], the pressure is time-explicit, and the convective term is linearised and when written in semi-implicit form as follows:

$$\frac{\rho_c^{n+1,k-1}\widetilde{\boldsymbol{u}}_c^k - \rho_c^n \boldsymbol{u}_c^n}{\Delta t} + \mathbf{Div}_c\left\{\left\langle \widetilde{\boldsymbol{u}}^k \right\rangle_f \otimes \boldsymbol{q}_f\big|_n^{n+1,k-1}\right\} = \boldsymbol{f}_c\big|_n^{n+1,k-1} - \mathbf{Grad}_c\left\{\left\langle p\big|_n^{n+1,k-1}\right\rangle_f\right\} + \mathbf{Div}_c\left\{\left(\boldsymbol{\tau}\left(\widetilde{\boldsymbol{u}}^k\right)\right)_f\right\}, \tag{1.27}$$

where $\widetilde{\boldsymbol{u}}^k$ is the predicted velocity field to be solved, and the mass flux $\boldsymbol{q}_f\big|_n^{n+1,k-1} \cdot \boldsymbol{S}_{c>f}$ is initialised by $\boldsymbol{q}_f\big|_n^{n+1,0} \cdot \boldsymbol{S}_{c>f} = \boldsymbol{q}_f\big|_{n-1}^{n} \cdot \boldsymbol{S}_{c>f}$, the mean pressure over interval $[n, n+1]$ is initialised by $p\big|_n^{n+1,0} = p\big|_{n-1}^{n}$, and the density $\rho_c^{n+1,0} = \rho_c^n$. Note that in Eq. (1.27), cell pressure gradient $\mathbf{Grad}_c\left\{\left\langle p\big|_n^{n+1,k-1}\right\rangle_f\right\}$ and external volume force $\boldsymbol{f}_c\big|_n^{n+1,k-1}$ appear at the same time location, so if they are in (partial) balance, they will not create spurious velocities. Face interpolation from cell values is denoted by $\langle . \rangle_f$ and requires a space discretisation scheme. Some standard convective schemes for $\left\langle \widetilde{\boldsymbol{u}}^k \right\rangle_f$ are available in code_saturne: either a full centred scheme with optional slope test limiter (this will be used in §4.4), or a Second Order Linear Upwind (SOLU), or a pure upwind scheme. Some details on convective schemes for scalars are given in Appendix B, with some special treatment to ensure minimum-maximum principle. $\left\langle p\big|_n^{n+1,k-1}\right\rangle_f$ is discretised by a centred scheme.

The unsteady term of Eq. (1.27) is algebraically rewritten using the conservation of mass:

$$\frac{\rho_c^{n+1,k-1}\widetilde{\boldsymbol{u}}_c^k - \rho_c^n \boldsymbol{u}_c^n}{\Delta t} = \rho_c^n \frac{\left(\widetilde{\boldsymbol{u}}_c^k - \boldsymbol{u}_c^n\right)}{\Delta t} - \mathrm{Div}_c\left\{\boldsymbol{q}_f\big|_n^{n+1,k-1}\right\}\widetilde{\boldsymbol{u}}_c^k. \tag{1.28}$$

Before the correction step, the density might be updated through an equation of state to be specified by the user (e.g., $\rho_c^{n+1,k}$ could be a function of buoyant scalars such as temperature or salinity). A fully compressible algorithm is also available and presented in Colas et al. (2019).

During the correction step, a simplified momentum equation is solved under the mass balance constraint:

$$\begin{aligned} \rho^{n+1,k-1}\frac{\boldsymbol{u}^{n+1,k} - \widetilde{\boldsymbol{u}}^k}{\Delta t} + \boldsymbol{\nabla}\phi^k &= \delta\boldsymbol{f}^k, \\ \boldsymbol{q}_f\big|_n^{n+1,k} \cdot \boldsymbol{S}_{c>f} &= \left\langle \rho^{n+1,k-1}\boldsymbol{u}^{n+1,k}\right\rangle_f \cdot \boldsymbol{S}_{c>f}, \\ \frac{\rho_c^{n+1,k} - \rho_c^n}{\Delta t} + \mathrm{Div}_c\left\{\boldsymbol{q}_f\big|_n^{n+1,k}\right\} &= 0, \end{aligned} \tag{1.29}$$

where potential $\phi^k = p\big|_n^{n+1,k} - p\big|_n^{n+1,k-1}$ is the pressure increment and $\delta\boldsymbol{f}^k = \boldsymbol{f}\big|_n^{n+1,k} - \boldsymbol{f}\big|_n^{n+1,k-1}$ is the external volume force increment. $\left\langle \rho^{n+1,k-1}\boldsymbol{u}^{n+1,k}\right\rangle_f$ is discretised by a centred scheme.

[2]The transpose of the velocity gradient and the isotropic term are in fact time-implicit with an iterative process used to reconstruct terms due to the non-orthogonalities of the mesh if present, for more details see Archambeau et al. (2004).

From Eq. (1.29), a Poisson equation is written for ϕ^k:

$$-\mathrm{Lap}_c\left(\{\Delta t\},\ \{\phi^k\}\right) = -\mathrm{Div}_c\left\{\Delta t \boldsymbol{\nabla}_f \phi^k\right\} = -\mathrm{Div}_c\left\{\left\langle \rho^{n+1,k-1}\widetilde{\boldsymbol{u}}^k + \Delta t \delta \boldsymbol{f}^k \right\rangle_f\right\} - \frac{\rho_c^{n+1,k} - \rho_c^n}{\Delta t}. \tag{1.30}$$

The Rhie and Chow filter (see Rhie and Chow, 1983) can be added to the right-hand side of Eq. (1.30) (see Archambeau et al., 2004). This filter is necessary due to the co-located algorithm. Term $\mathrm{Div}_c\left\{\left\langle \rho^{n+1,k-1}\widetilde{\boldsymbol{u}}^k \right\rangle_f\right\}$ requires face expression of a cell quantity (namely $\rho_c^{n+1,k-1}\widetilde{\boldsymbol{u}}_c^k$), which contains somehow $\Delta t \mathbf{Grad}_c\left\{\left\langle p|_n^{n+1,k}\right\rangle_f\right\}$. The kernel of the discrete operator $\mathrm{Div}_c\left\{\langle \mathbf{Grad}\,\{.\}\rangle_f\right\}$ is far too big, for instance on a 1-D Cartesian grid, +1, −1, +1 field, called checker-board is in the kernel of the operator. On the other hand the Laplacian operator obtained from a TPFA scheme smoothes checker-board field. Eq. (1.30) is therefore rewritten as:

$$-\mathrm{Lap}_c\left(\{\Delta t\},\ \left\{\underbrace{\phi^k + p|_n^{n+1,k-1}}_{=p|_n^{n+1,k}}\right\}\right) = -\mathrm{Div}_c\left\{\left\langle \rho^{n+1,k-1}\widetilde{\boldsymbol{u}}^k + \Delta t \boldsymbol{\nabla}\, p|_n^{n+1,k-1} + \Delta t \delta \boldsymbol{f}^k \right\rangle_f\right\} - \frac{\rho_c^{n+1,k} - \rho_c^n}{\Delta t}. \tag{1.31}$$

Space discretisation of the Laplacian operator is detailed in Ferrand et al. (2014) and presented here in appendix A for an anisotropic diffusion. Note that the first line of Eq. (1.29) is then used to update the cell velocity field $\boldsymbol{u}_c^{n+1,k}$.

An extension to second order with time-staggered time stepping:

We present in this Section a conservative second-order time scheme for variable density flows with a low Mach number we implemented in code_saturne. To reach second-order accuracy over time, the Crank-Nicolson scheme is often used due to its simplicity and low storage requirement. Most of the simulation codes actually use an implicit-explicit combination, thus combining an implicit scheme such as Crank-Nicolson or Backward-Differentiation Formula with an explicit one, such as Adams-Bashforth or Leap-Frog (Edwards et al., 2011).

The analysis of such combined schemes tends to be complex, as the stability condition of the combination usually differs from the stability condition of the individual schemes. For instance, the combination of Crank-Nicolson with Leap-Frog has been studied for decades but was proven to be stable under the time step conditions suggested by linear stability theory only recently (Layton and Trenchea, 2012).

The proposed scheme we integrated in code_saturne is similar to the one proposed by Wall et al. (2002). However, here, the scheme is derived on different grounds, with the finite volume discretisation perspective.

In the following, the time stepping is defined by the parameter θ, bounded in $[0, 1]$. When we consider the case $\theta = 1$, we fall back to the implicit Euler scheme presented earlier (1^{st} order in time), and, if $\theta = 1/2$, we get the Crank–Nicolson scheme (2^{nd} order in time).

For any field ψ, the time interpolation between times n and $n+1$ writes:

$$\Theta\left(\psi^n,\ \psi^{n+1}\right) = (1-\theta)\,\psi^n + \theta\psi^{n+1}. \tag{1.32}$$

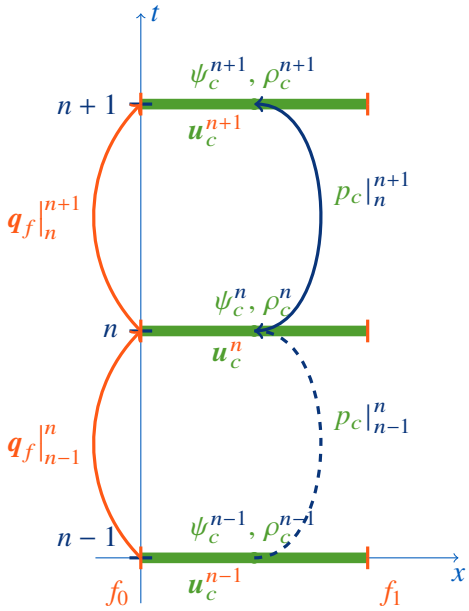


Figure 1.5: First-order in time Euler implicit time stepping for the finite volume approach, variables localisation. Mass fluxes $\boldsymbol{q}_f|_n^{n+1}$ are stored at faces and bring cell density from time n to time $n+1$. They are computed with respect to $\boldsymbol{u}^{n+1}$.

We use the notation $\Theta\left(\psi^n, \psi^{n+1}\right)$ to make clear what are the two boundaries of the time interpolation during the iterative process.

The dual time interval around time n is denoted by $[n-1+\theta, n+\theta]$, and is of length:

$$\Delta t|_{n-1+\theta}^{n+\theta} = \Theta\left(\Delta t|_{n-1}^n,\ \Delta t|_n^{n+1}\right),$$

where $\Delta t|_{n-1}^n$ is the length of time interval $[n-1, n]$ and $\Delta t|_n^{n+1}$ the one of $[n, n+1]$.

In case of non-equal time steps, $\widetilde{\theta} = \theta \frac{\Delta t|_n^{n+1}}{\Delta t|_{n-1+\theta}^{n+\theta}}$ is also defined, and the time interpolation between time intervals $[n-1, n]$ and $[n, n+1]$ writes:

$$\widetilde{\Theta}\left(\boldsymbol{q}|_{n-1}^n,\ \boldsymbol{q}|_n^{n+1}\right) := \frac{\Theta\left(\Delta t|_{n-1}^n\, \boldsymbol{q}|_{n-1}^n,\ \Delta t|_n^{n+1}\, \boldsymbol{q}|_n^{n+1}\right)}{\Delta t|_{n-1+\theta}^{n+\theta}} = \left(1-\widetilde{\theta}\right) \boldsymbol{q}|_{n-1}^n + \widetilde{\theta}\, \boldsymbol{q}|_n^{n+1}.$$

Note that if the time steps are equal, then $\widetilde{\theta} = \theta$, and that is what is assumed in the following.

R On the time localisation of variables. Mass balance is chosen to be done on primal mesh cell between primal time steps n and $n+1$. Therefore, since they are linked by the equation of state, scalar mass fractions ψ and density ρ are stored at the same space and the same time location. The mass balance imposes that the mass fluxes denoted by $\boldsymbol{q}$ are stored on time interval $[n, n+1]$ and on the primal mesh faces. Note that the time interval $[n, n+1]$ is the dual space of time $n+\frac{1}{2}$. Therefore, the momentum equation will be solved between times $n-\frac{1}{2}$ and $n+\frac{1}{2}$ when $\theta = 1/2$.

Initialisation: The first sub-iteration is still denoted by $k=1$ and therefore some values have to be initialised for $k=0$. For the density at time n, $\rho_c^{n,0} = \rho_c^{n-1}$ and for all other k, $\rho_c^{n,k} = \rho_c^n$. For the density at time $n+1$, $\rho_c^{n+1,0} = \rho_c^n$. For the mass flux, $\boldsymbol{q}_f|_n^{n+1,0} \cdot \boldsymbol{S}_{c>f} = \boldsymbol{q}_f|_{n-1}^n \cdot \boldsymbol{S}_{c>f}$.

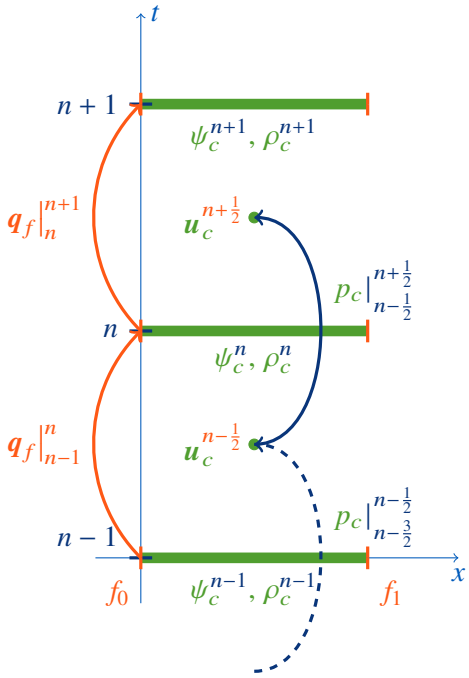


Figure 1.6: Second-order in time time stepping for the finite volume approach, variables localisation.

For the pressure, $p|_{n-1+\theta}^{n+\theta,0} = p|_{n-2+\theta}^{n-1+\theta}$. As a side note, at any iteration, the mass flux should verify a mass balance equation for all cells:

$$\Delta t|_n^{n+1} \, \mathrm{Div}_c \left\{ \boldsymbol{q}_f|_n^{n+1,k} \right\} = - \left(\rho_c^{n+1,k} - \rho_c^n \right).$$

Buoyant scalars step, solve $\psi^{n+1,k}$ and update $\rho^{n+1,k}$: During this stage, the values of the buoyant scalars ψ and of the density ρ are updated.

Firstly, the transport equation of buoyant scalars is solved:

$$\frac{\rho_c^{n+1,k-1}\psi_c^{n+1,k} - \rho_c^{n,k-1}\psi_c^n}{\Delta t|_n^{n+1}} + \mathrm{Div}_c \left\{ \left\langle \Theta\left(\psi^n, \psi^{n+1,k}\right) \right\rangle_f \boldsymbol{q}_f|_n^{n+1,k-1} \right\} = rhs_c^{n+\theta,k}, \quad (1.33)$$

where the right-hand-side term contains the source terms and the diffusion. For the first sub-iteration in k, namely $k = 1$, the reader is reminded that $\rho^{n,k-1}$ is ρ^{n-1}. Therefore, there is a time shift between ρ and ψ for the first sub-iteration.

Secondly, the density is updated with the equation of state: $\rho^{n+1,k} = \rho\left(\psi^{n+1,k}\right)$. Note that this new density is not balanced by any mass flux yet. This mass balance will be performed in the upcoming correction step, which ensures mass conservation.

Momentum step, prediction of the velocity, solve $\widetilde{\boldsymbol{u}}^k$: During this stage, the values of the predicted velocity $\widetilde{\boldsymbol{u}}$ are estimated. With the velocity-pressure coupling in mind, the prediction step is the preliminary one. The outcome of the velocity-pressure coupling is the mass flux $\boldsymbol{q}|_n^{n+1,k}$, which is naturally centred on the time interval $[n, n+1]$. Thus, the momentum equation

is solved on the time interval $[n-1+\theta, n+\theta]$:

$$\begin{aligned}
&\frac{\Theta\left(\rho_c^n,\, \rho_c^{n+1,k-1}\right)\widetilde{\boldsymbol{u}}_c^k - \Theta\left(\rho_c^{n-1},\, \rho_c^{n,k-1}\right)\boldsymbol{u}_c^{n-1+\theta}}{\Delta t|_{n-1+\theta}^{n+\theta}} + \mathbf{Div}_c\left\{\left\langle\widetilde{\Theta}\left(\boldsymbol{u}^{n-1+\theta},\, \widetilde{\boldsymbol{u}}^k\right)\right\rangle_f \otimes \boldsymbol{q}_f|_{n-1+\theta}^{n+\theta,k-1}\right\} \\
= \;& + \boldsymbol{f}_c|_{n-1+\theta}^{n+\theta,k-1} - \mathbf{Grad}_c\left\{\left\langle p|_{n-1+\theta}^{n+\theta,k-1}\right\rangle_f\right\} + \mathbf{Div}_c\left\{\left\langle\boldsymbol{\tau}\left(\widetilde{\Theta}\left(\boldsymbol{u}^{n-1+\theta},\, \widetilde{\boldsymbol{u}}^k\right)\right)\right\rangle_f\right\}
\end{aligned} \tag{1.34}$$

The mass flux $\boldsymbol{q}|_{n-1+\theta}^{n+\theta,k-1}$ is defined by $\boldsymbol{q}|_{n-1+\theta}^{n+\theta,k-1} = \widetilde{\Theta}\left(\boldsymbol{q}|_{n-1}^{n},\, \boldsymbol{q}|_{n}^{n+1,k-1}\right)$ and therefore verifies the mass balance with $\frac{\rho^{n+\theta,k-1}-\rho^{n-1+\theta}}{\Delta t|_{n-1+\theta}^{n+\theta}} = (1-\theta)\frac{\rho^n-\rho^{n-1}}{\Delta t|_n^{n+1}} + \theta\frac{\rho^{n+1,k-1}-\rho^n}{\Delta t|_{n-1}^{n}}$.

Once again, unsteady term and convection of (1.34) can be rewritten as:

$$\begin{aligned}
&\Theta\left(\rho_c^{n-1},\, \rho_c^{n,k-1}\right)\frac{\widetilde{\boldsymbol{u}}_c^k - \boldsymbol{u}_c^{n-1+\theta}}{\Delta t|_{n-1+\theta}^{n+\theta}} + \mathbf{Div}_c\left\{\left\langle\Theta\left(\boldsymbol{u}^{n-1+\theta},\, \widetilde{\boldsymbol{u}}^k\right)\right\rangle_f \otimes \boldsymbol{q}_f|_{n-1+\theta}^{n+\theta,k-1}\right\} \\
&-\mathrm{Div}_c\left\{\boldsymbol{q}_f|_{n-1+\theta}^{n+\theta,k-1}\right\}\widetilde{\boldsymbol{u}}_c^k,
\end{aligned} \tag{1.35}$$

this algebraic manipulation facilitates the observation that the obtained matrix for an upwind scheme on $\widetilde{\boldsymbol{u}}^k$ is an M-matrix.

Mass step, correction of the velocity, solve ϕ^k: During this stage, the values of the pressure increment ϕ, of the resulting pressure p, of the corrected velocity $\boldsymbol{u}$ and of the mass flux are updated.

With the velocity-pressure coupling in mind, the correction step is the concluding one. The resulting mass flux will balance the density obtained in the buoyancy step.

$$\begin{cases}
\Theta\left(\rho^{n,k-1},\, \rho^{n+1,k-1}\right)\dfrac{\boldsymbol{u}^{n+\theta,k} - \widetilde{\boldsymbol{u}}^k}{\Delta t|_{n-1+\theta}^{n+\theta}} + \boldsymbol{\nabla}\phi^k = \delta\boldsymbol{f}^k, \\
\mathrm{Div}_c\left\{\boldsymbol{q}_f|_n^{n+1,k}\right\} + \dfrac{\rho_c^{n+1,k} - \rho_c^n}{\Delta t|_n^{n+1}} = 0,
\end{cases} \tag{1.36}$$

After solving ϕ, the pressure over the time interval $[n-1+\theta, n+\theta]$ is updated with:

$$p|_{n-1+\theta}^{n+\theta,k} = p|_{n-1+\theta}^{n+\theta,k-1} + \phi^k. \tag{1.37}$$

Taking into account the Rhie and Chow filter, and assembling pressure gradients and forces which might be in partial equilibrium, the mass flux is therefore given by

$$\begin{aligned}
\boldsymbol{q}_f|_n^{n+1,k} &= \left\langle\Theta\left(\rho^{n,k-1},\, \rho^{n+1,k-1}\right)\boldsymbol{u}^{n+\theta,k}\right\rangle_f \\
&= \left\langle\Theta\left(\rho^{n,k-1},\, \rho^{n+1,k-1}\right)\widetilde{\boldsymbol{u}}^k + \Delta t\left(\mathbf{Grad}\left\{\left\langle p|_{n-1+\theta}^{n+\theta,k-1}\right\rangle\right\} - \boldsymbol{f}|_{n-1+\theta}^{n+\theta,k-1}\right)\right\rangle_f \\
&\quad - \Delta t|_{n-1+\theta}^{n+\theta}\left(\mathbf{grad}_f\; p|_{n-1+\theta}^{n+\theta,k} - \left\langle\boldsymbol{f}|_{n-1+\theta}^{n+\theta,k}\right\rangle_f\right),
\end{aligned}$$

which verifies the mass balance with $\frac{\rho_c^{n+1,k}-\rho_c^n}{\Delta t|_n^{n+1}}$. Note that no space indices are written for the first equation of (1.36), because this equation is used both for cells (to update cell velocity) and faces (to update mass fluxes).

At the end of this step, one gets the new velocity which is living at time $n+\theta$.

1.2.2 Arbitrary Lagrangian-Eulerian formulation

We now perform balances on moving subsets $V_c(t)$, let us switch to the arbitrary Lagrangian-Eulerian (ALE) approach. First it is necessary to formulate the equations governing the conservation of momentum in the domain, $\Omega(t)$, which may also move in time. We denote the velocity of mesh displacement field $\boldsymbol{v}(\boldsymbol{x}, t)$, for points $\boldsymbol{x}(\boldsymbol{x}_0, t)$ originally located at $\boldsymbol{x}_0$ and moving in the domain. The mesh velocity of $\Omega(t)$ is thus written:

$$\boldsymbol{v}(\boldsymbol{x}, t) = \frac{\partial \boldsymbol{x}(\boldsymbol{x}_0, t)}{\partial t}. \tag{1.38}$$

We now consider a sub-domain $V_c(t)$ of $\Omega(t)$. This domain corresponds to a cell of the mesh previously described. Note that there is no *a priori* guarantee that planar faces of V_c are kept planar over time. Using the Leibniz integration rule, we can evaluate the variation of an arbitrary tensor field $\boldsymbol{\Psi}$ in $V_c(t)$ over time. This is written as:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V_c(t)} \boldsymbol{\Psi}\mathrm{d}\Omega = \int_{V_c(t)} \frac{\partial \boldsymbol{\Psi}}{\partial t}\mathrm{d}\Omega + \int_{\partial V_c(t)} \boldsymbol{\Psi} \otimes \boldsymbol{v} \cdot \mathbf{d}\boldsymbol{S}. \tag{1.39}$$

It is important to note that $\frac{\mathrm{d}}{\mathrm{d}t}$ represents the total derivative of the integral quantity, that follows the motion of $V_c(t)$.

Applying the Leibniz integral rule, Eq. (1.39), to the unity scalar field gives the way cell volumes evolve:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V_c(t)} \mathrm{d}\Omega = \int_{\partial V_c(t)} \boldsymbol{v} \cdot \mathbf{d}\boldsymbol{S}. \tag{1.40}$$

This last equation (in a form integrated in time) is often called the Geometrical Conservation Law (GCL) and will be described in more detail in Chapter 4.

Performing the same calculation for the variation of mass $\frac{\mathrm{d}}{\mathrm{d}t}\int_{V_c(t)} \rho\mathrm{d}\Omega$ in the cell c and using the first line of Eq. (1.1) we obtain:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V_c(t)} \rho\mathrm{d}\Omega + \int_{\partial V_c(t)} \rho(\boldsymbol{u} - \boldsymbol{v}) \cdot \mathbf{d}\boldsymbol{S} = 0. \tag{1.41}$$

Applying again the Leibniz integral rule (1.39) to establish the conservation of momentum in the domain $V_c(t)$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V_c(t)} \rho\boldsymbol{u}\mathrm{d}\Omega = \int_{V_c(t)} \frac{\partial \rho\boldsymbol{u}}{\partial t}\mathrm{d}\Omega + \int_{\partial V_c(t)} \rho\boldsymbol{u} \otimes \boldsymbol{v} \cdot \mathbf{d}\boldsymbol{S}. \tag{1.42}$$

We apply the Cauchy momentum equation to replace $\frac{\partial \rho\boldsymbol{u}}{\partial t}$ of Equation (1.42), which gives us:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V_c(t)} \rho\boldsymbol{u}\mathrm{d}\Omega + \int_{\partial V_c(t)} \boldsymbol{u} \otimes \rho(\boldsymbol{u} - \boldsymbol{v}) \cdot \mathbf{d}\boldsymbol{S} = \int_{\partial V_c(t)} \boldsymbol{\sigma} \cdot \mathbf{d}\boldsymbol{S} + \int_{V_c(t)} \boldsymbol{f}. \tag{1.43}$$

This thesis will not detail all the techniques developed for finite volume schemes (the reader may look at Eymard et al. (2006)), but we will detail in Chapter 4 the space scheme and time stepping of code_saturne for the moving mesh module, for which the legacy schemes are

co-located in space (see e.g. Archambeau et al. (2004)). A particular attention will be paid to the free-surface condition.

The software code_saturne also provides state-of-the-art schemes, based on the so-called *compatible discrete operators* (CDO) framework developed in Bonelle and Ern (2014). Theses schemes offer the ability to solve equations not only for variables cell-wise located (cell-based schemes) but also vertex-based or hybrid cell-face based schemes on polyhedral meshes. This will be favorably used for solving the vertex displacement of the moving mesh in Chapter 4. These schemes can somehow be extended to high order approximations using the hybrid high order (HHO) schemes (see Di Pietro and Ern (2015)).

An attempt to extend this scheme to second order in time is proposed in Appendix C.

1.3 Smoothed particle hydrodynamics with a pure Lagrangian particle approach

1.3.1 A short introduction to SPH

Smoothed particle hydrodynamics (SPH) is a mesh-less Lagrangian method used to simulate continuous media motion, especially for fluids. The Lagrangian point of view may be compared to the Eulerian one: the first one consists in studying a flow following fluid particles. Each physical quantity is then attached to its fluid element and transported with it with respect to the laws of physics. On the contrary, Eulerian approaches study physical quantities at a fixed position. Many methods such as finite difference, finite volume and finite element are originally based on that approach, which are also called mesh-based methods. Highly non-linear deformation such as a 3-D dam break cannot be handled with the ALE finite volume approach without re-meshing, because of the topology changing of the computational domain. Two phase models with the computation of the volume of fluid fraction are an alternative. They can be either based on an homogeneous approach, with only one velocity field, such as Volume Of Fluid (VOF, see Pilliod and Puckett (2004)) or with two coupled momentum equations as in Méchitoua et al. (2003). However, these methods require specific numerical treatments for the convective terms to avoid numerical diffusion of sharp interfaces.

SPH is one of the most used Lagrangian techniques. Due to its mesh-less formulation, SPH is easily able to simulate merging domains, and due to its Lagrangian formulation, it does not diffuse any interface. It was first introduced by Lucy (1977), and Gingold and Monaghan (1977). The use of SPH has since widely expanded in fluid dynamics and solid mechanics Monaghan (1999). The basic concept of SPH is that continuous media are represented by discrete particles with volume and mass. A kernel function is used to define particle's range of interaction, and the fields are approximated by integral interpolations which is nothing else but the convolution of the variable with the kernel function.

The SPH method was invented primarily to investigate and carry out astrophysical simulations. This method was particularly better suited than mesh-based Eulerian approach for: strong non-linear interactions between cosmic objects and high deformations, concentration of masses and empty areas. SPH was for instance used for the star collisions and galaxy formation predictions (see Patsis and Athanassoula (2000)).

Secondly, although widely developed and used through finite elements methods, solid mechanics found an interest to investigate the SPH method to compute cases where grid-based approaches face problems: we may cite for instance fracture and fragmentation of a solid. Indeed, when one studies fracture, one should be able to refine meshes where large distortion appears, to be accurate. Knowing where a discontinuity will occur before the simulation is a difficult problem to handle. On the contrary, as a grid-less method, SPH deals easily with this type of problem. We could refer for example to Das and Cleary (2008) who used the SPH method to successfully simulate the 3-D fracture and fragmentation in a thin plate.

Last, but not least, SPH has been adapted to perform fluid mechanics simulation in many cases, mainly for violent deformations with complex free-surface flow. Two major variants exist:

(*i*) a weakly compressible version where the pressure is obtained analytically by an equation of state. The Mach number, based on a numerical speed of sound, is supposed to be less than 10%, and

(*ii*) an incompressible version, where the pressure field is obtained by solving a Poisson equation to enforce incompressibility (see Xu (2009), Lind et al. (2012) or Leroy et al. (2016) for more details).

The approach (*i*) was introduced first, and has as drawbacks a relatively small time step due to the numerical speed of sound and a pressure field possibly impacted by small perturbations of the density field. The main advantages is to ensure naturally free-surface condition for pressure without imposing it. However, method (*ii*) allows larger time steps in the time schemes and gives a smoother pressure field, but requires a Poisson problem to be solved and to enforce pressure boundary condition at the free surface.

We have chosen here to investigate the weakly compressible version of SPH mainly because of its simplicity. However, the framework developed to deal with wall and open boundaries can be adapted to incompressible formulation, see Leroy et al. (2014, 2016).

Other variants of SPH were developed to better reproduce pressure fields. SPH, as code_saturne's finite volume scheme stores pressure and velocity in a co-located manner, and though spurious pressure field can be generated and not filtered. Among the variants, we can cite δ-SPH (see e.g. Sun et al., 2017) which add corrections terms which can be seen as a Rhie & Chow filter presented in §1.2.

Finally, it is worth noting that ALE versions of SPH have been proposed over the last decades to better perform transition where flows distorts rapidly and calm zones, but also to gain accuracy on the pressure field. They are often used with hyperbolic solvers, introduced by pioneer work of Inutsuka (1994), which allow correct treatment of shocks. Vila (1999) first introduced an ALE and mathematical analysis of SPH and was followed by many authors (see e.g. Marongiu et al., 2010) using Riemann solvers. Riemann solvers and fixed (therefore Eulerian) boundary particles will be used in Chapter 3.

1.3.2 Basic SPH formulations for compressible Newtonian fluid

Smoothed particle approximation

Before starting with SPH, let us define particle approximation following the work of Vila (1999). As before, let Ω be the domain, open, included in $\mathbb{R}^d$, d being the space dimension.

Space is partitioned by volumes $V_a(t)$ moving in time, indexed by a or b, of volume also denoted by $V_a(t)$. The centre of a volume (or particle) a is $\boldsymbol{x}_a(t)$. The set of (fluid) particles is denoted by $\mathcal{F}$.

Space Ω is deformed by a velocity field $\boldsymbol{v}(\boldsymbol{x}, t)$ supposed regular enough. Thus, particles evolve as:

$$\frac{\mathrm{d}\boldsymbol{x}_a}{\mathrm{d}t} = \boldsymbol{v}(\boldsymbol{x}_a, t) := \boldsymbol{v}_a(t). \tag{1.44}$$

Concerning the volume of a particle, it evolves as previously seen:

$$\frac{\mathrm{d}V_a}{\mathrm{d}t} = \int_{V_a} \mathrm{div}\,(\boldsymbol{v})\,\mathrm{d}\Omega = \int_{\partial V_a} \boldsymbol{v}\cdot\mathrm{d}\boldsymbol{S}. \tag{1.45}$$

Note that usually, the partition $\Omega = \bigcup_{a\in\mathcal{F}} V_a$ is not explicitly known.

We can now define the particle approximation distribution $\Pi(\psi)$ of a field ψ as:

$$\Pi(\psi)(\boldsymbol{x}) \coloneqq \sum_{a\in\mathcal{F}} V_a \psi(\boldsymbol{x}_a)\,\delta(\boldsymbol{x}-\boldsymbol{x}_a), \tag{1.46}$$

δ being the Dirac distribution. Distribution (1.46) is then smoothed / regularised using a kernel function, noted $w(\boldsymbol{x})$, assumed in the following to have a compact support $\Omega_{\boldsymbol{x}}$ (or in short Ω_a if at particle position $\boldsymbol{x}_a$) of smoothing length denoted h. The regularisation is noted $\Pi_w^{SPH}(\psi)$ and is the convolution between $\Pi(\psi)$ and w:

$$\Pi_w^{SPH}(\psi)(\boldsymbol{x}) \coloneqq (\Pi(\psi)\star w)(\boldsymbol{x}) = \sum_{a\in\mathcal{F}} V_a\psi(\boldsymbol{x}_a)\,w(\boldsymbol{x}-\boldsymbol{x}_a). \tag{1.47}$$

The finite volume method can also be written as an interpolation operator:

$$\Pi_w^{FV}(\psi)(\boldsymbol{x}) \coloneqq \sum_c V_c\psi_c \underbrace{\frac{1_{V_c}(\boldsymbol{x})}{V_c}}_{\coloneqq w_c^{FV}(\boldsymbol{x})}. \tag{1.48}$$

but cannot be written as a convolution between the particle projection distribution Π and a kernel which is radial basis function. In a way SPH is more isotropic than FV.

Some kernel examples are given in §1.3.2, and *a priori*, Π_w^{SPH} does not interpolate exactly constant fields, contrary to (1.48), which is known to be the unity partitioning property. As reported in Vila (1999), the error tends to zero as the particle distance Δr tends to zero **and** $\Delta r/h$ tends to zero, meaning that one needs to decrease the kernel support size and increase the number of particles in the domain but also in the kernel support[3].

Note that we could also define the continuous interpolation of ψ as the convolution with the kernel $\psi\star w$:

$$\langle\psi\rangle(\boldsymbol{x}) \coloneqq (\psi\star w)(\boldsymbol{x}) = \int_{\Omega_{\boldsymbol{x}}} \psi(\boldsymbol{x}')\,w(\boldsymbol{x}-\boldsymbol{x}')\,\mathrm{d}x', \tag{1.49}$$

for which $\Pi(\psi)\star w = \Pi_w^{SPH}(\psi)$ is a discrete approximation. In other words, the smoothed particle approximation can either be seen as a particle approximation then smoothed with a kernel, or as a continuous smoothed approximation then discretised over a set of particles (the latter approach being more convenient to estimate the errors).

When studying bounded domain, the convolution with kernel w does not preserve constant field functions ψ. To circumvent this issue, an *analytical* factor $\gamma(\boldsymbol{x})$ of renormalisation is introduced next to the boundary $\partial\Omega$ (see Benz (1990) and Herant (1994)):

$$\gamma(\boldsymbol{x}) \coloneqq (1\star w)(\boldsymbol{x}) = \int_{\Omega\cap\Omega_{\boldsymbol{x}}} w(\boldsymbol{x}'-\boldsymbol{x})\,\mathrm{d}x'. \tag{1.50}$$

The introduction of this factor leads to the following smoothed particle approximation definition:

$$\Pi_w^{\gamma}(\psi)(\boldsymbol{x}) \coloneqq \left(\frac{\Pi_w(\psi)}{1\star w}\right)(\boldsymbol{x}) = \frac{1}{\gamma(\boldsymbol{x})}\sum_{a\in\mathcal{F}} V_a\psi(\boldsymbol{x}_a)\,w(\boldsymbol{x}-\boldsymbol{x}_a). \tag{1.51}$$

The use of *analytical* renormalisation will make boundary terms appear as detailed in Chapters 2 and 3.

Now equipped of these approximation interpolators, we will define the standard SPH operators starting from the density variation.

[3] Renormalisation techniques can improve the accuracy of the smoothed particle approximation as proven in Vila (1999), and is quite popular in the SPH community (see for instance Bonet and Lok (1999)) but is out of the scope of this thesis because it may break some conservation properties. Nevertheless, boundary terms proposed in this thesis can be adapted to renormalisation, see Mayrhofer et al. (2014).

Space discretised governing equations

The aim of this Section is to recover SPH operators. The (weakly) compressible Newtonian fluid is modelled by a set of particles denoted by the subscripts $(.)_a$ and $(.)_b$ in a domain Ω. Each particle $a \in \mathcal{F}$ possesses information such as its mass M_a, its position $\boldsymbol{x}_a$, its velocity $\boldsymbol{v}_a$ which is here the Lagrangian derivative of the position[4], its density ρ_a, and its pressure p_a. Most of the time, we use a pure Lagrangian framework and particle velocity $\boldsymbol{v}_a$ coincides with fluid velocity at point $\boldsymbol{x}_a$, denoted by $\boldsymbol{u}_a$. This will not be the case for motionless open boundaries for instance, where $\boldsymbol{v}$ is zero while $\boldsymbol{u}$ is not.

Volume, mass and density of particles are linked by the relation: $V_a = \frac{M_a}{\rho_a}$. We assume in the sequel that fluid particle mass is constant[5] and only one phase is considered with no mass fluxes between particles (see e.g. Fonty et al., 2019, for extension to multiphase flow).

We often denote by the subscripts $(.)_{ab}$ the difference of a quantity between the positions a and b. For instance $\boldsymbol{u}_{ab} := \boldsymbol{u}_a - \boldsymbol{u}_b$ and $\boldsymbol{x}_{ab} := \boldsymbol{x}_a - \boldsymbol{x}_b$. Three exceptions are made with the following notations $r_{ab} := |\boldsymbol{x}_{ab}|$, $w_{ab} := w(\boldsymbol{x}_{ab})$ and $\boldsymbol{\nabla} w_{ab} := \boldsymbol{\nabla} w(\boldsymbol{x}_{ab}) = \frac{\partial w}{\partial x}(\boldsymbol{x}_{ab})\boldsymbol{e}_x + \frac{\partial w}{\partial y}(\boldsymbol{x}_{ab})\boldsymbol{e}_y + \frac{\partial w}{\partial z}(\boldsymbol{x}_{ab})\boldsymbol{e}_z$, $(\boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z)$ being the basis vector triad of the Cartesian coordinate system in 3-D.

Mass equation, everything comes from density variation definition:

With these notations, a commonly used form of the continuity equation is (see e.g. Monaghan (1992)):

$$\frac{\mathrm{d}\rho_a}{\mathrm{d}t} = \sum_{b \in \mathcal{F}} M_b \boldsymbol{\nabla} w_{ab} \cdot \boldsymbol{u}_{ab}. \tag{1.52}$$

As mentioned in Vila (1999), as soon as fluid and particle velocities coincide, i.e. $\boldsymbol{u}_a = \boldsymbol{v}_a$ and particle masses are constant, Equation (1.52) is strictly equivalent to:

$$\frac{\mathrm{d}\rho_a}{\mathrm{d}t} = \frac{\mathrm{d}\Pi_w(\rho)}{\mathrm{d}t}(\boldsymbol{x}_a) = \frac{\mathrm{d}\left(\sum_{b \in \mathcal{F}} M_b w_{ab}\right)}{\mathrm{d}t}. \tag{1.53}$$

Therefore, (1.53) is taken as the definition of the density variation, and allows us to compute density over time for all particles knowing their displacements and supplemented with initial conditions[6].

Equation (1.52) can be used to define the discrete divergence operator:

$$\mathrm{Div}_a^- \{\boldsymbol{u}_b\} := -\frac{1}{\rho_a} \sum_{b \in \mathcal{F}} M_b \boldsymbol{u}_{ab} \cdot \boldsymbol{\nabla} w_{ab}. \tag{1.54}$$

which gives:

$$\frac{\mathrm{d}\rho_a}{\mathrm{d}t} = -\rho_a \mathrm{Div}_a^- \{\boldsymbol{u}_b\}. \tag{1.55}$$

[4] Note that other authors consider the particle velocity to be arbitrary Lagrangian-Eulerian, see for instance Marongiu et al. (2010); Neuhauser et al. (2014); Oger et al. (2016).

[5] This assumption is not mandatory, some authors work using the ALE framework (see e.g. Marongiu et al., 2010; Oger et al., 2016). This will not be assumed for open-boundary particles in Chapter 3.

[6] Note that some authors (see e.g. Inutsuka, 2002) define the density to be $\rho_a \simeq \Pi_w(\rho)(\boldsymbol{x}_a) = \sum_{b \in \mathcal{F}} M_b w_{ab}$. This formula requires to initialise particles' masses correctly to get the initial condition on the density and may disturb the free-surface. The choice made here is less restrictive because we only assume that the **variation** of ρ_a is approximated by the **variation** of $\Pi_w(\rho)$ when particles move with their (fluid) velocity.

Exponent $(.)^-$ stands for the minus in Equation (1.54) and therefore this operator is not conservative in the finite volume sense. To understand this, we can write the divergence operator as in the finite volume framework, defining the *outward* surface vector $\boldsymbol{S}_{a>b} := V_a V_b \boldsymbol{\nabla} w_{ab} = -\boldsymbol{S}_{b>a}$:

$$V_a \mathrm{Div}_a^- \{\boldsymbol{u}_b\} = \sum_{b \in \mathcal{F}} \frac{\rho_b}{\rho_a} \boldsymbol{u}_b \cdot \boldsymbol{S}_{a>b} - \frac{\boldsymbol{u}_a}{\rho_a} \cdot \sum_{b \in \mathcal{F}} \rho_b \boldsymbol{S}_{a>b}. \tag{1.56}$$

The second summation in (1.56) cancels out for constant density flow in the finite volume method because the integral on a closed volume V_c of $\boldsymbol{\nabla} 1$ gives that $\sum_{f_{c|\overline{c}}} \boldsymbol{S}_{c>f} = \boldsymbol{0}$. This property is only approximately true in SPH without any other treatment which means that the operator $\mathrm{Div}_a^- \{\boldsymbol{u}_b\}$ is not conservative.

Other choice in (1.53) would bring other discrete operator definition. Changing this definition can be of interest for flows with highly variable density (multi-fluids, mixture models, see e.g. Fonty et al. (2019) for a recent review).

Note that, using expression (1.53), exact time integration on $[t^n,\ t^{n+1}]$ is trivial:

$$\rho_a^{n+1} - \rho_a^n = \int_{t^n}^{t^{n+1}} \rho_a \mathrm{Div}_a^- \{\boldsymbol{u}_b\} \, \mathrm{d}t = \sum_{b \in \mathcal{F}} M_b \left(w_{ab}^{n+1} - w_{ab}^n \right). \tag{1.57}$$

This is an interesting property of reversibility: if (all) particles move back to their initial position, then densities also take back their initial values.

Please bear in mind that boundary terms are missing in these operators, and will be added in Chapter 2 for walls and in Chapter 3 for open boundaries.

Momentum equation, pressure gradient derived from a Lagrangian:

The inviscid momentum equation is usually written as follows:

$$\frac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t} = -\sum_{b \in \mathcal{F}} M_b \left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2} \right) \boldsymbol{\nabla} w_{ab} + \boldsymbol{g}, \tag{1.58}$$

where $\boldsymbol{g}$ is the gravity field.

This formula can be deduced from the choice made on density variation (1.53), and from a classical derivation of momentum from a Lagrangian *for an insulated system*:

$$\frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial L}{\partial \boldsymbol{u}_a} \right) - \left(\frac{\partial L}{\partial \boldsymbol{x}_a} \right) = 0, \tag{1.59}$$

where the Lagrangian is defined by:

$$L(\boldsymbol{x}_a,\ \boldsymbol{u}_a) = E_{ke}(\boldsymbol{u}_a) - E_{int}(\boldsymbol{x}_a) - E_{ext}(\boldsymbol{x}_a). \tag{1.60}$$

Note that here we still assume that fluid and particle velocities coincide, i.e. $\boldsymbol{u}_a = \boldsymbol{v}_a$. The energies E_{ke}, E_{int} and E_{ext} are respectively the kinetic energy, the internal energy and the external energy. The kinetic energy is given by (1.61a). If the external forces are due to the gravitational field $\boldsymbol{g}$, the external energy is given by (1.61b) and for an adiabatic reversible process, the internal energy can be written as (1.61c):

$$E_{ke} := \frac{1}{2} \sum_{b \in \mathcal{F}} M_b \boldsymbol{u}_b^2, \tag{1.61a}$$

$$E_{ext} := -\sum_{b \in \mathcal{F}} M_b \boldsymbol{x}_b \cdot \mathbf{g}, \tag{1.61b}$$

$$E_{int} := \sum_{b \in \mathcal{F}} M_b \, e_{int}(\rho_b), \tag{1.61c}$$

and the energy accumulated per unit of mass e_{int} varies with the density ρ according to (1.62).

$$p = \rho^2 \frac{\mathrm{d}e_{int}}{\mathrm{d}\rho}. \tag{1.62}$$

Note that (1.53) means that density is a continuous function of the position $\boldsymbol{x}$ and gives us a way to compute the value of $\rho_a := \rho(\boldsymbol{x}_a)$ if we know the positions of all particles.

Now we are able to derive the momentum equation from the Euler-Lagrange equation (1.59) and obtain for each particle a:

$$M_a \frac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t} = \mathbf{F}_a^{int} + M_a \boldsymbol{g}, \tag{1.63}$$

where the internal force due to the pressure, $\mathbf{F}_a^{int}$, is defined by:

$$\begin{aligned} \mathbf{F}_a^{int} &= -\frac{\partial E_{int}}{\partial \boldsymbol{x}_a} \\ &= -\frac{\partial}{\partial \boldsymbol{x}_a}\left[\sum_{b\in\mathcal{F}} M_b\, e_{int}(\rho_b)\right] \\ &= -\sum_{b\in\mathcal{F}} M_b \frac{\mathrm{d}e_{int}}{\mathrm{d}\rho_b}\frac{\mathrm{d}\rho_b}{\mathrm{d}\boldsymbol{x}_a} \\ &= -\sum_{b\in\mathcal{F}} M_b \frac{p_b}{\rho_b^2}\frac{\mathrm{d}\rho_b}{\mathrm{d}\boldsymbol{x}_a}, \end{aligned} \tag{1.64}$$

but we can derive the density using its definition (1.53):

$$\frac{\mathrm{d}\rho_b}{\mathrm{d}\boldsymbol{x}_a} = \sum_{c\in\mathcal{F}} M_c \frac{\mathrm{d}w_{bc}}{\mathrm{d}\boldsymbol{x}_a} = M_a \boldsymbol{\nabla}_a w_{ab} + \delta_{ab} \sum_{c\in\mathcal{F}} M_c\, \boldsymbol{\nabla}_a w_{ac},$$

and (1.64) becomes

$$\begin{aligned} \mathbf{F}_a^{int} &= -\sum_{b\in\mathcal{F}} M_b \frac{p_b}{\rho_b^2}\frac{\mathrm{d}\rho_b}{\mathrm{d}\boldsymbol{x}_a} \\ &= -\sum_{b\in\mathcal{F}} M_b \frac{p_b}{\rho_b^2}\left[M_a \boldsymbol{\nabla}_a w_{ab} + \delta_{ab}\left(\sum_{c\in\mathcal{F}} M_c\, \boldsymbol{\nabla}_a w_{ac}\right)\right] \\ &= -\sum_{b\in\mathcal{F}} M_a M_b \left(\frac{p_b}{\rho_b^2} + \frac{p_a}{\rho_a^2}\right) \boldsymbol{\nabla}_a w_{ab}. \end{aligned}$$

This derivation gives the natural definition for the discrete pressure gradient:

$$\mathbf{Grad}_a^+ \{p_b\} := \rho_a \sum_{b\in\mathcal{F}} M_b \left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2}\right) \boldsymbol{\nabla} w_{ab}. \tag{1.65}$$

The exponent $(.)^+$ means that the pressure gradient fulfills the action/reaction principle (pressure force exerted from particle a to particle b is minus the one exerted from particle b to particle a). In other words it is a conservative form. Indeed $V_a \mathbf{Grad}_a^+ \{p_b\}$ rewrites:

$$V_a \mathbf{Grad}_a^+ \{p_b\} = \sum_{b\in\mathcal{F}} p_{a|b} \boldsymbol{S}_{a>b}, \tag{1.66}$$

with $p_{a|b} := \rho_a \rho_b \left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2} \right) = p_{b|a}$.

Thus the discrete momentum equation (1.58) can be rewritten as follows:

$$\frac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t} = -\frac{1}{\rho_a} \mathbf{Grad}_a^+ \{p_b\} + \boldsymbol{g}. \tag{1.67}$$

The equation of state is considered barotropic:

$$p_a = \frac{\rho_0 c_0^2}{\gamma} \left[\left(\frac{\rho_a}{\rho_0} \right)^\gamma - 1 \right], \tag{1.68}$$

where ρ_0 is the reference density of the fluid, c_0 is the speed of sound and $\gamma = 7$ for water.

One can check that these equations satisfy the following conservation equations of linear and angular momentum:

$$\begin{aligned} \sum_{a \in \mathcal{F}} \boldsymbol{F}_a^{int} &= \mathbf{0}, \\ \sum_{a \in \mathcal{F}} \boldsymbol{x}_a \times \boldsymbol{F}_a^{int} &= \mathbf{0}. \end{aligned} \tag{1.69}$$

Properties (1.69) hold if the kernel w verifies properties of isotropy (which implies symmetry, i.e. $w_{ab} = w_{ba}$ and implies that $\boldsymbol{\nabla} w_{ab} = -\boldsymbol{\nabla} w_{ba}$ and $\boldsymbol{\nabla} w_{ab}$ is collinear to $\boldsymbol{x}_{ab}$).

Discrete operator definitions, gradient and divergence skew-adjointness:

We recall here the definition of the operators gradient ($\mathbf{Grad}_a^+ \{A_b\}$) and divergence ($\mathrm{Div}_a^- \{\boldsymbol{A}_b\}$) of discrete scalar or vector field ($\{A_b\}$ or $\{\boldsymbol{A}_b\}$):

$$\begin{aligned} \mathbf{Grad}_a^+ \{A_b\} &:= \rho_a \sum_{b \in \mathcal{F}} M_b \left(\frac{A_a}{\rho_a^2} + \frac{A_b}{\rho_b^2} \right) \boldsymbol{\nabla} w_{ab}, \\ \mathrm{Div}_a^- \{\boldsymbol{A}_b\} &:= -\frac{1}{\rho_a} \sum_{b \in \mathcal{F}} M_b \boldsymbol{A}_{ab} \cdot \boldsymbol{\nabla} w_{ab}, \end{aligned} \tag{1.70}$$

where $\boldsymbol{A}_{ab} := \boldsymbol{A}_a - \boldsymbol{A}_b$.

The operators $\mathbf{Grad}_a^+ \{.\}$ and $\mathrm{Div}_a^- \{.\}$ are said to be skew-adjoint (see Appendix D.3 for details), in other words, for two arbitrary fields of scalar $\{A_b\}$ and of vector $\{\boldsymbol{B}_b\}$, we have the property:

$$\left\langle \mathbf{Grad}_b^+ \{A_a\}, \boldsymbol{B}_b \right\rangle = -\left(A_b, \mathrm{Div}_b^- \{\boldsymbol{B}_a\} \right), \tag{1.71}$$

where the inner products are defined by a sum over the domain:

$$\begin{aligned} \langle \boldsymbol{A}_b, \boldsymbol{B}_b \rangle &:= \sum_{b \in \mathcal{F}} V_b \boldsymbol{A}_b \cdot \boldsymbol{B}_b, \\ (A_b, B_b) &:= \sum_{b \in \mathcal{F}} V_b A_b B_b. \end{aligned} \tag{1.72}$$

This property, which also holds for the continuous operators, could be used in simple variational calculus as follows, which is important for the conservation of the energy for instance.

Indeed, if we compute the work of pressure force associated with the virtual displacement field $\{\mathrm{d}\boldsymbol{x}_b\}$, we obtain:

$$\sum_{b\in\mathcal{F}} \boldsymbol{F}_b^{int}\cdot\mathrm{d}\boldsymbol{x}_b = \left\langle \frac{\rho_b}{M_b}\boldsymbol{F}_b^{int},\, \mathrm{d}\boldsymbol{x}_b \right\rangle \tag{1.73a}$$

$$= -\sum_{a,b\in\mathcal{F}} M_a \frac{p_a}{\rho_a^2}\frac{\mathrm{d}\rho_a}{\mathrm{d}\boldsymbol{x}_b}\cdot\mathrm{d}\boldsymbol{x}_b \tag{1.73b}$$

$$= -\sum_{a\in\mathcal{F}} V_a \frac{p_a}{\rho_a}\mathrm{d}\rho_a \tag{1.73c}$$

$$= \sum_{a\in\mathcal{F}} V_a p_a \mathrm{Div}_a^-\left\{\mathrm{d}\boldsymbol{x}_c\right\} \tag{1.73d}$$

$$= \left(p_b,\, \mathrm{Div}_b^-\left\{\mathrm{d}\boldsymbol{x}_c\right\}\right) \tag{1.73e}$$

$$= -\left\langle \mathbf{Grad}_b^+\left\{p_a\right\},\, \mathrm{d}\boldsymbol{x}_b\right\rangle . \tag{1.73f}$$

In the previous calculus, we have used the definition of the inner products (1.72) in (1.73a), then the definition of the internal forces in (1.73b), then the definition of the differential in (1.73c), (1.73d) is nothing but the continuity equation, and finally the adjoint property (1.71) is used in (1.73f).

The previous calculus proves that for any virtual displacement field $\{\mathrm{d}\boldsymbol{x}_b\}$, the inner products $\left\langle \frac{\rho_b}{M_b}\boldsymbol{F}_b^{int},\, \mathrm{d}\boldsymbol{x}_b\right\rangle$ and $-\left\langle \mathbf{Grad}_b^+\left\{p_a\right\},\, \mathrm{d}\boldsymbol{x}_b,\right\rangle$ are equal, which implies that for each particle b:

$$\frac{1}{M_b}\boldsymbol{F}_b^{int} = -\frac{1}{\rho_b}\mathbf{Grad}_b^+\left\{p_a\right\} . \tag{1.74}$$

That means that the continuity equation and the momentum equation are variationally consistent.

It is possible to define variants of these two operators conserving this property of skew-adjointness. For more details see Violeau (2012).

Viscous forces

The viscous term is of second-order and hence, necessitates treatment of second order spatial derivatives of the kernel. Usually in SPH, we avoid the use of such kernel derivatives, because they are sensitive to the particle disorder and therefore generate numerical instabilities (see Issa et al. (2005)). For this reason, the common used formulae for the discrete Laplacian are obtained by decomposing into a divergence of a gradient. Then, the divergence is approximated by a kernel derivative whereas the gradient is approximated by a finite differences formula.

The main two models for computing the viscous term $\nabla\cdot(\mu\boldsymbol{\nabla}\boldsymbol{u})$ are:

$$\boldsymbol{\nabla}\cdot(\mu\boldsymbol{\nabla}\boldsymbol{u})_a = \sum_{b\in\mathcal{F}} M_b \frac{\mu_a+\mu_b}{\rho_a\rho_b}\frac{\boldsymbol{u}_{ab}}{r_{ab}^2}\boldsymbol{x}_{ab}\cdot\boldsymbol{\nabla}w_{ab}, \tag{1.75a}$$

$$\boldsymbol{\nabla}\cdot(\mu\boldsymbol{\nabla}\boldsymbol{u})_a = 2(n+2)\sum_{b\in\mathcal{F}} M_b \frac{\mu_a+\mu_b}{\rho_a\rho_b}\frac{\boldsymbol{u}_{ab}\cdot\boldsymbol{x}_{ab}}{r_{ab}^2}\boldsymbol{\nabla}w_{ab}, \tag{1.75b}$$

where we recall that the dynamic viscosity μ is linked to the kinematic viscosity by $\mu{:=}\nu\rho$, and n is the space dimension. The first model (1.75a) is due to Morris et al. (1997) and the second one (1.75b) to Monaghan and Gingold (1983).

Basically, Monaghan's model is obtained by assuming that $\boldsymbol{\nabla}\cdot\left(\mu\boldsymbol{\nabla}\boldsymbol{u}^T\right)=0$ which is true for laminar of constant dynamic viscosity incompressible flows. A general formula could be derived

(see Espanol and Revenga (2003) for algebraic details):

$$\boldsymbol{\nabla}\cdot\left[\mu\left(\boldsymbol{\nabla}\boldsymbol{u}+\boldsymbol{\nabla}\boldsymbol{u}^{T}\right)\right]_{a}=\sum_{b\in\mathcal{F}}M_{b}\frac{\frac{\mu_{a}+\mu_{b}}{2}}{\rho_{a}\rho_{b}}\left[(n+2)\frac{\boldsymbol{u}_{ab}\cdot\boldsymbol{x}_{ab}}{r_{ab}^{2}}\boldsymbol{\nabla}w_{ab}+\frac{\boldsymbol{\nabla}w_{ab}\cdot\boldsymbol{x}_{ab}}{r_{ab}^{2}}\boldsymbol{u}_{ab}\right]. \quad (1.76)$$

Laplacian operator: divergence operator of a gradient field

The viscous force of Morris et al. (1997) can be derived as for the Laplacian operator in the finite volume method:

$$\mathbf{Lap}_{a}^{+}\left(\{\mu_{b}\},\{\boldsymbol{u}_{b}\}\right):=\boldsymbol{\nabla}\cdot\left(\mu\boldsymbol{\nabla}\,\boldsymbol{u}\right)_{a}=\mathbf{Div}_{a}^{+}\left\{\left(\mu\boldsymbol{\nabla}\,\boldsymbol{u}\right)\right\}_{b}, \quad (1.77)$$

where the conservative divergence operator is defined for an arbitrary gradient field $\{\boldsymbol{\nabla}\boldsymbol{A}_{b}\}$ and a scalar field $\{\boldsymbol{B}_{b}\}$ by:

$$\mathbf{Div}_{a}^{+}\left\{B_{b}\boldsymbol{\nabla}\boldsymbol{A}_{b}\right\}=\sum_{b\in\mathcal{F}}V_{b}\left(B_{a}\boldsymbol{\nabla}\boldsymbol{A}_{a}+B_{b}\boldsymbol{\nabla}\boldsymbol{A}_{b}\right)\cdot\boldsymbol{\nabla}w_{ab}. \quad (1.78)$$

This operator is indeed conservative as it can be rewritten:

$$V_{a}\mathbf{Div}_{a}^{+}\left\{B_{b}\boldsymbol{\nabla}\boldsymbol{A}_{b}\right\}=\sum_{b\in\mathcal{F}}\left(B_{a}\boldsymbol{\nabla}\boldsymbol{A}_{a}+B_{b}\boldsymbol{\nabla}\boldsymbol{A}_{b}\right)\cdot\boldsymbol{S}_{a>b}. \quad (1.79)$$

If then we use a Two Point Flux Approximation:

$$B_{a}\boldsymbol{\nabla}\boldsymbol{A}_{a}\cdot\boldsymbol{e}_{ab}\simeq B_{b}\boldsymbol{\nabla}\boldsymbol{A}_{b}\cdot\boldsymbol{e}_{ab}\simeq\langle B\rangle_{ab}\frac{\boldsymbol{A}_{ab}}{r_{ab}}, \quad (1.80)$$

where $\langle B\rangle_{a|b}$ is an interpolated value between B_a and B_b, taken here as the arithmetic mean $\langle B\rangle_{a|b}=\frac{B_a+B_b}{2}$. Note that continuity of the fluxes would give a geometric mean $\langle B\rangle_{a|b}=2\frac{B_aB_b}{B_a+B_b}$ as used in Ghaïtanellis et al. (2018) for sediment transport and Fonty et al. (2019) for a mixture model for two-phase flows.

Kernels used

Generally speaking, kernels used in the SPH formulation depend on the smoothing length-scale h, on the dimension of the working space ($n=2$ or $n=3$) and on the norm of the position vector $r=|\boldsymbol{x}|$ and can be written as:

$$w\left(r\right)=\frac{\alpha_{f,n}}{h^{n}}f(q), \quad (1.81)$$

where q is given by

$$q:=\frac{r}{h}, \quad (1.82)$$

and $\alpha_{f,n}$ is a normalisation factor to ensure that the integral of the kernel on its support is 1, i.e.:

$$\int_{\Omega_{0}}w\left(r\right)\mathrm{d}\boldsymbol{x}=1. \quad (1.83)$$

The kernels used in the present work are:

1. The spline based kernel of order 4 defined by:

$$f(q) = \begin{cases} \left(\frac{5}{2}-q\right)^4 - 5\left(\frac{3}{2}-q\right)^4 + 10\left(\frac{1}{2}-q\right)^4 & \text{when} \quad 0 \le q < \frac{1}{2}, \\ \left(\frac{5}{2}-q\right)^4 - 5\left(\frac{3}{2}-q\right)^4 & \text{when} \quad \frac{1}{2} \le q < \frac{3}{2}, \\ \left(\frac{5}{2}-q\right)^4 & \text{when} \quad \frac{3}{2} \le q < \frac{5}{2}, \\ 0 & \text{when} \quad \frac{5}{2} \le q, \end{cases} \tag{1.84}$$

with

$$\begin{aligned} \alpha_{f,2} &= \frac{96}{1199\pi}, \\ \alpha_{f,3} &= \frac{1}{20\pi}. \end{aligned} \tag{1.85}$$

2. The quintic kernel (Wendland (1995)) defined by:

$$f(q) = \begin{cases} \left(1-\frac{q}{2}\right)^4 (2q+1) & \text{when} \quad 0 \le q < 2, \\ 0 & \text{when} \quad 2 \le q, \end{cases} \tag{1.86}$$

with

$$\begin{aligned} \alpha_{f,2} &= \frac{7}{4\pi}, \\ \alpha_{f,3} &= \frac{21}{16\pi}. \end{aligned} \tag{1.87}$$

The Fourier transform of (1.86) is positive decreasing (see Robinson (2009); Violeau and Leroy (2014) for the calculation). This property is useful regarding to the stability of the system of particles, for more information see Dehnen and Aly (2012).

1.3.3 Methods for wall modelling

Many ways of implementing solid walls in SPH have been developed over the past decades, and boundary conditions topic is the second *Grand Challenge* of the SPHERIC community[7]. Among the most popular and commonly used, we can cite:

(*i*) Lennard-Jones repulsive forces proposed by Monaghan (1994),

(*ii*) fictitious particles (or also ghost particles) which consist in filling the empty area of the kernel support behind a wall with particles and giving them well-chosen physical quantities (such as pressure and velocity) to enforce no slip or free slip condition for example,

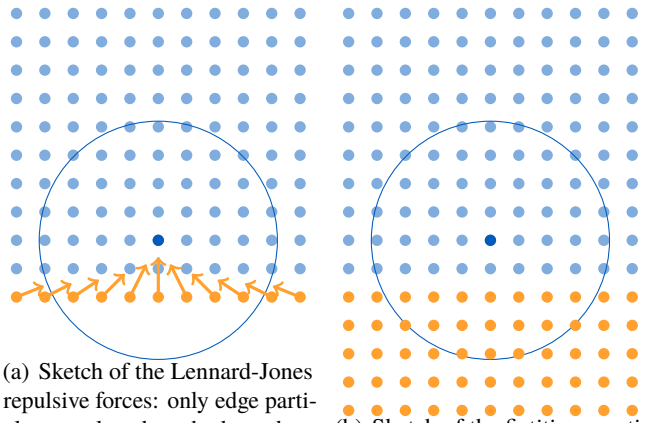
[7] See `www.spheric-sph.org`.

(*iii*) semi-analytical (or boundary integrals) boundary conditions based on a variational formulation introduced by Kulasegaram et al. (2004) where a wall renormalisation of the equations is made with respect to the missing area of the kernel support. We will describe a variant of this method in Chapter 2. Since the publication of this work in Ferrand et al. (2013), boundary integrals methods has become more and more popular, and a recent review of this is available in Vacondio et al. (2021). Chiron et al. (2019) recently propose an efficient 3-D version of the boundary integral method.

All of these include many variants and have advantages and drawbacks. The Lennard-Jones potential force (*i*) originally describes the interaction between pairs of atoms and is used to model the repulsion between a fluid particle and a boundary particle (see Figure 1.7(a)). This method is then easy to implement, even for complex geometries and computationally cheap. However, it leads to spurious behaviour: for instance it is impossible to maintain particles fixed along a vertical wall in the presence of gravity.

The fictitious particles (*ii*) prevent such a non-physical behaviour (see Figure 1.7(b)). However, the positioning of ghost particles in complex geometries can be really awkward, particularly in 3-D, even if recent a recent proposition Fourtakas et al. (2019) is applicable in 3-D and gives smooth field next to the boundary are partially due to the use of the δ-SPH techniques introduced in Antuono et al. (2010). Moreover, the computational effort required is not negligible, given that we increase the number of particles to take into account in the discrete summations (1.52) and (1.58).

Finally, the semi-analytical approach (*iii*) is attractive thanks to its variational formulation. That means that some physical quantities such as momentum will be automatically conserved. Unfortunately, the original attempt did not present a clear and simple way to compute renormalisation terms. Furthermore, the formulation proposed in Kulasegaram et al. (2004) was not able to reproduce hydrostatic pressure field or to take into account the shear stress along a wall. That is why Kulasegaram's original methodology needs to be modified, particularly if we want to focus on turbulence, heat transfer, etc.



(a) Sketch of the Lennard-Jones repulsive forces: only edge particles are placed on the boundary $\partial\Omega$.

(b) Sketch of the fictitious particles (in orange) used to complete the kernel area support for fluid particles on the "solid" side of the wall.

Figure 1.7: Sketch of the two easiest methods to model a wall: repulsive forces and fictitious particles.

1.3.4 Methods for open boundaries

In practical use, it is often mandatory to restrict the computational domain to the zone of interest. This implies enforcing boundary conditions where the fluid may enter or leave the domain, depending on the sign of the dot product between the fluid velocity field and the (inward) normal to the boundary.

This is natural when using the finite volume approach because all faces are seen as interfaces, no distinction made for inner or boundary face. The "only" difficulty is to specify the "correct" or "physical" value of the flow outside the domain (imposed entering mass flux, imposed pressure profile with homogeneous Neumann on the velocity field, external state of the far field solution, etc.).

As for SPH, that is less natural if the method is not seen as a flux method. A valuable description of the different approaches proposed for SPH can be found in Fonty (2019) and can be split into two families:

1. Buffer zone approach, where boundary is extruded and fluid quantities are imposed on a volume (see e.g. Crespo et al., 2007; Marrone et al., 2011; Vacondio et al., 2011; Adami et al., 2012; Bouscasse et al., 2013; Braun et al., 2015; Werdelmann et al., 2021). The main advantage of doing so is that free fluid particles next to the open-boundary zones see discrete contributions (for the pressure gradient in the momentum equation, and for the density variation in the continuity equation) from their surrounding neighbours as in the inner part of the domain. Therefore they are designed to limit spurious shocks when particles enter / leave the studied area. However, one has to specify quantities not only in a surface (as boundary conditions should do) but on a volume, which can be hazardous for complex geometries or to impose complex velocity and / or pressure profiles.

2. Flux methods as SPH-ALE (Marongiu et al. (2010)) where classical Riemann problems can be favorably used to impose wanted values. This method still requires to create and / or delete particles, and, when creating a fluid particle, its neighbours see a Dirac time-variation of the summation of fluxes. Renormalisation may become mandatory.

In Chapter 3, we present a methodology for weakly compressible SPH based on fluxes on the boundary, discretised in surface elements (segments in 2-D, triangles or quadrangles in 3-D) as introduced in Chapter 2. As for the finite volume method, integration by parts gives boundary values (and boundary terms) to be imposed at the open-boundary. The vertex particles, located at the ends of the surface element are used to inject mass in the domain: their mass is made variable in time, and they have exactly the mass of a particle subtracted when it is released. Therefore discrete operators do not suffer from Dirac time-variation. All classical methods based on the Riemann problem can be used for the flux part. Finally, a special care is given to the time integration of the mass equation.

Since the publication of this work in Ferrand et al. (2017), some valuable work has been performed on open (transparent) boundary conditions using both buffer zones removing spurious chocs using δ-SPH (see e.g., Bouscasse et al., 2017) and boundary integrals methods has become more and more popular, and a recent review of this is available in Vacondio et al. (2021).

1.3.5 Computation of the renormalisation factor

Computation in 2-D of renormalisation factor γ and its gradient $\boldsymbol{\nabla}\gamma$ are proposed in Chapter 2. Note that computational effort can be saved integrating analytically only $\boldsymbol{\nabla}\gamma$, and using a time evolution equation for γ itself.

Extension to 3-D was performed analytically with a decomposition of the boundary integral in Mayrhofer et al. (2015). However, this algorithm is tedious and sensitive to truncation errors.

A simpler algorithm has been published in Violeau et al. (2014), still requiring attention when dealing with truncation errors.

Similar types of analytical integrations have been proposed by Chiron et al. (2019) both in 2-D and 3-D where complex boundaries are handled with a triangulation. A recent and efficient proposition is made by Kostorz and Esmail-Yakas (2021a) for 3-D and in Kostorz and Esmail-Yakas (2021b) for spherical solids.

Bibliography

Adami, S., Hu, X.Y., Adams, N.A., 2012. A generalized wall boundary condition for smoothed particle hydrodynamics. Journal of Computational Physics 231, 7057–7075.

Antuono, M., Colagrossi, A., Marrone, S., Molteni, D., 2010. Free-surface flows solved by means of sph schemes with numerical diffusive terms. Computer Physics Communications 181, 532–549.

Archambeau, F., Méchitoua, N., Sakiz, M., 2004. Code_saturne: a finite volume method for the computation of turbulent incompressible flows - industrial applications. International Journal on Finite Volumes 1, 1–62.

Benz, W., 1990. Smoothed particle hydrodynamics: A review. In numerical modeling of nonlinear stellar pulsation: Problems and prospects. Kluwer Academic Dordrecht .

Bonelle, J., Ern, A., 2014. Analysis of Compatible Discrete Operator schemes for elliptic problems on polyhedral meshes. ESAIM Mathematical Modelling and Numerical Analysis 48, 553–581.

Bonet, J., Lok, T.S.L., 1999. Variational and momentum preservation aspects of Smooth Particle Hydrodynamic formulations. Computer Methods in applied mechanics and engineering 180, 97–115.

Bouscasse, B., Colagrossi, A., Marrone, S., Souto-Iglesias, A., 2017. Sph modelling of viscous flow past a circular cylinder interacting with a free surface. Computers & Fluids 146, 190–212.

Bouscasse, B., Marrone, S., Colagrossi, A., Di Mascio, A., 2013. Multi-purpose interfaces for coupling SPH with other solvers, in: 8th SPHERIC International Workshop.

Braun, S., Wieth, L., Koch, R., Bauer, H.J., 2015. A Framework for Permeable Boundary Conditions in SPH: Inlet, Outlet, Periodicity, in: 10th SPHERIC International Workshop.

Chiron, L., de Leffe, M., Oger, G., Touzé, D.L., 2019. Fast and accurate SPH modelling of 3D complex wall boundaries in viscous and non viscous flows. Computer Physics Communications 234, 93–111. URL: http://www.sciencedirect.com/science/article/pii/S0010465518302832, doi:https://doi.org/10.1016/j.cpc.2018.08.001.

Chorin, A.J., 1968. Numerical solution of the Navier-Stokes equations. Mathematics of computation 22, 745–762.

Colas, C., Ferrand, M., Hérard, J.M., Latché, J.C., Le Coupanec, E., 2019. An Implicit Integral Formulation to Model Inviscid Fluid Flows in Obstructed Media. Computers & Fluids 188, 136–163.

Crespo, A.J., Gómez-Gesteira, M., Dalrymple, R.A., 2007. Boundary conditions generated by dynamic particles in sph methods. Computers, materials and continua 5, 173–184.

Das, R., Cleary, P.W., 2008. Modelling 3D fracture and fragmentation in a thin plate under high velocity projectile impact using SPH, in: Proc. III^{rd} SPHERIC International Workshop, Lausanne, Suisse.

Dehnen, W., Aly, H., 2012. Improving convergence in smoothed particle hydrodynamics simulations without pairing instability. Monthly Notices of the Royal Astronomical Society 425, 1068–1082.

Di Pietro, D.A., Ern, A., 2015. Hybrid high-order methods for variable-diffusion problems on general meshes. Comptes Rendus Mathématique 353, 31–34.

Edwards, J.D., Morel, J.E., Knoll, D.A., 2011. Nonlinear variants of the TR/BDF2 method for thermal radiative diffusion. Journal of Computational Physics 230, 1198 – 1214. URL: http://www.sciencedirect.com/science/article/pii/S0021999110005917, doi:https://doi.org/10.1016/j.jcp.2010.10.035.

Espanol, P., Revenga, M., 2003. Smoothed dissipative particle dynamics. Physical Review E 67, 026705.

Eymard, R., Gallouët, T., Guichard, C., Herbin, R., Masson, R., 2014. TP or not TP, that is the question. Computational Geosciences 18, 285–296. doi:10.1007/s10596-013-9392-9.

Eymard, R., Gallouët, T., Herbin, R., 2000. Finite volume methods. Handbook of Numerical Analysis 7, 713–1018.

Eymard, R., Gallouët, T., Herbin, R., 2006. Finite Volume Methods. volume 7.

Ferrand, M., 2009. Etude comparative de quelques modèles algébriques explicites pour les tensions de Reynolds et les flux turbulents de scalaires passifs et actifs et application au cas d'un étang saumâtre. Base documentaire Eureka .

Ferrand, M., Fontaine, J., Angelini, O., 2014. An anisotropic diffusion finite volume algorithm using a small stencil, in: Finite Volumes for Complex Applications VII-Elliptic, Parabolic and Hyperbolic Problems, pp. 577–585.

Ferrand, M., Joly, A., Kassiotis, C., Violeau, D., Leroy, A., Morel, F.X., Rogers, B.D., 2017. Unsteady open boundaries for SPH using semi-analytical conditions and Riemann solver in 2D. Computer Physics Communications 210, 29 – 44. URL: http://www.sciencedirect.com/science/article/pii/S0010465516302806, doi:https://doi.org/10.1016/j.cpc.2016.09.009.

Ferrand, M., Laurence, D.R., Rogers, B.D., Violeau, D., Kassiotis, C., 2013. Unified semi-analytical wall boundary conditions for inviscid, laminar or turbulent flows in the meshless SPH method. International Journal for Numerical Methods in Fluids 71, 446–472. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.3666, doi:10.1002/fld.3666, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.3666.

Fonty, T., 2019. Modélisation de l'entraînement d'air dans la méthode SPH .

Fonty, T., Ferrand, M., Leroy, A., Joly, A., Violeau, D., 2019. Mixture model for two-phase flows with high density ratios: A conservative and realizable SPH formulation. International Journal of Multiphase Flow 111, 158–174.

Fourtakas, G., Dominguez, J.M., Vacondio, R., Rogers, B.D., 2019. Local uniform stencil (lust) boundary condition for arbitrary 3-d boundaries in parallel smoothed particle hydrodynamics (sph) models. Computers & Fluids 190, 346–361.

Ghaïtanellis, A., Violeau, D., Ferrand, M., Abderrezzak, K.E.K., Leroy, A., Joly, A., 2018. A SPH elastic-viscoplastic model for granular flows and bed-load transport. Advances in Water Resources 111, 156–173.

Gingold, R.A., Monaghan, J.J., 1977. Smoothed Particle hydrodynamic: theory and application to non-spherical stars. Monthly Notices of the Royal Astronomical Society 181, 375–389.

Hayward, A.T.J., 1967. Compressibility equations for liquids: a comparative study. British Journal of Applied Physics 18, 965–977. URL: https://doi.org/10.1088%2F0508-3443%2F18%2F7%2F312, doi:10.1088/0508-3443/18/7/312.

Herant, M., 1994. Dirty tricks for sph. Memorie della Societa Astronomica Italiana 65, 1013.

Inutsuka, S., 1994. Godunov-type SPH. Memorie della Societa Astronomica Italiana 65, 1027.

Inutsuka, S.i., 2002. Reformulation of smoothed particle hydrodynamics with Riemann solver. arXiv preprint astro-ph/0206401 .

Issa, R., Lee, E.S., Violeau, D., Laurence, D., 2005. Incompressible separated flows simulations with the smoothed particle hydrodynamics gridless method. International Journal for Numerical Methods in Fluids 47, 1101–1106. URL: http://dx.doi.org/10.1002/fld.864, doi:10.1002/fld.864.

Issa, R.I., Gosman, A., Watkins, A., 1986. The computation of compressible and incompressible recirculating flows by a non-iterative implicit scheme. Journal of Computational Physics 62, 66–82.

Kostorz, W., Esmail-Yakas, A., 2021a. Semi-analytical smoothed-particle hydrodynamics correction factors for polynomial kernels and piecewise-planar boundaries. International Journal for Numerical Methods in Engineering .

Kostorz, W., Esmail-Yakas, A., 2021b. Semi-analytical treatment of spherical solids in smoothed-particle hydrodynamics fluid simulations. International Journal for Numerical Methods in Engineering 122, 1732–1745.

Kulasegaram, S., Bonet, J., Lewis, R.W., Profit, M., 2004. A variational formulation based contact algorithm for rigid boundaries in two-dimensional SPH applications. Computational Mechanics 33, 316–325.

Layton, W., Trenchea, C., 2012. Stability of two IMEX methods, CNLF and BDF2-AB2, for uncoupling systems of evolution equations. Applied Numerical Mathematics 62, 112–120.

Leroy, A., Violeau, D., Ferrand, M., Fratter, L., Joly, A., 2016. A new open boundary formulation for incompressible SPH. Computers & Mathematics with Applications 72, 2417 – 2432. URL: http://www.sciencedirect.com/science/article/pii/S0898122116305107, doi:https://doi.org/10.1016/j.camwa.2016.09.008.

Leroy, A., Violeau, D., Ferrand, M., Kassiotis, C., 2014. Unified semi-analytical wall boundary conditions applied to 2-D incompressible SPH. Journal of Computational Physics 261, 106 – 129. URL: http://www.sciencedirect.com/science/article/pii/S0021999113008437, doi:https://doi.org/10.1016/j.jcp.2013.12.035.

Leveque, R.J., 1992. Numerical methods for conservation laws. volume 132. Springer.

Lind, S., Xu, R., Stansby, P., Rogers, B., 2012. Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves. Journal of Computational Physics 231, 1499 – 1523. URL: http://www.sciencedirect.com/science/article/pii/S0021999111006279, doi:https://doi.org/10.1016/j.jcp.2011.10.027.

Lucy, L.B., 1977. A Numerical Approach to Testing the Fission Hypothesis. The Astronomical Journal 82, 1013–1924.

Marongiu, J.C., Leboeuf, F., Caro, J., Parkinson, E., 2010. Free surface flows simulations in Pelton turbines using an hybrid SPH-ALE method. Journal of Hydraulic Research 48, 40–49. URL: https://doi.org/10.1080/00221686.2010.9641244, doi:10.1080/00221686.2010.9641244, arXiv:https://doi.org/10.1080/00221686.2010.9641244.

Marrone, S., Antuono, M., Colagrossi, A., Colicchio, G., Le Touzé, D., Graziani, G., 2011. δ-sph model for simulating violent impact flows. Computer Methods in Applied Mechanics and Engineering 200, 1526–1542.

Mayrhofer, A., Ferrand, M., Kassiotis, C., Violeau, D., Morel, F.X., 2015. Unified semi-analytical wall boundary conditions in SPH: analytical extension to 3-D. Numerical Algorithms 68, 15–34.

Mayrhofer, A., Laurence, D., Rogers, B., Violeau, D., 2014. Large eddy simulation with SPH: Mission impossible, in: Proc. 9th international SPHERIC workshop, pp. 216–222.

Méchitoua, N., Boucker, M., Laviéville, J., Hérard, J., Pigny, S., Serre, G., 2003. Neptune-CFD: an Unstructured FV Solver for two-phase Water/Vapor Flows Based on an Elliptic Oriented Fractional Step Method. NURETH-10, Seoul, Oct. 5 9.

Monaghan, J.J., 1992. Smoothed Particle Hydrodynamics. Annual review of Astronomy and Astrophysics 30, 543–574.

Monaghan, J.J., 1994. Simulating free surface flows with SPH. Journal of Computational Physics 110, 399–406.

Monaghan, J.J., 1999. SPH without a tensile instability. Journal of Computational Physics 159, 290–311.

Monaghan, J.J., Gingold, R.A., 1983. Shock simulation by the particle method sph. Journal of computational physics 52, 374–389.

Morris, J.P., Fox, P.J., Zhu, Y., 1997. Modeling Low Reynolds Number Incompressible Flows Using SPH. Journal of Computational Physics 136, 214–226.

Neuhauser, M., Leboeuf, F., Marongiu, J.C., Parkinson, E., Robb, D., 2014. Simulations of Rotor–Stator Interactions with SPH-ALE. Springer Singapore, Singapore. pp. 349–361. URL: https://doi.org/10.1007/978-981-4451-42-0_29, doi:10.1007/978-981-4451-42-0_29.

Oger, G., Marrone, S., Le Touzé, D., De Leffe, M., 2016. SPH accuracy improvement through the combination of a quasi-Lagrangian shifting transport velocity and consistent ALE formalisms. Journal of Computational Physics 313, 76–98.

Patankar, S.V., 1980. Numerical heat transfer and fluid flow .

Patsis, P.A., Athanassoula, E., 2000. SPH simulations of gas flow in barred galaxies. Effect of hydrodynamical and numerical parameters. aap 358, 45–56.

Pilliod, J.E., Puckett, E.G., 2004. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. Journal of Computational Physics 199, 465 – 502. URL: http://www.sciencedirect.com/science/article/pii/S0021999104000920, doi:https://doi.org/10.1016/j.jcp.2003.12.023.

Rhie, C., Chow, W.L., 1983. Numerical study of the turbulent flow past an airfoil with trailing edge separation. AIAA journal 21, 1525–1532.

Robinson, M.J., 2009. Turbulence and viscous mixing using smoothed particle hydrodynamics. Ph.D. thesis. Monash University.

Sun, P., Colagrossi, A., Marrone, S., Zhang, A., 2017. The δplus-SPH model: Simple procedures for a further improvement of the SPH scheme. Computer Methods in Applied Mechanics and Engineering 315, 25–49.

Temam, R., 1968. Une méthode d'approximation de la solution des équations de navier-stokes. Bulletin de la Société Mathématique de France 96, 115–152.

Toro, E.F., 2013. Riemann solvers and numerical methods for fluid dynamics: a practical introduction. Springer Science & Business Media.

Vacondio, R., Altomare, C., De Leffe, M., Hu, X., Le Touzé, D., Lind, S., Marongiu, J.C., Marrone, S., Rogers, B.D., Souto-Iglesias, A., 2021. Grand challenges for smoothed particle hydrodynamics numerical schemes. Computational Particle Mechanics 8, 575–588.

Vacondio, R., Rogers, B.D., Stansby, P.K., Mignosa, P., 2011. SPH modeling of shallow flow with open boundaries for practical flood simulation. Journal of Hydraulic Engineering 138, 530–541.

Versteeg, H.K., Malalasekera, W., 2007. An introduction to computational fluid dynamics the finite volume method. 2^{nd} ed., Pearson Education Ltd., Harlow (GB) New York Boston [etc.]. URL: http://www.sudoc.fr/11762473X.

Vila, J.P., 1999. On particle weighted methods and smooth particle hydrodynamics. Mathematical Models and Methods in Applied Sciences 9, 161–209.

Violeau, D., 2012. Fluid Mechanics and the SPH Method: Theory and Applications. Oxford University Press. URL: http://ukcatalogue.oup.com/product/academic/earthsciences/hydrology/9780199655526.do.

Violeau, D., Leroy, A., 2014. On the maximum time step in weakly compressible SPH. Journal of Computational Physics 256, 388–415.

Violeau, D., Leroy, A., Mayrhofer, A., 2014. Exact computation of SPH wall renormalising integrals in 3-D. by D. Violeau, A. Hérault, and A. Joly. Paris , 95–102.

Wall, C., Pierce, C.D., Moin, P., 2002. A semi-implicit method for resolution of acoustic waves in low Mach number flows. Journal of Computational Physics 181, 545–563.

Wendland, H., 1995. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. Advances in Computational Mathematics 4, 389–396.

Werdelmann, B., Koch, R., Krebs, W., Bauer, H.J., 2021. An approach for permeable boundary conditions in sph. Journal of Computational Physics 444, 110562.

Xu, R., 2009. An Improved Incompressible Smoothed Particle Hydrodynamics Method and Its Application in Free-Surface Simulations. Ph.D. thesis. University of Manchester Institute of Science and Technology. Manchester, UK.

2. Unified semi-analytical wall boundary conditions for inviscid, laminar or turbulent flows in the mesh-less SPH method

Abstract 2.1 Wall boundary conditions in smoothed particle hydrodynamics (SPH) are a key issue to perform accurate simulations. We propose here a new approach based on a renormalising factor for writing all boundary terms. This factor depends on the local shape of the wall and on the position of a particle relative to the wall, which is described by segments (in 2-D), instead of the cumbersome fictitious or ghost particles used in most of the existing SPH models. By solving a dynamic equation for the renormalising factor, we significantly improve traditional wall treatment in SPH, for pressure forces, wall friction and turbulent conditions. The new model is demonstrated for cases including hydrostatic conditions for still water in a tank of complex geometry and a dam break over triangular bed profile with sharp angle where significant improved behaviour is obtained in comparison with the conventional boundary techniques. The latter case is also compared to a VOF scheme. The performance of the model is tested on a 2-D laminar flow which show that the profiles of velocity are in good agreement with the theoretical ones. This demonstrates that the derived wall shear stress balances correctly the pressure gradient. Finally, the performance of the model is demonstrated for flow in a schematic fish-pass where velocity field and turbulent viscosity field are satisfactorily reproduced compared to the result of mesh-based codes.

This chapter is an adaptation of Ferrand et al. (2013).

Résumé 2.1 Le traitement des conditions aux limites de paroi dans la méthode *smoothed particle Hydrodynamics* (SPH) est fondamental pour effectuer des simulations précises. Nous proposons ici une nouvelle approche basée sur un facteur de renormalisation permettant d'expliciter tous les termes de bord. Ce facteur dépend uniquement de la forme des frontières solides et de la position d'une particule par rapport à celles-ci, décrites par des segments en dimension 2 en lieu et place de particules fictives ou fantômes encore largement utilisées. En résolvant une équation d'évolution sur le facteur de renormalisation, le traitement proche paroi des forces de pression, du frottement visqueux et des grandeurs turbulentes est amélioré de manière significative comparé aux approches traditionnelles. Cette approche est confrontée à différents cas tests comme l'équilibre hydrostatique dans un bassin de géométrie complexe, une rupture de barrage par dessus une rampe triangulaire d'angle aigu avec des améliorations

significatives comparé à des conditions aux limites conventionnelles (le dernier cas est aussi comparé à une simulation VOF). Les performances du modèle sont testées sur écoulement de Poiseuille laminaire plan permettant de vérifier les profils de vitesse et de démontrer que les frottements visqueux équilibrent correctement le gradient de pression moteur. Pour finir, le modèle est testé sur une configuration de passe à poisson schématique pour laquelle les champs de vitesse, et turbulence sont cohérents avec des résultats obtenus avec la méthode des volumes finis.

Ce chapitre est une adaptation de Ferrand et al. (2013).

2.1 Introduction

Treating wall boundary conditions is one of the most challenging parts of the smoothed particle hydrodynamics (SPH) method and many different approaches have been developed before the publication of this work in Ferrand et al. (2013) (see e.g. Kulasegaram et al. (2004); Oger et al. (2007); Di Monaco et al. (2009); Monaghan and Kajtar (2009a); Marongiu et al. (2009); De Leffe et al. (2009)). An updated review of progress on boundary conditions can be found in Vacondio et al. (2021). Accurate boundary conditions are essential since in many applications where precise loading on walls is required such as forces on floating bodies or shoreline structures, tank walls, wind-wave exchanges, fluid-structure interactions in power-plants etc. It is an obvious prerequisite to improve turbulence modelling near walls.

Many methods for implementing solid walls in SPH have been developed over the past two decades. Among the most popular and commonly used include three broad categories:

1. repulsive forces such as Lennard-Jones potential Monaghan (1994), or more recently Monaghan and Kajtar (2009b),

2. fictitious particles (or alternatively ghost particles) which fill the empty area of the kernel support behind a boundary with artificial particles with prescribed physical quantities (such as pressure and velocity) to enforce no slip or free slip condition,

3. semi-analytical boundary conditions based on a variational formulation introduced by Kulasegaram et al. (2004) where a wall renormalisation of the equations is made with respect to the missing area of the kernel support. We will describe a variant of this method in § 2.3, where intrinsic *gradient* and *divergence* operators that ensure conservation properties are employed.

All of these methods have advantages and drawbacks. The Lennard-Jones potential force originally describes the interaction between pairs of atoms and is used to model the repulsion between a fluid particle and a boundary particle. This method is easy to implement, even for complex geometries and computationally cheap. However, it leads to spurious behaviour: for instance it is impossible to maintain particles fixed along a vertical wall in the presence of gravity.

The fictitious particles recommended in Violeau and Issa (2007) prevent such a non-physical behaviour. However, the positioning of ghost particles in complex geometries can be particularly unwieldy, particularly in 3-D. Moreover, the computational effort required is not negligible, given that we increase the number of particles to take into account in the discrete summations (2.1) and (2.3). This might be rethought with recent proposition such as Fourtakas et al. (2019), but smooth fields next to the boundary are partially due to the use of the δ-SPH techniques introduced in Antuono et al. (2010).

Finally, the semi-analytical approach is attractive thanks to its variational derivation which means that some physical quantities such as momentum will be automatically conserved. Unfortunately, the original attempt did not present a clear and simple way to compute renormalisation terms introduced. Furthermore, the formulation proposed was not able to reproduce hydrostatic pressure fields or to take into account the shear stress along a wall.

This chapter therefore takes the semi-analytical approach of Kulasegaram et al. (2004) and extends it so that the accuracy of the physical field such as the pressure next to walls is considerably improved, and the consistent manner developed for wall-correct operators allows us to perform simulations with turbulence models. This work belongs to the family of boundary integrals which as become increasingly popular since the publication of this work (see e.g. Chiron et al., 2019, for a recent review and development in 3-D). This work will present three key advances:

- The time integration scheme used for the continuity equation requires particular attention, and as already mentioned by Vila (1999), we prove there is no point in using a dependence in time of the particles' density if no kernel gradient corrections are added. Thus, by using a near-boundary kernel-corrected version of the time integration scheme of the form proposed in Vila (1999), long-time simulations ideally suited for turbulent flows in the context of accurate boundary conditions are possible.

- To compute the kernel correction, Feldman and Bonet (2007) use an analytical value which is computationally expensive whereas Kulasegaram et al. (2004) and De Leffe et al. (2009) use polynomial approximation which can be difficult to define for complex geometries. We propose here to compute the renormalisation term of the kernel support near a solid with a novel time integration scheme, allowing us any shape for the boundary.

- All boundary terms issued from the continuous approximation are given by surface summations which only require information from a CAD mesh file of the boundary. The technique developed here allows us to correct the pressure gradient and viscous terms and hence provides a physically correct wall-shear stress so that even the diffusion equation of a scalar quantity can be solved accurately using SPH such as the turbulent kinetic energy or its dissipation in a $k-\epsilon$ model of turbulence.

This chapter is organised as follows: in the next Section we introduce the formulations for weakly compressible SPH introducing the basic discretisations for operators, viscous forces, and turbulence modelling. In the Section following, we then develop the consistent boundary conditions with improved time integration and a position-dependent technique to compute the density. This Chapter then presents the computation of the renormalisation terms using a novel time integration scheme before presenting the enhanced behaviour in the Section on numerical results.

2.2 Basic SPH formulations for weakly compressible Newtonian fluid

2.2.1 Conservative governing equations

The slightly compressible Newtonian fluid is modelled by a set of particles denoted by the subscripts $(.)_a$ and $(.)_b$ in a domain Ω. The set of all the fluid particles is denoted by $\mathcal{P}$ where each particle $a \in \mathcal{P}$ possesses information such as its mass M_a (assumed constant), its position $\boldsymbol{x}_a$, its velocity $\boldsymbol{u}_a$ (the Lagrangian derivative of the position), its density ρ_a, its volume $V_a = \frac{M_a}{\rho_a}$, and its pressure p_a. The spatial discretisation is based on a weighting interpolation or kernel function, w, with compact support. Ω_a then refers to the support of the kernel function centred or $\boldsymbol{x}_a$ of radius R. We generally denote by the subscripts $(.)_{ab}$ the difference of a quantity between the positions a and b. For instance $\boldsymbol{u}_{ab} := \boldsymbol{u}_a - \boldsymbol{u}_b$ and $\boldsymbol{x}_{ab} := \boldsymbol{x}_a - \boldsymbol{x}_b$. Two exceptions are made with the following notations $w_{ab} := w(\boldsymbol{x}_{ab})$ and $\boldsymbol{\nabla} w_{ab} := \boldsymbol{\nabla}_a w(\boldsymbol{x}_{ab})$. Here, the symbol $\boldsymbol{\nabla}_a$ denotes the *gradient* at the point $\boldsymbol{x}_a$ [1]. With these notations, a commonly used form of the continuity equation is (see e.g. Monaghan (1992)):

$$\frac{\mathrm{d}\rho_a}{\mathrm{d}t} = \sum_{b \in \mathcal{P}} M_b \boldsymbol{\nabla} w_{ab} \cdot \boldsymbol{u}_{ab}, \tag{2.1}$$

[1] The gradient operator is defined by $\boldsymbol{\nabla} := \boldsymbol{e}_x \frac{\partial}{\partial x} + \boldsymbol{e}_y \frac{\partial}{\partial y} + \boldsymbol{e}_z \frac{\partial}{\partial z}$, $(\boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z)$ being the basis vector triad of the Cartesian coordinate system in 3-D.

where $\frac{\mathrm{d}}{\mathrm{d}t}$ denotes the Lagrangian derivative, that is to say the derivative along the particle path. It can be derived from the following definition of the density:

$$\rho_a = \sum_{b \in \mathcal{P}} M_b w_{ab}. \tag{2.2}$$

The inviscid momentum equation can be written as follows:

$$\frac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t} = -\sum_{b \in \mathcal{P}} M_b \left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2} \right) \boldsymbol{\nabla} w_{ab} + \boldsymbol{g}, \tag{2.3}$$

where $\boldsymbol{g}$ is gravity. The equation of state links density and pressure:

$$p_a = \frac{\rho_0 c_0^2}{\gamma} \left[\left(\frac{\rho_a}{\rho_0} \right)^\gamma - 1 \right], \tag{2.4}$$

where ρ_0 is the reference density of the fluid, c_0 is the speed of sound and $\gamma = 7$ is generally chosen for water.

2.2.2 Viscous forces

The viscous term is used in this work for computing the viscous term $\frac{1}{\rho}\nabla \cdot (\mu \boldsymbol{\nabla} \boldsymbol{u})$ is due to Morris et al. (see Morris et al. (1997)):

$$\frac{1}{\rho_a} \boldsymbol{\nabla} \cdot (\mu \boldsymbol{\nabla} \boldsymbol{u})_a = \sum_{b \in \mathcal{P}} M_b \frac{\mu_a + \mu_b}{\rho_a \rho_b} \frac{\boldsymbol{u}_{ab}}{r_{ab}^2} \boldsymbol{r}_{ab} \cdot \boldsymbol{\nabla} w_{ab}, \tag{2.5}$$

where the dynamic viscosity μ is given by:

$$\mu := \nu \rho, \tag{2.6}$$

and ν is the kinematic molecular viscosity.

Operator definitions:

For later analysis, we define discrete operators gradient ($\mathbf{Grad}_a^+ \{A_b\}$), divergence ($\mathrm{Div}_a^- \{\boldsymbol{A}_b\}$) and Laplacian ($\mathbf{Lap}_a (\{B_b\}, \{\boldsymbol{A}_b\})$) of arbitrary discrete scalar ($\{A_b\}$ and $\{B_b\}$) or vector fields ($\{\boldsymbol{A}_b\}$) as:

$$\begin{aligned}
\mathbf{Grad}_a^+ \{A_b\} &:= \rho_a \sum_{b \in \mathcal{P}} M_b \left(\frac{A_a}{\rho_a^2} + \frac{A_b}{\rho_b^2} \right) \boldsymbol{\nabla} w_{ab}, \\
\mathrm{Div}_a^- \{\boldsymbol{A}_b\} &:= -\frac{1}{\rho_a} \sum_{b \in \mathcal{P}} M_b \boldsymbol{A}_{ab} \cdot \boldsymbol{\nabla} w_{ab}, \\
\mathbf{Lap}_a (\{B_b\}, \{\boldsymbol{A}_b\}) &:= \rho_a \sum_{b \in \mathcal{P}} M_b \frac{B_a + B_b}{\rho_a \rho_b} \frac{\boldsymbol{A}_{ab}}{r_{ab}^2} \boldsymbol{r}_{ab} \cdot \boldsymbol{\nabla} w_{ab},
\end{aligned} \tag{2.7}$$

where $\boldsymbol{A}_{ab} := \boldsymbol{A}_a - \boldsymbol{A}_b$. They are approximations of the continuous *gradient*, *divergence* and Laplacian operators (respectively denoted by $\boldsymbol{\nabla}_a$, $\boldsymbol{\nabla}.$ and $\boldsymbol{\nabla} \cdot \boldsymbol{\nabla}$). Thus the equation of continuity (2.1) and the momentum equation (2.3) can be rewritten as follows:

$$\begin{aligned}
\frac{\mathrm{d}\rho_a}{\mathrm{d}t} &= -\rho_a \mathrm{Div}_a^- \{\boldsymbol{u}_b\}, \\
\frac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t} &= -\frac{1}{\rho_a} \mathbf{Grad}_a^+ \{p_b\} + \boldsymbol{g} + \frac{1}{\rho_a} \mathbf{Lap}_a (\{\mu_b\}, \{\boldsymbol{u}_b\}).
\end{aligned} \tag{2.8}$$

The operators $\mathbf{Grad}_a^+ \{.\}$ and $\mathrm{Div}_a^- \{.\}$ are said to be skew-adjoint. It is possible to define variants of these three operators conserving this property of adjointness (see Violeau (2012)).

2.2.3 Turbulence modelling in SPH

The Reynolds approach consists in considering only the mean part[2] denoted by $\overline{\boldsymbol{u}}$ of the velocity field $\boldsymbol{u}$ in the equation of Navier–Stokes, then modelling the effects of the fluctuating part of the velocity field on the mean velocity.

The $k-\epsilon$ turbulence model developed by Launder and Spalding (1972) add two transport equations of k, the turbulent kinetic energy, and ϵ, its dissipation, to the momentum equation (2.8) which is modified as follows:

$$\begin{aligned}
\frac{\mathrm{D}\rho_a}{\mathrm{D}t} &= -\rho_a \mathrm{Div}_a^- \{\overline{\boldsymbol{u}}_b\}, \\
\frac{\mathrm{D}\overline{\boldsymbol{u}}_a}{\mathrm{D}t} &= \frac{1}{\rho_a}\mathbf{Grad}_a^+ \left\{\overline{p}_b + \tfrac{2}{3}\rho k_b\right\} + \mathbf{Lap}_a\left(\{\mu_b + \mu_{Tb}\}, \{\overline{\boldsymbol{u}}_b\}\right) + \boldsymbol{g}, \\
\frac{\mathrm{D}k_a}{\mathrm{D}t} &= \frac{1}{\rho_a}\mathrm{Lap}_a\left(\left\{\mu_b + \frac{\mu_{Tb}}{\sigma_k}\right\}, \{k_b\}\right) + P_a - \epsilon_a, \\
\frac{\mathrm{D}\epsilon_a}{\mathrm{D}t} &= \frac{1}{\rho_a}\mathrm{Lap}_a\left(\left\{\mu_b + \frac{\mu_{Tb}}{\sigma_\epsilon}\right\}, \{\epsilon_b\}\right) + \frac{\epsilon_a}{k_a}\left(C_{\epsilon 1}P_a - C_{\epsilon 2}\epsilon_a\right),
\end{aligned} \tag{2.9}$$

where the derivative $\dfrac{\mathrm{D}}{\mathrm{D}t} := \dfrac{\partial}{\partial t} + \overline{\boldsymbol{u}} \cdot \boldsymbol{\nabla}$ is the Lagrangian derivative along the Reynolds averaged field $\{\overline{\boldsymbol{u}}_b\}$. The $k-\epsilon$ model links the turbulent kinetic viscosity $\nu_T := \frac{\mu_T}{\rho}$ to the turbulent kinetic energy k and its dissipation ϵ by:

$$\nu_{Ta} = C_\mu \frac{k_a^2}{\epsilon_a}, \tag{2.10}$$

with constants σ_k, σ_ϵ, $C_{\epsilon 1}$ and $C_{\epsilon 2}$ given by Launder and Spalding (1972).

The production term of k, P_a, is defined by:

$$P_a = \nu_{Ta} S_a^2, \tag{2.11}$$

where $S_a^2 := 2\boldsymbol{S}_a : \boldsymbol{S}_a$ is the scalar mean rate-of-strain. The tensor strain rate of the mean velocity field is given by $\boldsymbol{S}_a := \frac{1}{2}\left(\boldsymbol{\nabla}_a \overline{\boldsymbol{u}} + \boldsymbol{\nabla}_a \overline{\boldsymbol{u}}^T\right)$; Violeau and Issa (2007) discretised in the SPH form the velocity *gradient*:

$$\mathbf{Grad}_a^- \{\overline{\boldsymbol{u}}_b\} := -\frac{1}{\rho_a}\sum_{b \in \mathcal{P}} M_b \overline{\boldsymbol{u}}_{ab} \otimes \boldsymbol{\nabla} w_{ab}. \tag{2.12}$$

In the following, for sake of simplicity we will drop all overbars to denote velocity and pressure, and thus denote Lagrangian derivatives $\frac{\mathrm{d}}{\mathrm{d}t}$. The reader should keep in mind that under turbulent conditions, the latter quantities are considered as Reynolds-averaged and the Lagrangian derivative is along the Reynolds averaged velocity field.

2.3 New boundary conditions and accurate time stepping

2.3.1 Derivation of wall boundary terms using continuous interpolation

Kulasegaram et al.'s renormalisation:

Instead of assuming that $\rho_a \simeq \sum_{b \in \mathcal{P}} M_b w_{ab}$, which underestimates ρ_a when the particle a is close to a boundary (see Figure 2.1), Kulasegaram et al. (2004) renormalise the estimation using a function γ_a:

$$\rho_a \simeq \frac{1}{\gamma_a}\sum_{b \in \mathcal{P}} M_b w_{ab}, \tag{2.13}$$

[2] in the stochastic point of view.

where γ_a is defined by:

$$\gamma_a := \int_{\Omega \cap \Omega_a} w\left(\boldsymbol{x}' - \boldsymbol{x}_a\right) \mathrm{d}V'. \tag{2.14}$$

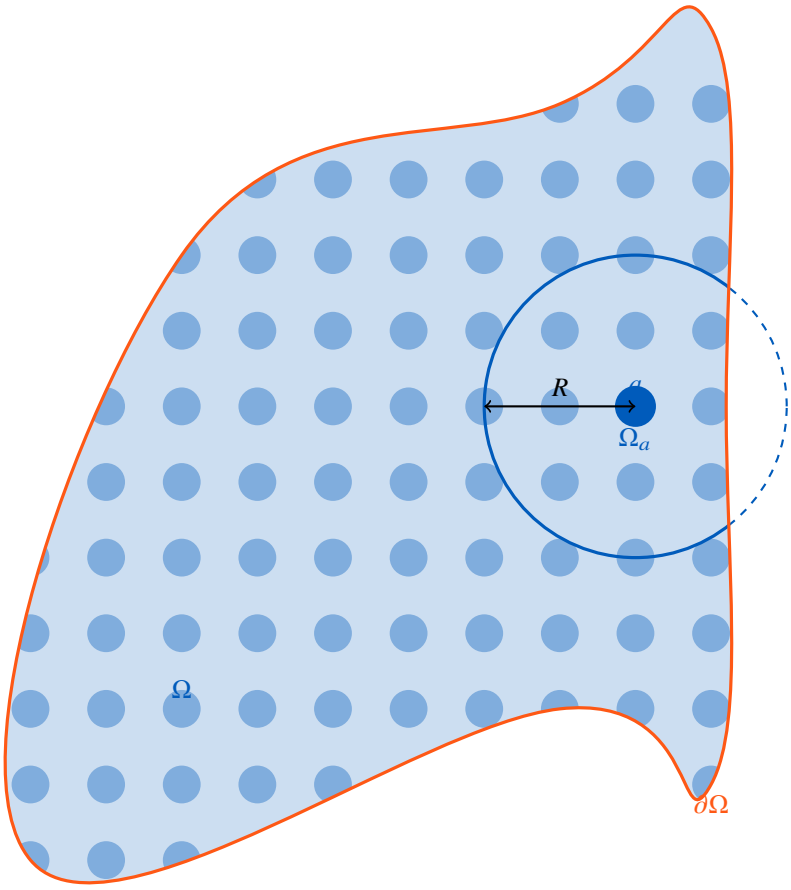


Figure 2.1: Kernel-boundary interaction.

In general, γ_a is an Eulerian field depending only on the position of the particle a with respect to boundaries of Ω. Far from a solid boundary $\gamma_a = 1$.

The introduction of γ_a into the derivation of the governing equations leads to some key differences such as the new continuity equation of Kulasegaram et al. (2004). (2.13) is rewritten:

$$\frac{\mathrm{d}\rho_a}{\mathrm{d}t} = \frac{1}{\gamma_a} \sum_{b \in \mathcal{P}} M_b \boldsymbol{\nabla} w_{ab} \cdot \boldsymbol{u}_{ab} - \frac{\rho_a}{\gamma_a} \boldsymbol{\nabla} \gamma_a \cdot \boldsymbol{u}_a. \tag{2.15}$$

This can be compared with (2.1). Accordingly, the gradient, $\boldsymbol{\nabla}\gamma_a$ is defined by:

$$\boldsymbol{\nabla}\gamma_a := \int_{\Omega \cap \Omega_a} \boldsymbol{\nabla}_a w\left(\boldsymbol{x}' - \boldsymbol{x}_a\right) \mathrm{d}V' = \int_{\partial\Omega \cap \Omega_a} w\left(\boldsymbol{x}' - \boldsymbol{x}_a\right) \boldsymbol{n} \mathrm{d}S', \tag{2.16}$$

where the second integral is obtained using the Gauss theorem and where $\boldsymbol{n}$ is the inward boundary normal.

In order to evaluate the new internal forces and contact forces, Kulasegaram et al. (2004) derive the internal energy using the equation of Lagrange. They obtain a new internal force due to the pressure. The correction of Kulasegaram et al. (2004) can be expressed in terms of new compatible (in a variational sense) operators gradient and divergence. Indeed, one can define:

$$\begin{aligned} \mathbf{Grad}_a^K \{A_b\} &:= \rho_a \sum_{b \in \mathcal{P}} M_b \left(\frac{A_a}{\gamma_a \rho_a^2} + \frac{A_b}{\gamma_b \rho_b^2} \right) \boldsymbol{\nabla} w_{ab} - \frac{A_a}{\gamma_a} \boldsymbol{\nabla} \gamma_a, \\ \mathrm{Div}_a^K \{\boldsymbol{A}_b\} &:= -\frac{1}{\gamma_a \rho_a} \sum_{b \in \mathcal{P}} M_b \boldsymbol{A}_{ab} . \boldsymbol{\nabla} w_{ab} + \frac{1}{\gamma_a} \boldsymbol{A}_a \cdot \boldsymbol{\nabla} \gamma_a . \end{aligned} \tag{2.17}$$

Thus the operators (2.17) verify the skew-adjointness property and variational calculus still holds, thus proving the consistency between the momentum equation and the continuity equation. The final term of $\mathbf{Grad}_a^K \{A_b\}$ was used by Kulasegaram et al. (2004) as a boundary force. In the following, we propose an enhanced form of the latter model, giving a more accurate representation of gradients on the wall.

General shape of the boundary:

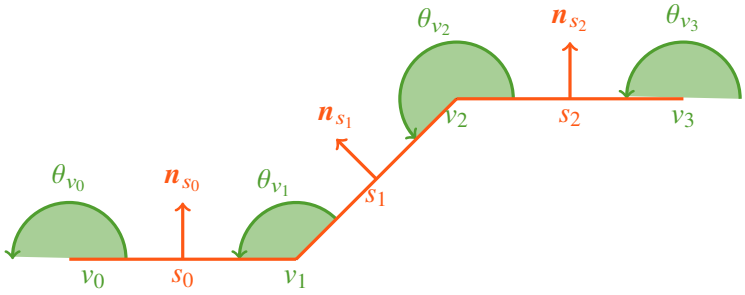


Figure 2.2: Boundary property definitions. Boundary with vertex *semi* particles v (in green) and with segments s which have a surface S_s and an inward normal $\boldsymbol{n}_s$.

For boundaries of arbitrary shape, their shape, $\partial\Omega$ of the domain Ω, is approximated with straight segments in 2-D denoted by the subscript $(.)_s$ which have a normal $\boldsymbol{n}_s$ and a surface area S_s (see Figure 2.2). The set containing all the segments is denoted by $\mathcal{S}$. Each segment is defined by two vertex points denoted by the subscripts $(.)_{v1}$ and $(.)_{v2}$ which have an initial volume V_v given by $V_v = \frac{M_v}{\rho_0}$ where ρ_0 is a reference density. The initial volume of vertex particles is a fraction of the initial volume of fluid particles V_f. For instance, for a plane $V_v = \frac{1}{2} V_f$. More generally, for a vertex particle on a wedge with an angle θ displayed on the Figure 2.2, $V_v = \frac{\theta}{2\pi} V_f$.

The set containing all the vertex particles is denoted by $\mathcal{V}$. These vertex particles (also called semi particles in the following) are of particular interest for recording the pressure field at the solid boundary (and hence for fluid and structure coupling for example). They are also useful to improve accuracy of the continuity equation, as they mimic a wet wall. It is important to notice that they are taken into account in the continuity equation and in the momentum equation (i.e. $\mathcal{V} \subset \mathcal{P}$), even if they are Eulerian particles, that is to say they are fixed if the wall is motionless and does not depend on the momentum equation.

Next we define the contribution of the segment s in the value of $\boldsymbol{\nabla} \gamma_a$ to be:

$$\boldsymbol{\nabla} \gamma_{as} := \left(\int_{\boldsymbol{x}_{v1}}^{\boldsymbol{x}_{v2}} w(r) \, \mathrm{d}l \right) \boldsymbol{n}_s, \tag{2.18}$$

and then $\boldsymbol{\nabla} \gamma_a$ can be decomposed in:

$$\boldsymbol{\nabla} \gamma_a = \sum_{s \in \mathcal{S}} \boldsymbol{\nabla} \gamma_{as}. \tag{2.19}$$

The description of the boundary geometry can be extended to 3-D by substituting the segments by triangles. In this case this volume V_v of vertex particles would use the solid angle of the wedge.

Wall corrected *gradients*:

The main disadvantage of previous techniques to evaluate $\boldsymbol{\nabla}\gamma_a$ is that the *gradient* operator defined by (2.17) is not accurate near a boundary or the free-surface. Indeed, we can see that *gradients* of constants are non zero: if we simulate uniform overpressure in a periodic pipe without any body force, the *gradient* (2.17) of the pressure is not zero everywhere and particles rearrange themselves (see Figure 2.3(a)). To correct that, we can go back to the continuous interpolation of an arbitrary integrable function f at a point $\boldsymbol{x}$:

$$\langle f \rangle (\boldsymbol{x}) = \frac{1}{\gamma(\boldsymbol{x})} \int_{\Omega \cap \Omega_r} f(\boldsymbol{x}') \, w(\tilde{r}) \, \mathrm{d}V', \tag{2.20}$$

where $\tilde{r} := |\boldsymbol{x} - \boldsymbol{x}'|$ and Ω_r is the kernel support centred in $\boldsymbol{x}$. By interpolating the gradient of the function f in the same way, it becomes:

$$\begin{aligned} \langle \boldsymbol{\nabla} f \rangle (\boldsymbol{x}) = & -\frac{1}{\gamma(\boldsymbol{x})} \int_{\Omega \cap \Omega_r} f(\boldsymbol{x}') \, \boldsymbol{\nabla} w(\tilde{r}) \, \mathrm{d}V' \\ & -\frac{1}{\gamma(\boldsymbol{x})} \int_{\partial\Omega \cap \Omega_r} f(\boldsymbol{x}') \, w(\tilde{r}) \, \boldsymbol{n} \mathrm{d}S', \end{aligned} \tag{2.21}$$

where the right-hand-side is obtained by an integration by parts and $\boldsymbol{n}$ is the inward normal of the domain at the position $\boldsymbol{r}'$. Hence, we can see that the boundary conditions appear naturally through the second integral of (2.21).

Moreover, if we consider that the gradient defined by (2.7) is a discrete approximation of the continuous gradient $\boldsymbol{\nabla} f := \rho \boldsymbol{\nabla} \frac{f}{\rho} + \frac{f}{\rho} \boldsymbol{\nabla} \rho$ (to obtain a symmetric formulation), we obtain (see also De Leffe et al. (2009)):

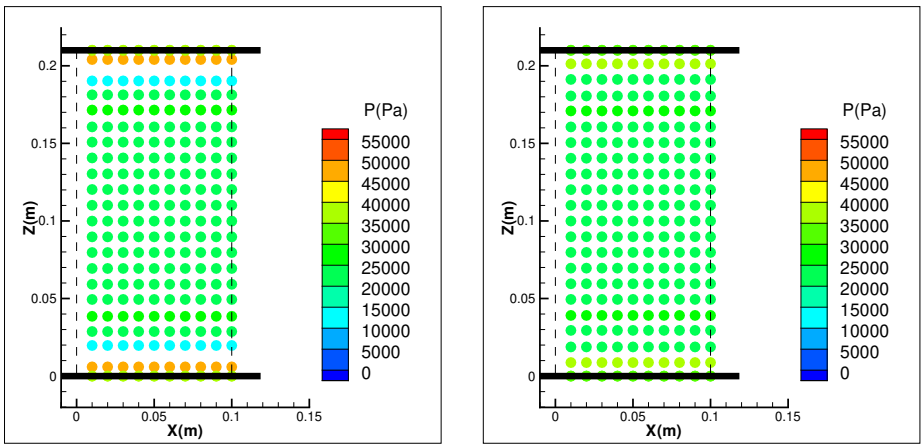
$$\begin{aligned} \langle \boldsymbol{\nabla} f \rangle (\boldsymbol{x}) &\simeq \left\langle \rho(\boldsymbol{x}) \boldsymbol{\nabla} \frac{f}{\rho} + \frac{f}{\rho}(\boldsymbol{x}) \boldsymbol{\nabla} \rho \right\rangle (\boldsymbol{x}) \\ &= -\frac{1}{\gamma(\boldsymbol{x})} \int_{\Omega \cap \Omega_r} \left[\frac{f}{\rho}(\boldsymbol{x}') \, \rho(\boldsymbol{x}) + \frac{f}{\rho}(\boldsymbol{x}) \, \rho(\boldsymbol{x}') \right] \boldsymbol{\nabla} w(\tilde{r}) \, \mathrm{d}V' \\ &\quad -\frac{1}{\gamma(\boldsymbol{x})} \int_{\partial\Omega \cap \Omega_r} \left[\frac{f}{\rho}(\boldsymbol{x}') \, \rho(\boldsymbol{x}) + \frac{f}{\rho}(\boldsymbol{r}) \, \rho(\boldsymbol{x}') \right] w(\tilde{r}) \, \boldsymbol{n} \mathrm{d}S'. \end{aligned} \tag{2.22}$$

Once again the boundary conditions appear naturally, and we can now set the following discrete operator *gradient* for an arbitrary field $\{A_b\}$ as:

$$\mathbf{Grad}_a^{\gamma^+} \{A_b\} := \frac{\rho_a}{\gamma_a} \sum_{b \in \mathcal{P}} M_b \left(\frac{A_a}{\rho_a^2} + \frac{A_b}{\rho_b^2} \right) \boldsymbol{\nabla} w_{ab} - \frac{\rho_a}{\gamma_a} \sum_{s \in \mathcal{S}} \left(\frac{A_a}{\rho_a^2} + \frac{A_s}{\rho_s^2} \right) \rho_s \boldsymbol{\nabla} \gamma_{as}. \tag{2.23}$$

The method to compute $\boldsymbol{\nabla}\gamma_{as}$ will be discussed in § 2.3.5 whereas the computation of ρ_s and A_s will be investigated in the paragraph on dynamic boundary conditions 2.3.2. Notice that the discrete gradient (2.23) differs from Kulasegaram et al. (2004)'s defined by (2.17) only next to a boundary, which means that conservation properties are still valid "far from" the walls.

Let us now correct using the same idea the SPH *gradient* of a vectorial field which appears in the velocity gradient (2.12) for instance. The latter quantity plays a key role in the $k - \epsilon$ turbulence model as it is responsible for the production of kinetic energy. Given the fact that in a channel flow the strain rate is the largest in the vicinity of the boundary, it is important to be



(a) Kulasegaram et al.'s *gradient* operator (2.17) is unable to maintain uniform pressure field. Particles move to an non-physical equilibrium.

(b) The present *gradient* operator (2.23) is able to almost maintain uniform pressure field as an equilibrium.

Figure 2.3: Comparison of gradient operators (2.17) and (2.23) in an over-pressurized periodic pipe.

accurate in this area. This formula needs to be corrected with respect to the boundaries, given that it tends to underestimate the strain rate next to a wall. To correct that we propose in a similar way as (2.23):

$$\mathbf{Grad}_a^{\gamma^-}\{\boldsymbol{u}_b\} = -\frac{1}{\gamma_a \rho_a}\sum_{b\in\mathcal{P}} M_b \boldsymbol{u}_{ab}\otimes\boldsymbol{\nabla} w_{ab} + \frac{1}{\gamma_a\rho_a}\sum_{s\in\mathcal{S}}\rho_s \boldsymbol{u}_{as}\otimes\boldsymbol{\nabla}\gamma_{as}. \tag{2.24}$$

Hence, with the corrected components the strain rate S is computed according to the definition given by (2.24).

Laplacian:

The Laplacian (2.7) resulting from the Morris model (2.5) can be seen as a discretisation of the continuous interpolation $\langle \boldsymbol{\nabla}\cdot\mu\boldsymbol{\nabla} f\rangle(\boldsymbol{x})$ (where f is the continuous scalar field and μ the continuous viscosity field):

$$\begin{aligned}\langle \Delta(\mu,\ f)\rangle(\boldsymbol{x}) := \langle \boldsymbol{\nabla}\cdot\mu\boldsymbol{\nabla} f\rangle(\boldsymbol{x}) &= \langle \boldsymbol{\nabla}_{r'}\cdot[(\mu\boldsymbol{\nabla} f)(\boldsymbol{x}) + (\mu\boldsymbol{\nabla} f)(\boldsymbol{x}')]\rangle(\boldsymbol{x}) \\ &= -\frac{1}{\gamma(\boldsymbol{x})}\int_{\Omega\cap\Omega_r}[(\mu\boldsymbol{\nabla} f)(\boldsymbol{x}) + (\mu\boldsymbol{\nabla} f)(\boldsymbol{x}')]\,\boldsymbol{\nabla} w(\boldsymbol{x}-\boldsymbol{x}')\,\mathrm{d}V' \\ &\quad -\frac{1}{\gamma(\boldsymbol{x})}\int_{\partial\Omega\cap\Omega_r}[(\mu\boldsymbol{\nabla} f)(\boldsymbol{x}) + (\mu\boldsymbol{\nabla} f)(\boldsymbol{x}')]\cdot\boldsymbol{n} w(\boldsymbol{x}-\boldsymbol{x}')\,\mathrm{d}S',\end{aligned} \tag{2.25}$$

combined with the finite difference approximation:

$$\boldsymbol{\nabla} f(\boldsymbol{x})\cdot(\boldsymbol{x}-\boldsymbol{x}') \simeq (f(\boldsymbol{x}) - f(\boldsymbol{x}')) \simeq -\boldsymbol{\nabla} f(\boldsymbol{r}')\cdot(\boldsymbol{x}'-\boldsymbol{x}). \tag{2.26}$$

As in § 2.3.1, the boundary terms appear naturally in the second line of (2.25) from an integration by parts. Again, $\boldsymbol{n}$ is the inward normal of the domain at the position $\boldsymbol{x}'$. Wall contributions vanish when the position $\boldsymbol{x}$ is far from the boundary since the kernel w has a compact support.

Thus, the proposed wall-corrected *Laplacian* operator is:

$$\begin{aligned}\frac{1}{\rho_a}\mathrm{Lap}_a^{\gamma}\left(\{B_b\},\{A_b\}\right) = \frac{1}{\gamma_a}\sum_{b\in\mathcal{P}} M_b \frac{B_a+B_b}{\rho_a\rho_b}\frac{A_{ab}}{r_{ab}^2}\boldsymbol{x}_{ab}\cdot\boldsymbol{\nabla}w_{ab} \\ -\frac{1}{\gamma_a\rho_a}\sum_{s\in\mathcal{S}}\left(B_s\boldsymbol{\nabla}A_s + B_a\boldsymbol{\nabla}A_a\right)\cdot\boldsymbol{\nabla}\gamma_{as}.\end{aligned} \tag{2.27}$$

The same calculus still holds with an arbitrary vector field $\{\boldsymbol{A}_b\}$ and is used to correct the diffusion term (e.g. (2.5)) in the Navier–Stokes equation.

2.3.2 Wall corrected operators in the discrete Navier–Stokes equations

Dynamic boundary conditions on the pressure:

The *gradient* operator (2.23) applied to the pressure field gives:

$$\mathbf{Grad}_a^{\gamma^+}\{p_b\} \;:=\; \frac{\rho_a}{\gamma_a}\sum_{b\in\mathcal{P}} M_b\left(\frac{p_a}{\rho_a^2}+\frac{p_b}{\rho_b^2}\right)\boldsymbol{\nabla}w_{ab} - \frac{\rho_a}{\gamma_a}\sum_{s\in\mathcal{S}}\left(\frac{p_a}{\rho_a^2}+\frac{p_s}{\rho_s^2}\right)\rho_s\boldsymbol{\nabla}\gamma_{as}. \tag{2.28}$$

Hence, we are left to compute the pressure p_s and the density ρ_s on the wall nodes defined in Figure 2.2. A robust way but first order dynamic condition is $\frac{\partial\rho}{\partial n}=0$. This condition is consistent with the renormalisation done in equation (2.13). In the presence of gravity and motion the analogous condition on the pressure field is written as:

$$\frac{\partial}{\partial n}\left(\frac{p^\star}{\rho}+\frac{u^2}{2}\right) \;=\; 0, \tag{2.29}$$

where $p^\star := p-\rho\boldsymbol{g}\cdot\boldsymbol{x}$ and u is the magnitude of the velocity.

To compute the pressure and the density at the wall an SPH interpolation can be used for the vertex particles in $\mathcal{V}$. The choice made is to average in space to give a wall value:

$$\begin{aligned}\rho_v &= \frac{1}{\alpha_v}\sum_{b\in\mathcal{F}} V_b\rho_b w_{bv}, \\ \frac{p_v}{\rho_v} &= \frac{1}{\alpha_v}\sum_{b\in\mathcal{F}} V_b\left(\frac{p_b}{\rho_b}-\boldsymbol{g}\cdot\boldsymbol{x}_{bv}+\frac{u_b^2-u_v^2}{2}\right)w_{bv},\end{aligned} \tag{2.30}$$

where the set $\mathcal{F}$ denotes all fluid particles $\mathcal{P}$, excluding particles in $\mathcal{V}$ and where α_v is defined by:

$$\alpha_v := \sum_{b\in\mathcal{F}} V_b w_{bv}. \tag{2.31}$$

A description of the Shepard filter α_v is given in § 2.3.4 (see Equation 2.47)).

Here, it is important to note that the interpolation done is based on fluid particles in $\mathcal{P}$ which do not belong to the set of vertex particles $\mathcal{V}$. For simplicity, we remain in 2-D and define the density and the pressure at the wall elements s to be:

$$\begin{aligned}\rho_s &= \frac{\rho_{v1}+\rho_{v2}}{2}, \\ \frac{p_s}{\rho_s} &= \frac{p_{v1}/\rho_{v1}+p_{v2}/\rho_{v2}}{2},\end{aligned} \tag{2.32}$$

where the wall nodes (or vertex particles) $v1$ and $v2$ are defined in § 2.3.1. This strategy to evaluate quantities at the wall will be reused to compute the wall shear stress in § 2.3.2 and also the wall value of a scalar transported by the flow in Appendix 2.A. There is no theoretical restriction to the spacing of the wall nodes.

Wall shear stress:

The wall-corrected formula (2.27) applied to the field $\boldsymbol{u}$ gives:

$$\frac{1}{\rho_a}\mathbf{Lap}_a^{\gamma}\left(\{\mu_b\},\{\boldsymbol{u}_b\}\right)=\frac{1}{\gamma_a}\sum_{b\in\mathcal{P}}M_b\frac{\mu_a+\mu_b}{\rho_a\rho_b}\frac{\boldsymbol{u}_{ab}}{r_{ab}^2}\boldsymbol{x}_{ab}\cdot\boldsymbol{\nabla}w_{ab}-\frac{1}{\gamma_a\rho_a}\sum_{s\in\mathcal{S}}|\boldsymbol{\nabla}\gamma_{as}|\left(\mu_a\boldsymbol{\nabla}\boldsymbol{u}_a+\mu_s\boldsymbol{\nabla}\boldsymbol{u}_s\right)\cdot\boldsymbol{n}_s, \tag{2.33}$$

and should be used in place of $\mathbf{Lap}_a\left(\{.\},\{.\}\right)$ in (2.8-2.9).

The boundary terms are then treated using the friction velocity $\boldsymbol{u}_\tau$ defined by:

$$\mu\left.\frac{\partial\boldsymbol{u}}{\partial\boldsymbol{n}}\right|_{wall}:=\rho u_\tau\boldsymbol{u}_\tau, \tag{2.34}$$

which represents the shear stress at the wall. In the present article, by convention, $\boldsymbol{u}_\tau$ is chosen to have the same direction as the local velocity field. This then replaces $(\mu\boldsymbol{\nabla}\boldsymbol{u})_s\cdot\boldsymbol{n}_s$ in the boundary term of (2.33) as in classical finite element or finite volume Navier–Stokes codes:

$$(\mu\boldsymbol{\nabla}\boldsymbol{u})_a\cdot\boldsymbol{n}_s\simeq\rho u_{\tau s}\boldsymbol{u}_{\tau s}. \tag{2.35}$$

Computation of the friction velocity in a laminar case:

The friction velocity $\boldsymbol{u}_\tau$ is a quantity defined at the boundary. To compute it in a computational fluid dynamic (CFD) code, instead of using its definition, we usually take the advantage of knowing the physical behaviour of the velocity field in the vicinity of the boundary. For example, in a laminar test case, the velocity profile is expected to be linear close to the wall and then the following relationship between distance to the wall z and velocity along the wall $\boldsymbol{u}$ holds:

$$u_\tau\boldsymbol{u}_\tau=\lim_{z\to0}\frac{\nu\boldsymbol{u}}{z}. \tag{2.36}$$

The main advantage is that we do not need to estimate the derivative of the velocity field next to the wall, where it is difficult to compute. Another advantage is that we can extend the definition of the friction velocity in the area where particles interfere with the boundary, that is, when the kernel support intersects the walls. Hence, we define:

$$u_{\tau a}\boldsymbol{u}_{\tau a}=\frac{\nu\boldsymbol{u}_a}{z_a}, \tag{2.37}$$

where z_a is the distance to the wall for a particle a.

Eventually, to evaluate $\rho_s u_{\tau s}\boldsymbol{u}_{\tau s}$ we use again the continuity of stresses (2.35) to establish:

$$\rho_v u_{\tau v}\boldsymbol{u}_{\tau v}\quad=\quad\frac{1}{\alpha_v}\sum_{b\in\mathcal{F}}V_b\rho_b u_{\tau b}\boldsymbol{u}_{\tau b}w_{bv}, \tag{2.38}$$

and $\rho_s u_{\tau s}\boldsymbol{u}_{\tau s}$ is the average value between the vertex particles $v1$ and $v2$ defined in § 2.3.1. These formulae are similar to (2.32).

The same wall treatment is extended to a transport equation of a scalar such as k or ϵ in the $k-\epsilon$ turbulence model in the Appendix 2.A.

2.3.3 Conservation issues: time integration for the continuity equation

The original time integration scheme used in previous work Violeau (2012), Violeau and Issa (2007) was a simple first-order symplectic scheme where an implicit velocity value was used in

the updated values of the position and density:

$$\begin{cases} \boldsymbol{u}_a^{n+1} &= \boldsymbol{u}_a^n - \dfrac{\Delta t}{\rho_a^n}\mathbf{Grad}_a^{+,n}\left\{p_b^n\right\} + \boldsymbol{g}, \\ \boldsymbol{x}_a^{n+1} &= \boldsymbol{x}_a^n + \Delta t \boldsymbol{u}_a^{n+1}, \\ \boldsymbol{\rho}_a^{n+1} &= \boldsymbol{\rho}_a^n + \Delta t \sum\limits_{b\,\in\,\mathcal{P}} M_b \boldsymbol{\nabla}^n w_{ab} \cdot \boldsymbol{u}_{ab}^{n+1}, \end{cases} \tag{2.39}$$

where the superscript $(.)^n$ refers to the time step n and to the time $t = \sum\limits_{i=1}^{n} \Delta t$.

In this semi-implicit scheme the velocities are explicit, whereas the positions are implicit. In the continuity equation, positions are explicit whereas the velocities are implicit; for this reason we do not write the right-hand side of Equation (2.39) as $\rho_a \mathrm{Div}_a^- \{\boldsymbol{u}_b\}$.

2.3.4 Improving the time integration of the continuity equation

To adapt the previous time integration scheme to the method of Kulasegaram et al. (2004) and the present modified one, the following scheme is possible:

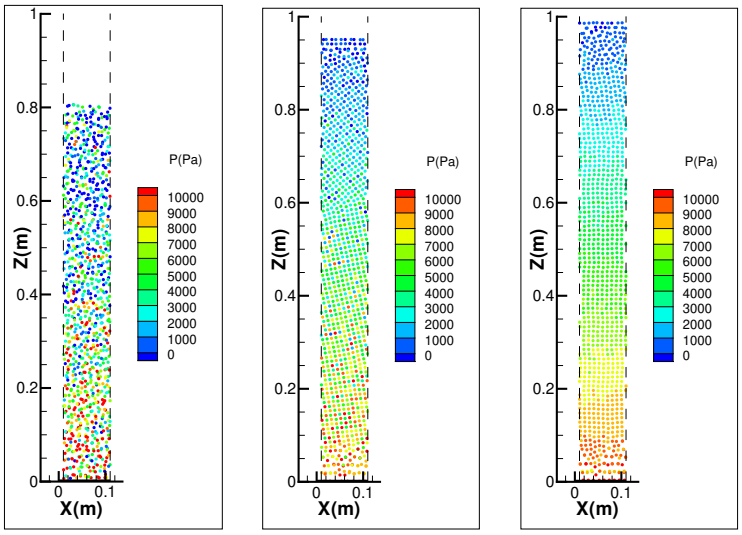
$$\begin{cases} \boldsymbol{u}_a^{n+1} &= \boldsymbol{u}_a^n - \dfrac{\Delta t}{\rho_a^n}\mathbf{Grad}_a^{\gamma^+,n}\left\{p_b^n\right\} + \boldsymbol{g}, \\ \boldsymbol{x}_a^{n+1} &= \boldsymbol{x}_a^n + \Delta t \boldsymbol{u}_a^{n+1}, \\ \rho_a^{n+1} &= \rho_a^n + \dfrac{\Delta t}{\gamma_a^n}\left[\sum\limits_{b\,\in\,\mathcal{P}} M_b \boldsymbol{\nabla}^n w_{ab}.\boldsymbol{u}_{ab}^{n+1} - \rho_a^n \boldsymbol{\nabla}^n \gamma_a \cdot \boldsymbol{u}_a^{n+1}\right], \end{cases} \tag{2.40}$$

where the operator $\mathbf{Grad}_a^{\gamma^+}\{.\}$ is either (2.17) or (2.23). Experience of the authors has shown this approach seems to give satisfactory results for a dam break case in a basin where the solid walls are kept impermeable with a relatively small time step. However, when running long-time simulations in a channel with a relatively large time step, particles near the wall move vertically downwards slowly and eventually pass through the boundary as we see in Figure 2.4(a) (a relatively big time step is set by decreasing the numerical speed of sound c_0) .

The problem is caused by the continuity equation: when particles near the boundary are oscillating, i.e. moving back and forth, their densities decrease, and then the pressure, related to the density by the equation of state, becomes insufficient to create a repulsive force to balance the other forces.

The origin of this phenomenon is the term $\frac{\Delta t}{\gamma_a^n}\rho_a^n \boldsymbol{\nabla}^n \gamma_a \cdot \boldsymbol{u}_a^{n+1}$ in the time discretised continuity equation (2.40). Indeed, if we consider a single particle moving towards the wall between the times t_n and t_{n+1} (from the distance z_n to z_{n+1}), the exact variation of the density is not reproduced by the discrete form Ferrand et al. (2010). This systematic error of the time discretisation of the continuity equation is undesirable .

Many other integration time schemes can be considered, such as a leap-frog time scheme, which would reduce the errors in the integration of the continuity equation. The choice made in the present work is to consider an approach that gives the density explicitly as a function of the particles' positions. This is a more robust approach, especially if there is a wall repulsive force function of the pressure field (and hence of the density field in compressible flows).



(a) Previous time scheme (2.40) with $c_0 = 20m.s^{-1}$.

(b) Previous time scheme (2.40) with $c_0 = 100m.s^{-1}$.

(c) New time scheme (2.42) with $c_0 = 20m.s^{-1}$.

Figure 2.4: Comparison of the pressure field and the water depth in a periodic open channel flowing from left to right for two different time schemes after the same physical time.

A completely position-dependent way to compute the density:

If we return to the main idea of correcting the incomplete kernel support, we can see that the corrected continuity equation comes from:

$$\begin{array}{ccc} \dfrac{\mathrm{d}\left(\gamma_a \rho_a\right)}{\mathrm{d}t} & = & \dfrac{\mathrm{d}}{\mathrm{d}t}\left(\displaystyle\sum_{b\,\in\,\mathcal{P}} M_b w_{ab}\right). \\ \| & & \| \\ \gamma_a \dfrac{\mathrm{d}\rho_a}{\mathrm{d}t} + \rho_a \dfrac{\mathrm{d}\gamma_a}{\mathrm{d}t} & = & \displaystyle\sum_{b\,\in\,\mathcal{P}} M_b \boldsymbol{\nabla} w_{ab} \cdot \boldsymbol{u}_{ab} \end{array} \tag{2.41}$$

In a similar way, Vila (1999) stated that the continuity equation (2.1) is strictly equivalent to $\dfrac{\mathrm{d}\rho_a}{\mathrm{d}t} = \dfrac{\mathrm{d}}{\mathrm{d}t}\left(\displaystyle\sum_{b\,\in\,\mathcal{P}} M_b w_{ab}\right)$ if time is considered a continuous variable. Equation (2.41) gives us a way to integrate exactly in time the quantity $\gamma_a \rho_a$ if the particles' positions move from $\left\{\boldsymbol{x}_b^n\right\}$ to $\left\{\boldsymbol{x}_b^{n+1}\right\}$. This prevents systematic time integration errors, and makes ρ_a^n depend only on the positions of particles at the same time (this property, together with symplectic time-stepping, ensures the conservation of a given energy, see Goldstein et al. (2002) for details).

This leads to the following time integration scheme:

$$\begin{cases} \boldsymbol{u}_a^{n+1} = \boldsymbol{u}_a^n - \dfrac{\Delta t}{\rho_a^n} \mathbf{Grad}_a^{\gamma^+,n}\left\{p_b^n\right\} + \boldsymbol{g}, \\ \boldsymbol{x}_a^{n+1} = \boldsymbol{x}_a^n + \Delta t \boldsymbol{u}_a^{n+1} \\ \left(\gamma_a \rho_a\right)^{n+1} = \left(\gamma_a \rho_a\right)^n + \displaystyle\sum_{b\,\in\,\mathcal{P}} M_b \left(w_{ab}^{n+1} - w_{ab}^n\right). \end{cases} \tag{2.42}$$

Figure 2.4 shows the result obtained by reducing the time-step (and hence the systematic error in the density equation) in the time scheme (2.40) with the result we obtain with the time-scheme (2.42) with a larger time step[3]. By doing so, scheme of (2.40) is forced to be stable and to converge but it is clear that we have lost density during the stabilisation time: the water depth has decreased (see 2.4(b)) whereas it is not the case on the Figure 2.4(c) with the new scheme.

Initialization of the density field:

The time-scheme (2.42) requires initial values for the density field. Many choices are possible. First of all, a particle a can have the reference density ρ_0 as initial value, i.e.

$$\rho_a^0 = \rho_0. \tag{2.43}$$

This was done previously with the continuity equation of (2.39). The advantage is that the continuity equation only measures the variation of density and not the initial disorder of the particles. However, the main drawback is an inhomogeneity between particles could lead to non-physical behaviour, such as particles originally at the free-surface creating a region of repulsion when surrounded by others later in the simulation. Hence, the initial density field, $\left\{\rho_a^0\right\}$, is initialized according to:

$$\rho_a^0 = \frac{1}{\gamma_a^0} \sum_{b \in \mathcal{P}^0} M_b w_{ab}^0. \tag{2.44}$$

This initialization has the advantage to maintain homogeneity between particles, but then requires a free-surface correction.

Free-surface correction:

The γ correction presented so far does not take into account any free-surface correction. The variable $\tilde{\rho}_a \coloneqq \sum_{b \in \mathcal{P}} M_b w_{ab}$ depends only on the particles positions which is useful for conservation properties. The problem is that $\tilde{\rho}_a$ measures two different quantities:

1. the divergence of the particles, which is a quantity of interest, and
2. the presence of voids within the kernel support of a particle.

It is required that the flow is corrected with γ_a next to a wall, but not next to the free-surface. To achieve this, we use the following Shepard filter defined by:

$$\alpha(\boldsymbol{x}) = \sum_{b \in \mathcal{P}} \frac{M_b}{\rho_b} w(\boldsymbol{x} - \boldsymbol{x}_b), \tag{2.45}$$

so that, for a fluid particle $a \in \mathcal{F}$, we have:

$$\alpha_a \coloneqq \sum_{b \in \mathcal{P}} \frac{M_b}{\rho_b} w_{ab}. \tag{2.46}$$

For a vertex particle $v \in \mathcal{V}$ and for the middle of a segment $s \in \mathcal{S}$ two definitions of α are used:

$$\alpha_v \coloneqq \sum_{b \in \mathcal{F}} \frac{M_b}{\rho_b} w_{vb}, \qquad \alpha_s \coloneqq \sum_{b \in \mathcal{F}} \frac{M_b}{\rho_b} w_{sb}, \tag{2.47}$$

[3]We have reduced the time step by setting the speed of sound at $100\,\mathrm{m\,s^{-1}}$ instead of $20\,\mathrm{m\,s^{-1}}$.

since α_v and α_s are used to evaluate quantities such as the density or the pressure at the wall. Hence we have chosen not to take into account vertex particles and have interpolation of the physical quantities (such as the density ρ or the pressure p) at the walls (see § 2.3.2) using only the fluid particles.

The aim is to apply the Shepard filter (2.46) on the density field at each time step but only next to the free-surface so that the density field is not corrected everywhere with α_a . Thus, in the vicinity of a free-surface there is a continuous mix to correct the continuity equation :

$$\rho_a \left[\beta\gamma_a + (1-\beta)\,\alpha_a\right] = \tilde{\rho}_a = \sum_{b\,\in\,\mathcal{P}} M_b w_{ab}, \tag{2.48}$$

where

$$\beta = \exp\left[-K\left(\min\left\{\frac{\alpha_a}{\gamma_a};\,1\right\} - 1\right)^2\right], \tag{2.49}$$

and K is taken to be an arbitrary high value of $-\dfrac{\ln(0.05)}{0.01^2} \simeq 30000$, so that $\beta \leq 0.05$ when $\frac{\alpha_a}{\gamma_a} \leq 0.99$. Note β is in fact a surface-marker: inside the fluid its value is almost one whereas it tends to zero as we approach the free-surface.

Note that this free-surface correction is used only in this chapter. This can be replaced by a volume diffusion correction presented in Chapter 3, Section 3.2.6.

2.3.5 Computation of the renormalisation terms

The formal definitions of the geometrical quantities γ_a and $\boldsymbol{\nabla}\gamma_a$ for a particle a are:

$$\begin{aligned} \gamma_a &:= \int_{\Omega\cap\Omega_a} w\left(\boldsymbol{x}-\boldsymbol{x}_a\right)\mathrm{d}\boldsymbol{x}, \\ \boldsymbol{\nabla}\gamma_a &:= \int_{\Omega\cap\Omega_a} \boldsymbol{\nabla}_a w\left(\boldsymbol{x}-\boldsymbol{x}_a\right)\mathrm{d}\boldsymbol{x} = \int_{\partial\Omega\cap\Omega_a} w\left(\boldsymbol{x}-\boldsymbol{x}_a\right)\boldsymbol{n}\mathrm{d}S. \end{aligned} \tag{2.50}$$

We recall that $\boldsymbol{\nabla}\gamma_a$ represents an approximation of the normal to the wall for a particle located at the position $\boldsymbol{x}_a$. Previous approaches proposed to compute it using polynomial approximation Kulasegaram et al. (2004); De Leffe et al. (2009), or an analytical solution Feldman and Bonet (2007), or again a discrete summation over boundary points Marongiu et al. (2007).

These approaches all had advantages and drawbacks discussed earlier. Herein, computation of the renormalisation term of the kernel support near a solid wall is obtained with a time integration scheme, thereby accounting more easily for any shape of boundaries presented in § 2.3.1.

Analytical value of $\boldsymbol{\nabla}\gamma_a$:

With the boundary of the domain composed of segments denoted with the subscript $(.)_s$, each segment has an inward normal $\boldsymbol{n}_s$, a beginning point $\boldsymbol{x}_{v1}$ and an ending point $\boldsymbol{x}_{v2}$ (see Figure 2.2). Then we can compute the analytical value of the contribution $\boldsymbol{\nabla}\gamma_{as}$ (defined by 2.18):

$$\boldsymbol{\nabla}\gamma_{as} := \left(\int_{\boldsymbol{x}_{v1}}^{\boldsymbol{x}_{v2}} w\left(r\right)\mathrm{d}l\right)\boldsymbol{n}_s.$$

which gives for the quintic Wendland kernel used in this work:

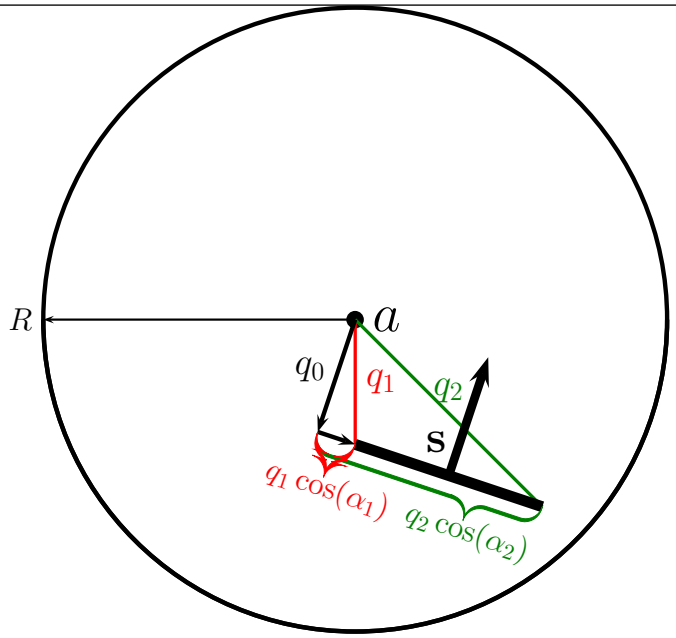


Figure 2.5: Schema showing the definitions of the geometrical parameters used to compute the analytical value of $\nabla\gamma_{as}$ with (2.51).

$$
\begin{aligned}
h\int_{\boldsymbol{x}_{v1}}^{\boldsymbol{x}_{v2}} w(r)\,\mathrm{d}l \;=\; & \frac{(q_2\cos\alpha_2)}{\pi}P_{q_0}(q_2)-\frac{(q_1\cos\alpha_1)}{\pi}P_{q_0}(q_1) \\
& +\frac{q_0^4}{\pi}\left(\frac{105}{64}+\frac{35}{512}q_0^2\right)\left[\begin{array}{l}\mathtt{sign}(q_2\cos\alpha_2)\ln\left(\dfrac{q_2+|q_2\cos\alpha_2|}{|q_0|}\right) \\ -\mathtt{sign}(q_1\cos\alpha_1)\ln\left(\dfrac{q_1+|q_1\cos\alpha_1|}{|q_0|}\right)\end{array}\right],
\end{aligned}
\tag{2.51}
$$

where the polynomial function P_{q_0} is defined by:

$$
\begin{aligned}
P_{q_0}(X) \;=\; & \frac{7}{192}X^5-\frac{21}{64}X^4+\frac{35}{32}X^3-\frac{35}{24}X^2+\frac{7}{4} \\
& +q_0^2\left(\frac{35}{768}X^3-\frac{7}{16}X^2+\frac{105}{64}X-\frac{35}{12}\right) \\
& +q_0^4\left(\frac{35}{512}X-\frac{7}{8}\right),
\end{aligned}
\tag{2.52}
$$

where $q_0 := \frac{|\boldsymbol{x}_{aei}\cdot\boldsymbol{n}_s|}{h}$, $q_i := \frac{|\boldsymbol{x}_{aei}|}{h}$ and $q_i\cos\alpha_i$ ($i\in\{1,2\}$) are displayed in Figure 2.5.

These analytical values enable estimates of the error due to the approximations. In Figure 2.6 we compare the analytical and approximate values of $\nabla\gamma_a$ against the distance to a plane wall. The discrete approximation used is given by:

$$
\nabla\gamma_{as} \simeq w_{as}S_s\boldsymbol{n}_s,
\tag{2.53}
$$

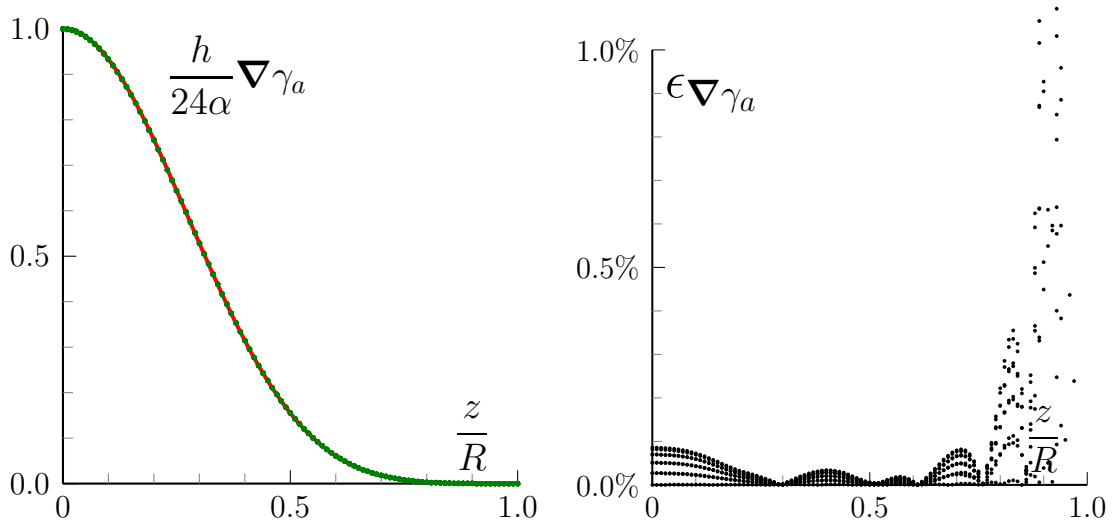
where it is assumed that the kernel is constant on a segment s.

The kernel used is the quintic kernel (see Wendland (1995)) and the ratio $\frac{\Delta r}{h}=2$ where Δr is the initial distance between two fluid particles and h is the smoothing length. The absolute errors

are defined as:

$$\epsilon_{\nabla\gamma_a} = \frac{\left|\nabla\gamma_a^{\text{analytical}} - \nabla\gamma_a^{\text{discrete}}\right|}{\nabla\gamma_a^{\text{analytical}}}, \tag{2.54}$$

(We consider only the component of $\nabla\gamma_a$ orthogonal to the wall). In Figure 2.6, the error for the discretisation of $\nabla\gamma_a$ is very good for such a poor discretisation ratio $\frac{\Delta r}{h}$ for a plane wall (less than 0.1%) whereas the discretisation has a systematic error for the approximation of the value of γ_a on the order of 3%.



(a) Values of the $\nabla\gamma_a$ function against the distance of the wall.

(b) Values of the error function $\epsilon_{\nabla\gamma_a}$ against the distance of the wall.

Figure 2.6: Analytical in red and computed values in green of the functions against the distance to a plan wall.

The error is larger when considering the discretisation error of $\nabla\gamma_a$ in the presence of a complex boundary such as a wedge. Moreover, the error is systematic, so that the magnitude of $\nabla\gamma_a$ is always underestimated. Within a simulation, this leads to non-physical behaviour, i.e. particles slide towards the wedge boundary since the gravity is not balanced completely by the wall repulsive force proportional to $\nabla\gamma_a$.

Governing equation for γ_a:

Herein, a method to compute γ_a for a particle a near a solid boundary is suggested without the need for fictitious particles and is also simpler than an analytic computation. The main idea of the present method is to use a governing equation of γ_a:

$$\begin{cases} \dfrac{\mathrm{d}\gamma_a}{\mathrm{d}t} = \nabla\gamma_a \cdot \boldsymbol{u}_a \quad , \\ \gamma_a = 1 \qquad\qquad \text{if } \partial\Omega \cap \Omega_a = \emptyset, \end{cases} \tag{2.55}$$

where the definition of the *gradient* is combined with the fact that $\frac{\mathrm{d}\boldsymbol{x}_a}{\mathrm{d}t} = \boldsymbol{u}_a$. Another way to consider this equation is to remark that (2.55) is equivalent to:

$$\left\{ \begin{array}{rcl} \dfrac{\partial \gamma_a}{\partial t} & = & 0, \\ \gamma_a & = & 1 \quad \text{if } \partial\Omega \cap \Omega_a = \emptyset. \end{array} \right. \tag{2.56}$$

This means that the γ_a field does not depend on the time[4], but only on position, and is therefore an Eulerian field. Thus, it gives us a means to compute γ_a to be consistent with $\boldsymbol{\nabla}\gamma_a$ which is easier to compute, since it can be expressed as a surface integral.

The relation (2.55) can be extended for moving boundaries. Recalling the boundary defined by Figure 2.2, if we have a moving deformable wall (in the sense that each segment or triangle composing the wall is moving with its velocity) the following formula is obtained:

$$\left\{ \begin{array}{rcll} \dfrac{\mathrm{d}\gamma_a}{\mathrm{d}t} & = & \displaystyle\sum_{s \in \mathcal{S}} \boldsymbol{\nabla}\gamma_{as} \cdot \boldsymbol{u}_a^{\mathcal{R}_s} & \\ \gamma_a & = & 1 & \text{if } \partial\Omega \cap \Omega_a = \emptyset, \end{array} \right. \tag{2.57}$$

where $\boldsymbol{u}_a^{\mathcal{R}_s}$ is the velocity of the particle a in a reference frame $\mathcal{R}_s$ where the segment s is fixed. The field γ_a is now computed by solving the above equation, based on the knowledge of $\boldsymbol{\nabla}\gamma_{as}$, computed from (2.51).

Initialisation of the γ_a field:

The initialization step of γ_a^0 is done by imagining a virtual transformation. For each particle initially next to a solid wall (a criterion is $\left|\boldsymbol{\nabla}\gamma_a^0\right| > 0$), we move it from its starting position $\boldsymbol{x}_a^0$ to an area where the function $\gamma(\boldsymbol{x}) := 1$. For instance:

$$\boldsymbol{x}_a = \boldsymbol{x}_a^0 + l\frac{\boldsymbol{\nabla}\gamma_a^0}{\left|\boldsymbol{\nabla}\gamma_a^0\right|}, \tag{2.58}$$

where the length l is taken to be $2R$ (R is the radius of the compact kernel support). Note that the factor 2 is to ensure that the particle is far enough from a zone where $\gamma_a < 1$ (for a square angle the factor which ensures that is $\sqrt{2}$).

A sketch of the proposed method is displayed in Figure 2.7. For example, the circled particle in Figure 2.7 is placed behind the dashed line where the value of γ is 1, and is moved back using many sub-steps (here 200) to its initial position along the path of the large black arrow updating the value of γ_a with respect to the governing Equation (2.57).

Note that the equation of γ_a is integrated in time with a second-order time integration scheme to prevent systematic integration errors (see § 2.3.4), leading to:

$$\left\{ \gamma_a^{n+1} = \gamma_a^n + \frac{1}{2}\left(\boldsymbol{\nabla}^n\gamma_a + \boldsymbol{\nabla}^{n+1}\gamma_a\right) \cdot \left(\boldsymbol{x}_a^{n+1} - \boldsymbol{x}_a^n\right), \right. \tag{2.59}$$

if the solid boundary is motionless. The general formula for a moving deformable wall is:

$$\left\{ \gamma_a^{n+1} = \gamma_a^n + \frac{\Delta t}{2}\sum_{s \in \mathcal{S}}\left(\boldsymbol{\nabla}^n\gamma_{as} + \boldsymbol{\nabla}^{n+1}\gamma_{as}\right) \cdot \left(\boldsymbol{u}_a^{\mathcal{R}_s}\right)^{n+1}. \right. \tag{2.60}$$

[4] if the wall does not move.

A condition on the time step is required, to keep the integration of γ_a stable:

$$\Delta t \leq C_{t,\gamma} \frac{1}{\max\limits_{a \in \mathcal{P}; s \in \mathcal{S}} \left\{ \left| \nabla^n \gamma_{as} \cdot \left(\boldsymbol{u}_a^{\mathcal{R}_s} \right)^n \right| \right\}}, \tag{2.61}$$

where $C_{t,\gamma} = 0.005$, from numerical experience. This is a natural condition, in the sense that the time step decreases when particles have a fast velocity when approaching a boundary such as splashing against a wall. The latter condition is considered in addition to the usual time-stepping (CFL) conditions in SPH (see e.g. Monaghan (1994)).

2.4 Numerical results

2.4.1 Laminar channel flow test case

To test the friction terms and validate the wall shear stress formulation, laminar Poiseuille flow in a closed-channel with periodic open boundaries is simulated. The channel has a diameter of $1m$ while the viscosity ν is set at 10^{-1} m^2 s^{-1} so that the Reynolds number is 10. The viscous term is modelled with the wall-corrected model of Morris (2.33) combined with (2.37) to compute the friction velocity. Figure 2.8 shows that the horizontal velocity profile is in good agreement with the analytical solution (2.62) even in the vicinity of the wall, thus demonstrating that the wall shear stress correctly balances the body force.

Computation of the strain:

The laminar case allows us to validate formula (2.24), since there is no influence of the strain rate on the flow itself because the viscosity is constant. Thus, such a test case allows us to compare the value of the computed strain rate to its analytical value. The analytical profile of velocity is:

$$u_x(z) = 4\,Re \frac{\nu\, z}{D^2} \left(1 - \frac{z}{D}\right), \tag{2.62}$$

which leads to the following analytical value of S:

$$S(z) = 4\,Re \frac{\nu}{D^2} \left| 2\frac{z}{D} - 1 \right|. \tag{2.63}$$

We notice in Figure 2.9 that in this theoretical test case, the corrected method (2.24) gives a satisfactory reproduction of the shear stress next to the wall unlike the previous methods.

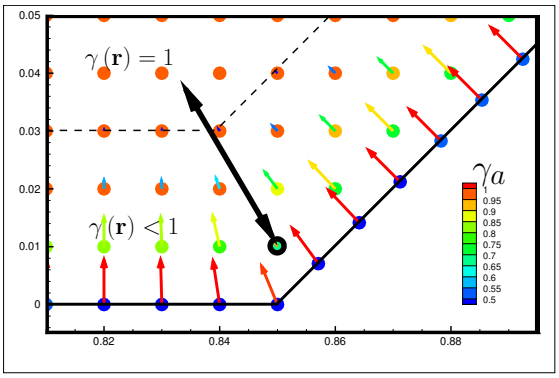


Figure 2.7: Sketch of the initialization of the γ field next to a solid wall.

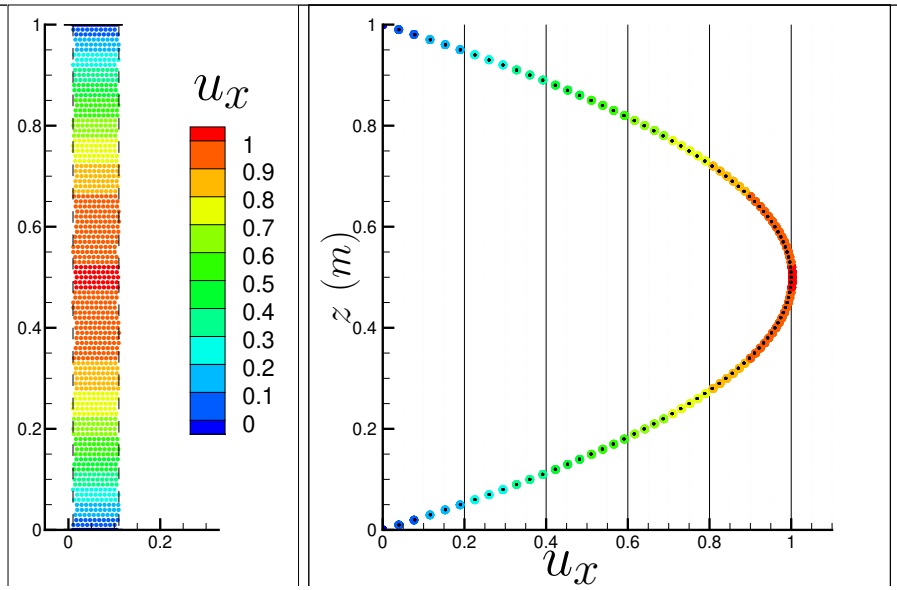


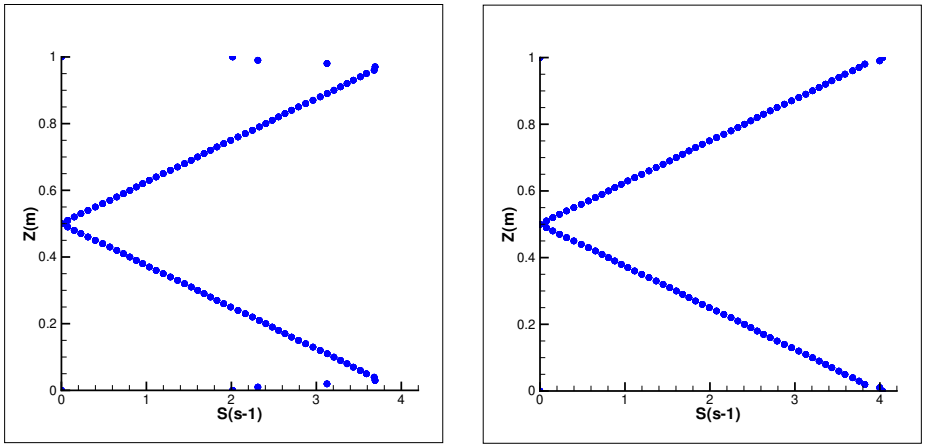
Figure 2.8: Poiseuille flow in a periodic pipe with a Reynolds number of 10. Coloured dots represent the velocity of particles at the steady state whereas the black dots • are the analytical profile (2.62).

2.4.2 Still water and dam break in a tank with a wedge

In this Section, the proposed scheme is tested on a more complex geometry which consists of a basin of approximately $2m$ length and $1m$ height with a wedge of $\frac{\pi}{2}\ rad$ angle and $\frac{\sqrt{2}}{8}$m of height in the bottom middle of the tank. This geometry was chosen specifically since the wedge features both a discontinuous point and a sloping profile which tests the correction of the kernel in the presence of gravity. A comparison is made between different models in a still water case and a dynamic case where the viscosity ν is set at 10^{-2} m^2 s^{-1}.

Still water case:

Some treatments for solid boundaries suffer from an inability to reproduce correctly a still water case. Here we compare the results obtained when the basin is filled with $0.5m$ of water for three



(a) Standard model (2.12) (b) Correction of Standard model (2.24)

Figure 2.9: Comparison of the strain rate for different models in a laminar channel flow.

cases: the Lennard-Jones repulsive forces (see Monaghan (1994)), the fictitious particles method (see Violeau and Issa (2007)) and then the new proposed method. As expected, the repulsive forces produce poor results (see Figures 2.10(b)) in the sense that particles keep sliding along vertical wall. That is due to the fact that the missing area in the kernel support is not compensated, and thus the gravity is not balanced sufficiently. The plot of the pressure of particles against the depth is therefore noisy and badly reproduced next to the bottom. The fictitious particles method (see Figure 2.10(a)) gives better results, but the condition (2.29) is not ensured and so the pressure profile is still noisy, even if that conclusion could be improved with recent developments (see e.g. Fourtakas et al., 2019). Moreover this approach is problematic to describe in complex geometries and requires additional particles to mimic the boundary.The present method gives superior results: a linear pressure profile even near the bottom and a zero velocity field as shown in Figure 2.10(c).

Dynamic case:

A simulation of a dam break with the same geometry has been performed for the two boundary techniques previously described and the present one. The water is initially a column of $1m$ height and $0.5m$ width on the left-hand side of the basin. In the results shown in Figure 2.11, all approaches ensure impermeable boundaries, but both repulsive forces method and fictitious particles method give a noisier pressure field. Furthermore, a simulation with finer resolution has been computed by doubling resolution and hence the number of particles. Snapshots of the pressure field at the same physical time are plotted on the Figure 2.11.

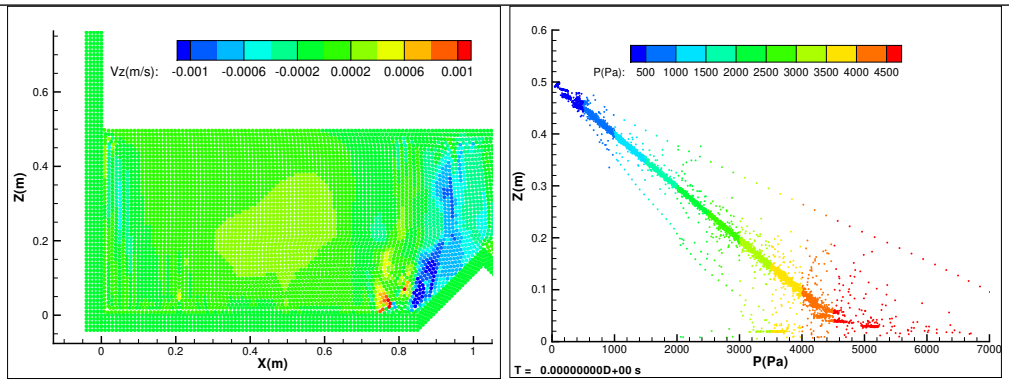
The comparison with a VOF simulation (done with finite volume open-source code code_saturne with similar refinements) of the pressure on the left side of the wedge is performed and displayed on Figure 2.12. The two methods give similar pressure evolution, even if SPH does not take air into account.

2.4.3 Comparison of the $k-\epsilon$ model with SPH and finite volumes: simulation of a fish pass

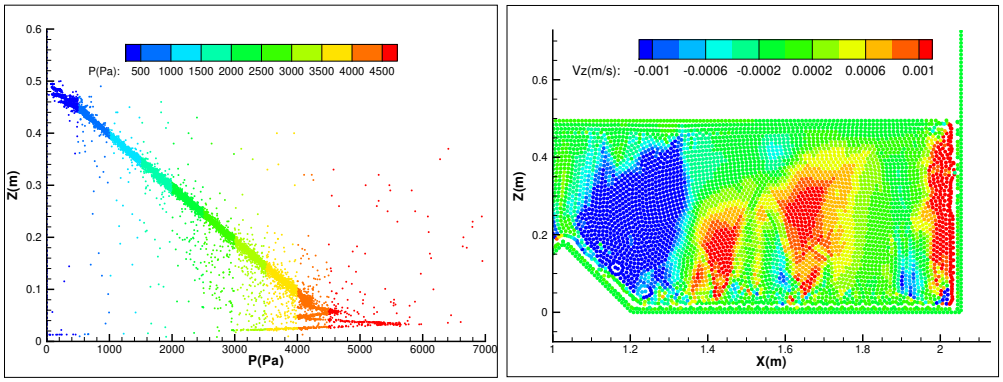
Once a dam is built on a river, the continuity of the flow is disrupted and the migration of fish is interrupted. For species such as some salmons, the life cycle requires the fishes to migrate far upriver. To restore the migration process, fish passes are installed which consist of many repeating elements and can be considered as a periodic flow. The dimensioning of these components requires a detailed knowledge of the turbulent flow within each section since the size of the large eddies affects the ability of the fish to swim upstream.

Although the behaviour of the flow though a fish pass is a 3-D free-surface flow, we consider here 2-D simulations where the vertical variations are assumed to be negligible. We repeat here the simulations presented in Violeau et al. (2008). The geometry of the $x-$periodic simulation is presented in Figure 2.13(a). The results obtained by SPH are compared to simulations done with code_saturne, a widely validated finite volume code developed by EDF R&D (see Archambeau et al. (2004) or Team (2021)). The aim of such a comparison is to evaluate the performance of SPH for a turbulent simulation. The computation is compared effectively solving the same equations (the Reynolds-averaged Navier–Stokes with the $k-\epsilon$ model) with the same pressure gradient driving the flow ($\frac{\Delta p}{\rho \Delta x} = 1.885\,\mathrm{m\,s^{-2}}$) but with two different approaches (Lagrangian and Eulerian) with two different discretisation approaches (SPH and finite volume).

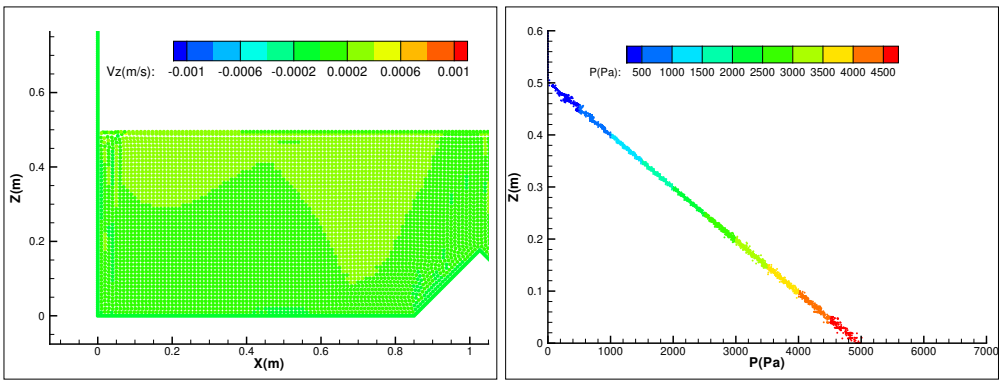
The velocity profiles at locations P_1, P_2 and P_3 defined in Figure 2.13(a) are plotted in Figure 2.14. The results show that the mesh-based code gives profiles close to the ones obtained with the present SPH scheme. It should be noted that with the standard SPH method Violeau et al. (2008), the predicted velocity and eddy viscosity distributions did not fit the finite volume ones. With the present model, one can see that the agreement is very satisfactory.



(a) Fictitious particles.



(b) Lennard-Jones type repulsive force.



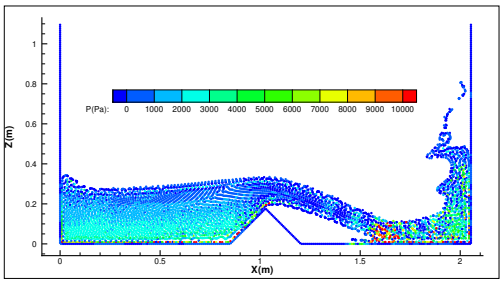
(c) Present method.

Figure 2.10: Comparison of the vertical velocity and particles pressure against water depth for still water in a tank with a wedge for different boundary conditions after 20 s.

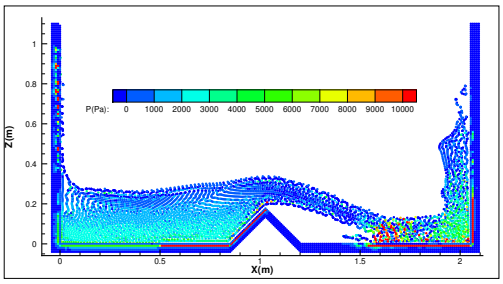
2.5 Conclusion

The present article has presented a new approach to deal with solid boundary condition which is both simple and robust. The simplicity lies in the manner we compute the Eulerian renormalisation term γ_a with an integration in time which only requires the computation of its gradient $\boldsymbol{\nabla}\gamma_a$. The robustness is due to the integration in time of the continuity equation which makes the density field depend only on the particles' positions. This allows a long time simulation with a relatively higher time step and is a major advantage for conservation properties.

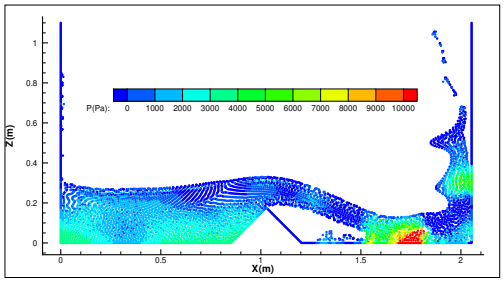
The definition of new boundary corrected *gradient* and Laplacian operators gives us the



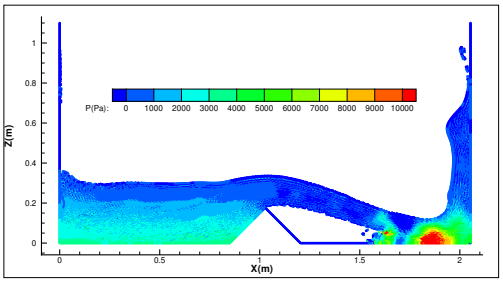
(a) Lennard and Jones repulsive forces.



(b) Fictitious particles.



(c) Present method.



(d) Present method with a twice smaller space discretisation.

Figure 2.11: Comparison of the pressure field for a dam break test case in a tank with a wedge for different boundary conditions. One can notice that the jet is more attached to the wall when refining.

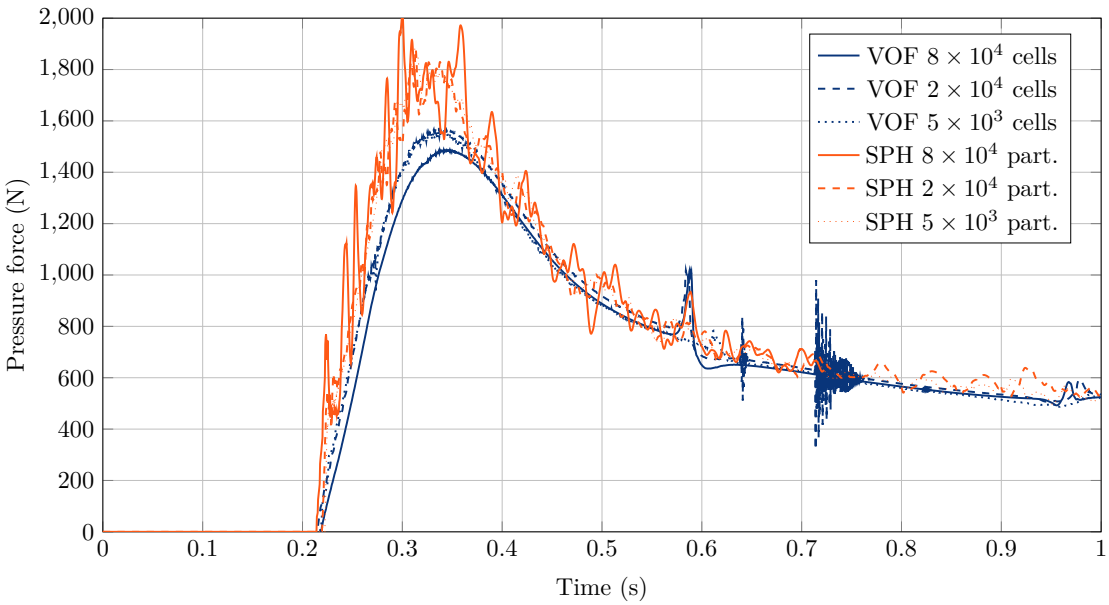
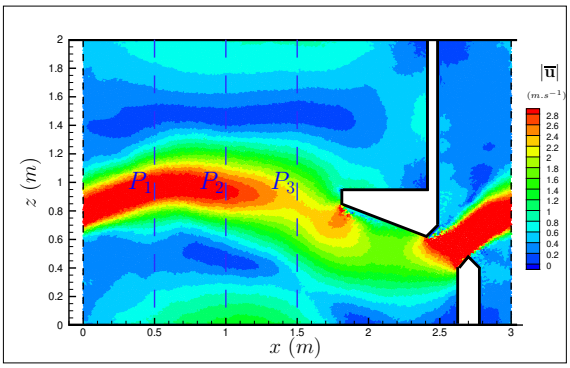
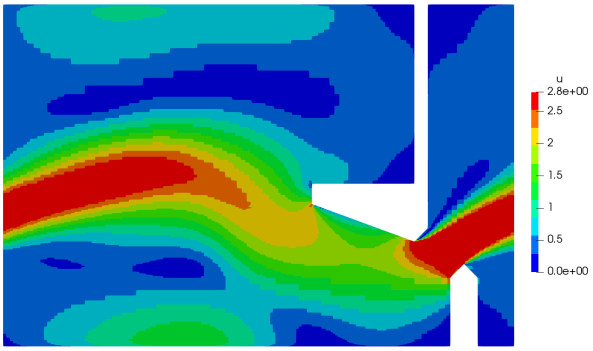


Figure 2.12: Comparison of the time evolution of the pressure along the left side of the wedge between the present SPH formulation and the finite volume VOF formulation at different spatial discretisations. One can notice that SPH pressure monitoring oscillates more than the VOFones, which can be due to the compressible model used.

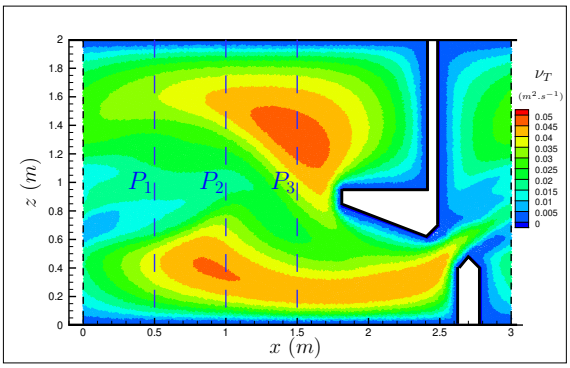
opportunity to fix boundary conditions and fluxes on the pressure field, the wall shear stress and



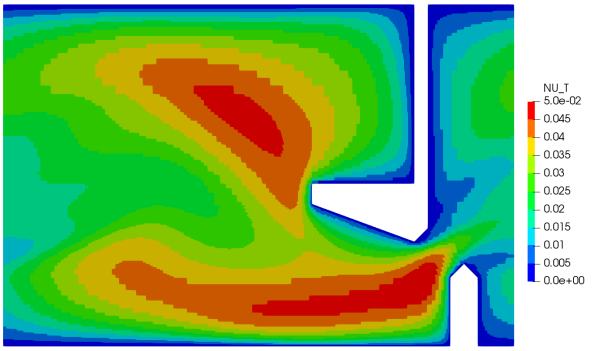
(a) Magnitude of the velocity field with *SPARTACUS-2D*.



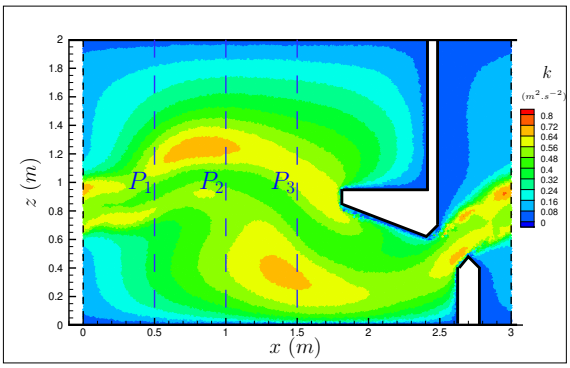
(b) Magnitude of the velocity field with code_saturne.



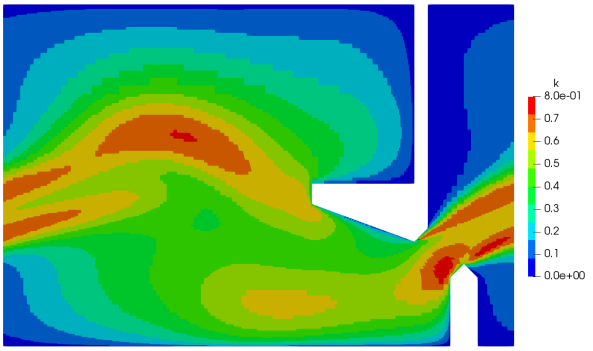
(c) Turbulent viscosity ν_T field with *SPARTACUS-2D*.



(d) Turbulent viscosity ν_T field with code_saturne.



(e) Kinetic energy k field with *SPARTACUS-2D*.



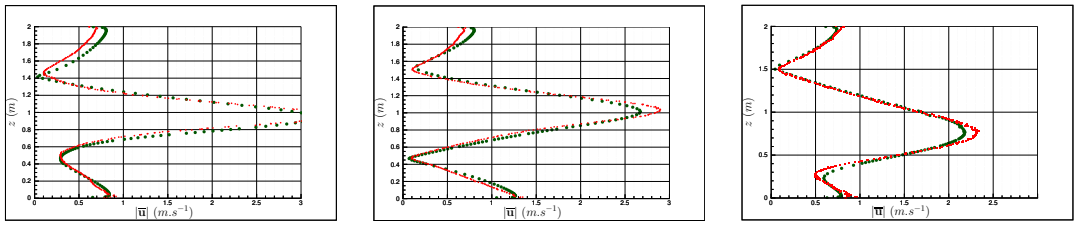
(f) Kinetic energy k field with code_saturne.

Figure 2.13: Comparison of the $k-\epsilon$ model (2.9) with the Lagrangian SPH approach and an Eulerian finite volume method in a schematic fish pass.

even the scalar fields such as k and ϵ in a model of turbulence.

However, numerous issues still require investigation and development, namely:

1. Validate the present formulation on different test cases such as a periodic turbulent bump.
2. Adapt the wall renormalisation to 3-D; the main challenge is to find an analytical formula for the computation of the contribution of a surface element s over the value of $\boldsymbol{\nabla}\gamma_a$ of a fluid particle a (which is denoted by $\boldsymbol{\nabla}\gamma_{as}$), or a way to compute accurately an approximated value of it.
3. Study the theoretical conservation (or non-conservation) of momentum and angular



(a) Profiles in P_1. (b) Profiles in P_2. (c) Profiles in P_3.

Figure 2.14: Profiles of the velocity magnitude in three different plans in the fish pass. The $k-\epsilon$ model (2.9) in red and the $k-\epsilon$ model with code_saturne in green.

momentum, and especially in periodic cases.

4. Combine the present approach with non-periodic entrance conditions. This point will be addressed in Chapter 3.

2.A Turbulent wall conditions in SPH

2.A.1 Computation of the friction velocity in a turbulent case

We can apply the same wall correction of the diffusion term of the momentum equation as the laminar one, noticing that $(\mu + \mu_T)\boldsymbol{S}\cdot\boldsymbol{n} \simeq \rho u_\tau \boldsymbol{u}_\tau$ in the vicinity of a wall[5]. Unlike the Equation (2.37) which links friction velocity to the mean velocity, in the turbulent case, we need another wall function. The viscosity is not constant anymore, and is supposed to be linear in the vicinity of a wall[6]. Then it can be shown that the velocity profile in that area is a logarithmic shape: this zone is called the log layer. If we consider the particle a to be in the log layer of a smooth wall, u_τ can be obtained from the following wall law, with an iterative algorithm:

$$\frac{|\boldsymbol{u}_a|}{u_{\tau a}} = \frac{1}{\kappa}\ln\left(\frac{z_a u_{\tau a}}{\nu}\right) + 5.2. \tag{2.64}$$

To be valid, Equation (2.64) must verify that the non-dimensional distance to the wall $\frac{z_a u_{\tau a}}{\nu}$ is greater than 11. We could also use log laws for rough walls, or use laws which hold both in the laminar and the log layer such as Reichards' law, but then the $k-\epsilon$ model (2.9) must also be modified for low Reynolds effects. For more information see Pope (2000).

2.A.2 Velocity at the wall

We observe in the formula (2.24) that the velocity at the wall $\boldsymbol{u}_s$ is considered in the boundary term. Formally, the no-slip condition would impose that the velocity at the wall is the velocity of the wall itself (i.e. 0 for a motionless wall). This is imposed for a laminar flow. In the turbulent case, it is preferable not to do so: the slope of the velocity profile is much larger at the wall than in the log layer where fluid particles are assumed to be. Thus if we want to evaluate accurately S_a next to the wall, we need to interpolate the velocity at the wall. To do so, we let the velocity

[5]Formally, at a wall $k = 0$ and then $\nu_T = 0$ so that we recover the laminar case. However, the viscous sub-layer where the laminar viscosity is more important than the turbulent one is usually, for environmental flows, very thin so that we do not enforce k to be zero at a wall.

[6]In a channel flow, this assumption is well verified in 10% of the channel depth.

$\boldsymbol{u}_v$ of vertex particles balance the viscous and friction terms:

$$\frac{\mathrm{d}\boldsymbol{u}_v}{\mathrm{d}t} = \underbrace{\frac{1}{\gamma_v}\sum_{b\,\in\,\mathcal{P}} M_b \frac{\mu_{Te}+\mu_{Tb}}{\rho_v\rho_b}\frac{\boldsymbol{u}_{vb}}{r_{vb}^2}\boldsymbol{x}_{vb}\cdot\boldsymbol{\nabla}w_{vb}}_{\text{viscous term}} - \underbrace{\frac{2u_{\tau v}\boldsymbol{u}_{\tau v}}{\gamma_v}\sum_{s\,\in\,\mathcal{S}}|\boldsymbol{\nabla}\gamma_{es}|}_{\text{friction term}}. \tag{2.65}$$

We notice here that Equation (2.65) is the momentum equation applied to a vertex particle with neither gravity nor pressure *gradient*. And we define $\boldsymbol{u}_s$ to be the average value between wall particles $v1$ and $v2$ defined in § 2.3.1. This tactic to allow a kind of "slip" velocity at the wall in a high Reynolds number simulation is also used in many CFD codes such as *TELEMAC-3D* in finite elements (see Hervouet (2007)). Eventually, we have to bear in mind that the vertex particles in $\mathcal{V}$ are in fact Eulerian points and do not move with the velocity $\boldsymbol{u}_v$ but with the wall velocity. That is to say we only use the velocity $\boldsymbol{u}_v$ to update viscous forces of fluid particles interacting with the wall, and to compute the strain rate S.

2.A.3 Flux conditions on the kinetic energy

The Laplacian operator (2.27) applied to the turbulent kinetic energy reduces to:

$$\frac{1}{\rho_a}\mathrm{Lap}_a^{\gamma}\left(\left\{\mu_b+\frac{\mu_{Tb}}{\sigma_k}\right\},\{k_b\}\right) = \frac{1}{\gamma_a}\sum_{b\,\in\,\mathcal{P}} M_b\frac{2\mu+\mu_{Ta}/\sigma_k+\mu_{Tb}/\sigma_k}{\rho_a\rho_b}\frac{k_{ab}}{r_{ab}^2}\boldsymbol{x}_{ab}\cdot\boldsymbol{\nabla}w_{ab}, \tag{2.66}$$

because it is assumed that there is no flux of k from the boundary, i.e. $\dfrac{\partial k}{\partial n}=0$ at the wall. Equation (2.66) should be used in place of $\mathrm{Lap}_a(\{.\},\{.\})$ in (2.9). The physical meaning is that the turbulent kinetic energy is only created by the mean flow. Moreover, instead of specifying boundary conditions **at** the wall, the "wall function" approach describes the balance of terms in the vicinity of the wall where it is assumed that $P=\epsilon$: this implies that the condition $\dfrac{\partial k}{\partial n}=0$ is valid not only at the wall, but in the whole vicinity of the solid boundary.

If we assume the flow to be highly turbulent, that is to say, with a thin viscous sub-layer, we can use high Reynolds number laws for $k-\epsilon$. In fact we do not solve the $k-\epsilon$ model up to the wall, where k is theoretically expected to be 0, but up to a small distance δ from the wall, where the turbulence is fully established ($\nu_T \gg \nu$). The main advantage of the present Lagrangian approach compared to an Eulerian one is that all the Lagrangian fluid particles in $\mathcal{F}$ will be at least at a distance of the order of Δr from any wall. It is one of the main advantages compared to the mesh-based methods where the boundary elements are supposed to be at a virtual distance to the actual wall, here we only focus on particles which are effectively at a non-zero distance from the walls.

To estimate k at the wall we can evaluate as we did for the density, the pressure or the wall shear stress:

$$k_v = \frac{1}{\alpha_v}\sum_{b\,\in\,\mathcal{F}} V_b k_b w_{bv}. \tag{2.67}$$

We notice that this approximation is consistent with the assumption $\dfrac{\partial k}{\partial n}=0$. Finally:

$$k_s = \frac{k_{v1}+k_{v2}}{2}. \tag{2.68}$$

2.A.4 Flux conditions on the dissipation of kinetic energy

The Laplacian operator (2.27) applied to ϵ requires the value for $\partial\epsilon/\partial n$. Once again, if we assume that the flow is highly turbulent, then every single particle in the area of influence of a wall (i.e. $\exists s \in \mathcal{S}/|\boldsymbol{\nabla}\gamma_{as}| > 0$) is in the log layer where[7]:

$$\begin{aligned} k &\simeq \frac{u^{\star 2}}{\sqrt{C_\mu}}, \\ \epsilon &= \frac{u^{\star 3}}{\kappa z}, \\ \nu_T &= \kappa u^\star z, \end{aligned} \tag{2.69}$$

where z is the distance to the wall (if a particle a is interacting with a surface s, we state $z = \max(\boldsymbol{x}_{as} \cdot \boldsymbol{n}_s; \Delta r)$), where Δr is the particle averaged initial spacing, κ is the Von Kármán constant with the value of 0.41, and $u^\star$ is a friction velocity measuring the turbulence:

$$u_s^\star = \frac{\sqrt{k_s}}{C_\mu^{\frac{1}{4}}}. \tag{2.70}$$

Then, from Equation (2.69) we can deduce a value for the flux of ϵ:

$$\frac{\nu_{Ta}}{\sigma_\epsilon}\frac{\partial \epsilon_a}{\partial n_s} = -\frac{2u_s^{\star 4}}{\sigma_\epsilon \kappa \delta r_{as}}. \tag{2.71}$$

The factor 2 is provided by a first-order approximation (i.e. the flux is evaluated at the distance $\frac{z}{2}$). This accuracy is needed especially here because ϵ is supposed to vary as $\frac{1}{z}$ where z is the distance to the wall.

Thus the Laplacian (2.27) becomes:

$$\begin{aligned} \frac{1}{\rho_a}\mathrm{Lap}_a^\gamma\left(\left\{\mu_b + \frac{\mu_{Tb}}{\sigma_\epsilon}\right\}, \{\epsilon_b\}\right) &= \frac{1}{\gamma_a}\sum_{b\in\mathcal{P}} M_b \frac{2\mu + \mu_{Ta}/\sigma_\epsilon + \mu_{Tb}/\sigma_\epsilon}{\rho_a\rho_b}\frac{\epsilon_{ab}}{r_{ab}^2}\boldsymbol{x}_{ab}.\boldsymbol{\nabla}w_{ab} \\ &\quad + \frac{4}{\gamma_a\rho_a}\sum_{s\in\mathcal{S}} |\boldsymbol{\nabla}\gamma_{as}|\,\rho_s \frac{u_s^{\star 4}}{\sigma_\epsilon\kappa\delta r_{as}}, \end{aligned} \tag{2.72}$$

which is used in place of $\mathrm{Lap}_a(\{.\}, \{.\})$ in (2.9).

Bibliography

Antuono, M., Colagrossi, A., Marrone, S., Molteni, D., 2010. Free-surface flows solved by means of sph schemes with numerical diffusive terms. Computer Physics Communications 181, 532–549.

Archambeau, F., Méchitoua, N., Sakiz, M., 2004. Code_saturne: a finite volume method for the computation of turbulent incompressible flows - industrial applications. International Journal on Finite Volumes 1, 1–62.

Chiron, L., de Leffe, M., Oger, G., Touzé, D.L., 2019. Fast and accurate SPH modelling of 3D complex wall boundaries in viscous and non viscous flows. Computer Physics Communications 234, 93–111. URL: http://www.sciencedirect.com/science/article/pii/S0010465518302832, doi:https://doi.org/10.1016/j.cpc.2018.08.001.

[7] All the Equations (2.69) can be derived from the equilibrium $P = \epsilon$.

De Leffe, M., Le Touzé, D., Alessandrini, B., 2009. Normal flux method at the boundary for SPH, in: 4^{th} SPHERIC workshop, Nantes, France. pp. 149 – 156.

Di Monaco, A., Manenti, S., Gallati, M., Sibilla, S., Agate, G., Guandalini, R., Maffio, A., 2009. A semi-analytic approach for SPH modelling of solid boundaries, in: 4^{th} SPHERIC workshop, Nantes, France. pp. 165 – 172.

Feldman, J., Bonet, J., 2007. Dynamic refinement and boundary contact forces in SPH with applications in fluid flow problems. Int. Journal for Numerical Methods in Engineering 72, 295 – 324. Meshfree Methods: Recent Advances and New Applications.

Ferrand, M., Laurence, D., Rogers, B.D., Violeau, D., 2010. Improved time scheme integration approach for dealing with semi-analytical wall boundary conditions in Spartacus2D, in: 5th International SPHERIC SPH Workshop.

Ferrand, M., Laurence, D.R., Rogers, B.D., Violeau, D., Kassiotis, C., 2013. Unified semi-analytical wall boundary conditions for inviscid, laminar or turbulent flows in the meshless SPH method. International Journal for Numerical Methods in Fluids 71, 446–472. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.3666, doi:10.1002/fld.3666, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.3666.

Fourtakas, G., Dominguez, J.M., Vacondio, R., Rogers, B.D., 2019. Local uniform stencil (lust) boundary condition for arbitrary 3-d boundaries in parallel smoothed particle hydrodynamics (sph) models. Computers & Fluids 190, 346–361.

Goldstein, H., Poole, C., Safko, J., 2002. Conservation of the Hamiltonian. Addison Wesley, San Francisco. $3^{rd} edition$.

Hervouet, J., 2007. Hydrodynamics of Free Surface Flows: Modelling with the Finite Element Method. John Wiley.

Kulasegaram, S., Bonet, J., Lewis, R.W., Profit, M., 2004. A variational formulation based contact algorithm for rigid boundaries in two-dimensional SPH applications. Computational Mechanics 33, 316–325.

Launder, B.E., Spalding, D.B., 1972. Mathematical Models of Turbulence. London: Academic Press .

Marongiu, J.C., Leboeuf, F., Caro, J., 2009. Low Mach number numerical schemes for the SPH-ALE method. Application in free surface flows in Pelton turbines, in: 4^{th} SPHERIC workshop, Nantes, France. pp. 323 – 330.

Marongiu, J.C., Leboeuf, F., Parkinson, E., 2007. Numerical simulation of the flow in a Pelton turbine using the meshless method SPH and a new simple solid boundary treatment, in: 7th European Conference on Turbomachinery Fluid Dynamics and Thermodynamics, Athens, Greece.

Monaghan, J., Kajtar, J., 2009a. SPH particle boundary forces for arbitrary boundaries. Computer Physics Communications 180, 1811 – 1820.

Monaghan, J.J., 1992. Smoothed Particle Hydrodynamics. Annual review of Astronomy and Astrophysics 30, 543–574.

Monaghan, J.J., 1994. Simulating free surface flows with SPH. Journal of Computational Physics 110, 399–406.

Monaghan, J.J., Kajtar, J.B., 2009b. Sph particle boundary forces for arbitrary boundaries. Computer physics communications 180, 1811–1820.

Morris, J.P., Fox, P.J., Zhu, Y., 1997. Modeling Low Reynolds Number Incompressible Flows Using SPH. Journal of Computational Physics 136, 214–226.

Oger, G., Doring, M., Alessandrini, B., Ferrant, P., 2007. An improved SPH method: Towards higher order convergence. Journal of Computational Physics 225, 1472 – 1492.

Pope, S.B., 2000. Turbulent Flows. Cambridge University Press.

Team, D., 2021. code_saturne 7.0 Theory Guide. EDF R&D. URL: https://code-saturne.org/cms/sites/default/files/docs/7.0/theory.pdf.

Vacondio, R., Altomare, C., De Leffe, M., Hu, X., Le Touzé, D., Lind, S., Marongiu, J.C., Marrone, S., Rogers, B.D., Souto-Iglesias, A., 2021. Grand challenges for smoothed particle hydrodynamics numerical schemes. Computational Particle Mechanics 8, 575–588.

Vila, J.P., 1999. On particle weighted methods and smooth particle hydrodynamics. Mathematical Models and Methods in Applied Sciences 9, 161–209.

Violeau, D., 2012. Fluid Mechanics and the SPH Method: Theory and Applications. Oxford University Press. URL: http://ukcatalogue.oup.com/product/academic/earthsciences/hydrology/9780199655526.do.

Violeau, D., Issa, R., 2007. Numerical modelling of complex turbulent free-surface flows with the SPH method: an overview. International Journal for Numerical Methods in Fluids 53, 277–304. URL: http://dx.doi.org/10.1002/fld.1292, doi:10.1002/fld.1292.

Violeau, D., Issa, R., Benhamadouche, S., Saleh, K., Chorda, J., Maubourguet, M.M., 2008. Modelling a fish passage with SPH and Eulerian codes: the influence of turbulent closure, in: Proc. III^{rd} SPHERIC International Workshop, Lausanne, Suisse. pp. 85–91.

Wendland, H., 1995. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. Advances in Computational Mathematics 4, 389–396.

3. Unsteady open boundaries for free-surface flows using semi-analytical conditions and Riemann solver in SPH

Abstract 3.1 Due to the Lagrangian nature of SPH, treating inlet/outlet boundaries (that are intrinsically Eulerian) is a challenging issue. An extension to the unified semi-analytical boundary conditions developed by Ferrand et al. (2013) is presented to deal with unsteady open boundaries in free surface flows. The presented method uses Riemann invariants to calculate fields of good quality near the open boundaries, thus allowing the possibility to treat complex shapes. Furthermore this method is implemented in a GPU framework, for which the specificities are presented. To finish, simple test cases are displayed to illustrate the performance of the proposed method as well as the ability to deal with complex problems such as generation of water waves and free-outlets. Appendix of Ferrand et al. (2013) has been corrected as already done in Fonty (2019).

Résumé 3.1 De par la nature souvent lagrangienne de la méthode *SPH*, le traitement des conditions aux limites de type entrée/sortie (ces limites sont intrinsèquement eulériennes) est un challenge. Une extension des conditions aux limites *unified semi-analytical* développées dans Ferrand et al. (2013) est présentée pour traiter les conditions aux limites de milieux ouverts instationnaires pour des écoulements à surface libre. La méthode utilise les invariants de Riemann pour imposer les variables sur les frontières ouvertes du domaine, permettant la prise en compte de frontières de forme complexe. De plus, cette méthode est implémentée sur carte graphique *GPU* et quelques spécificités de l'implémentation sont présentées. Enfin, des cas tests simples sont exposés afin d'illustrer les performances de la méthode ainsi que sa capacité à simuler la génération de vagues avec des sorties libres. L'annexe de Ferrand et al. (2013) a été corrigée, conformément à la correction déjà effectuée dans Fonty (2019) .

3.1 Introduction

The intrinsic computational cost of smoothed particle hydrodynamics (SPH) makes natural the interest in simulations where only part of a larger problem is treated. More generally, open boundaries are required in many fluid mechanics problems. For this kind of simulations, inlet/outlet conditions need to be developed. Efficient inlet/outlet boundary treatment is also required when solving coupled problems, especially when strong coupling algorithms are used, since the fields near the open boundaries are shared by the coupled models, and perturbations near the shared boundary can lead to instabilities Kassiotis et al. (2011).

The simplest way to treat inlet/outlet in SPH is to use a buffer layer, where the values of the fields at the boundary are imposed on several layers of particles ("buffer particles") that complete the kernel support of free particles close to the open boundary Vacondio et al. (2011); Bouscasse et al. (2013); Braun et al. (2015). In the inlet case, when entering the fluid domain a buffer particle is marked as fluid particle and is then free to move. This sudden modification can generate spurious shocks. Similarly, a fluid particle leaving the fluid domain through an outlet is first marked as a buffer particle, and some of its physical quantities are suddenly prescribed, generating shocks. In the framework of Weakly Compressible SPH (WCSPH), using Riemann solvers can partially solve this issue Mahmood et al. (2012); Lastiwka et al. (2009), but modelling a complex boundary where the flow can enter and/or leave the domain remains a difficult issue.

A slightly more complicated approach would be to use mirror particles at open boundaries. This work is just starting and has been successfully applied to Dirichlet pressure boundaries by Kunz et al. (2016).

The unified semi-analytical strategy proposed by Ferrand et al. (2013) has shown promising results to treat both Dirichlet and Neumann conditions for wall boundary treatment in SPH (see also Leroy et al. (2014)). In this approach, the boundaries are discretised using boundary elements hereafter referred to as 'segments', as well as 'vertex' particles at their intersections (see Figure 3.1). This treatment of the boundaries will be extended here to open boundaries. For this purpose, the main novelties are the following:

- Two additional terms in the SPH continuity equation will be introduced, naturally derived from the unified semi-analytical approach;
- The masses of vertex particles evolve over time according to the desired ingoing/outgoing mass flux at the connected segments;
- The vertex particles are used to create or delete fluid particles which are free to move according to the SPH momentum equation;
- The fields at the open boundaries are specified from the Riemann invariants of the Euler equations.

All these changes allow the formulation to treat particles entering and exiting the domain without perturbations of the fields for both confined and free surface flows.

The outline of this Chapter is as follows: first a short description of the unified semi-analytical boundary conditions for walls as proposed in Ferrand et al. (2013) is given in Section 3.2. These wall conditions will be extended to inlets and outlets (Section 3.3). It will be followed by a method prescribing the fields of the ingoing/outgoing particles from Riemann invariants. These developments will the be tested in Section 3.4, where the ability of the present method to treat wall-bounded and free-surface flows with complex open boundaries will be tested.

All of this work will use only the WCSPH approach, assuming the reader is familiar with it (for more details about this method, see Monaghan (2005)). Developing similar open boundaries in an Incompressible SPH formulation is possible Leroy (2014), however the equations in this

case are elliptic, which will change the treatment of fields at the open boundaries. The article will focus on treating the hyperbolic equations of the WCSPH approach, and the specificities in imposing compatible pressure and velocity fields on the open boundaries. Nonetheless, treating the mass fluxes should be equivalent in both approaches.

Finally, this approach will be restricted to 2-D one-fluid flows.

This work has already been continued and adapted to incompressible formulation by Leroy et al. (2016), and to two-phase flows with transport equation on air fraction by Fonty et al. (2020). An updated review of the progress in open boundaries for SPH can be found in Vacondio et al. (2021).

3.2 Weakly compressible SPH with semi-analytical boundary conditions

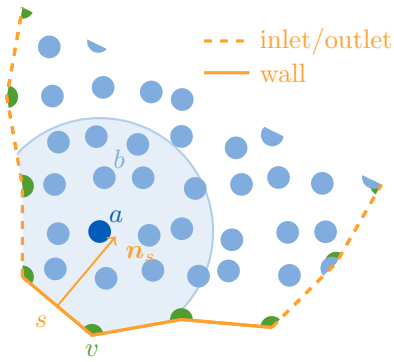


Figure 3.1: Space discretisation. Vertex particles are shown in green and segments in orange.

3.2.1 Space discretisation

As illustrated in Figure 3.1, the weakly compressible fluid domain Ω is discretised by a set of SPH fluid particles $\mathcal{F}$ denoted by the subscripts $(.)_a$ and $(.)_b$, while the boundaries $\partial\Omega$ are discretised by a set of vertex particles $\mathcal{V}$ denoted by $(.)_v$ and connected to boundary segments $\mathcal{S}$ denoted by $(.)_s$. Let $\mathcal{P}$ denote the particle set that is the union of the sets $\mathcal{F}$ and $\mathcal{V}$. Each element (particle or segment) stores information such as mass M_a, position centre $\boldsymbol{x}_a$, particle velocity $\boldsymbol{v}_a$ (i.e. the Lagrangian derivative of the position), fluid velocity $\boldsymbol{u}_a$ (i.e. the velocity field at position $\boldsymbol{x}_a$), density ρ_a, dynamic viscosity $\mu_a := \nu\rho_a$, pressure p_a and volume $V_a = M_a/\rho_a$.

It is important to underline that two sets of velocities are needed here, since the vertex particles and boundary segments at open boundaries are (most of the time) fixed in space but carry an information on the fluid velocity, namely $\boldsymbol{u}_v$ and $\boldsymbol{u}_s$. In other words, we have $\boldsymbol{u}_a = \boldsymbol{v}_a$ except for vertex particles and segments located onto the open boundaries. On the other hand, this equality holds for wall segments and vertex particles, thanks to the no-slip condition. Furthermore, in our model all particle masses M_a are constant in time, with the exception of the vertex particles of open boundaries, as detailed later.

Let ρ_0 be the reference density of the considered fluid. In the Weakly Compressible SPH approach used herein, the pressure field is deduced from the density field using Tait's equation of state Tait (1888), given as:

$$p_a = \frac{\rho_0 c_0^2}{\xi}\left[\left(\frac{\rho_a}{\rho_0}\right)^{\xi} - 1\right], \tag{3.1}$$

where c_0 is the numerical speed of sound and the Laplace coefficient, denoted by ξ instead of γ for the sake of clarity, is taken to 7 for water.

The SPH interpolation is based on a weighted interpolation from a kernel function w, generally compactly supported. Here we use the Wendland kernel of order 5 Wendland (1995). In our notation, Ω_a refers to the support of the kernel function centred on $\boldsymbol{x}_a$. The subscripts $(.)_{ab}$ generally denote the difference of a quantity between the particles a and b (unless stated otherwise). For instance $\boldsymbol{u}_{ab} \coloneqq \boldsymbol{u}_a - \boldsymbol{u}_b$ and $\boldsymbol{x}_{ab} \coloneqq \boldsymbol{x}_a - \boldsymbol{x}_b$. Some exceptions are made, including the following notations: $w_{ab} \coloneqq w(\boldsymbol{x}_{ab})$ and $\boldsymbol{\nabla} w_{ab} \coloneqq \boldsymbol{\nabla}_a w(\boldsymbol{x}_{ab})$. The symbol $\boldsymbol{\nabla}_a$ denotes the gradient with respect to the position $\boldsymbol{x}_a$.

3.2.2 Boundary renormalisation

Following several authors, in particular Kulasegaram et al. (2004) and Ferrand et al. (2013), an additional field denoted by γ_a is used to take boundary terms into account. This field measures the part of the kernel support which is inside the computational domain and is defined as:

$$\gamma_a \coloneqq \int_{\Omega_a \cap \Omega} w(\boldsymbol{x}_a - \boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}. \tag{3.2}$$

The field γ_a is computed from a dynamic governing equation (see Section 3.2.5). The discrete SPH operators presented in the next Section also require the computation of $\boldsymbol{\nabla}\gamma_a$, which is performed by a decomposition onto the boundary segments. Each segment contribution $\boldsymbol{\nabla}\gamma_{as}$ is defined as:

$$\boldsymbol{\nabla}\gamma_{as} \coloneqq \left(\int_s w(r) \, \mathrm{d}S \right) \boldsymbol{n}_s, \tag{3.3}$$

where $\boldsymbol{n}_s$ is the inward unit normal to the boundary segment s. Now $\boldsymbol{\nabla}\gamma_a$ is written as:

$$\boldsymbol{\nabla}\gamma_a = \sum_{s \in \mathcal{S}} \boldsymbol{\nabla}\gamma_{as}. \tag{3.4}$$

This gradient can be calculated analytically, see Ferrand et al. (2013).

3.2.3 Space differential corrected operators

In the present Section, space discretised differential operators defined by Ferrand et al. (2013) will be used. These operator include $\mathbf{Grad}_a^{\gamma^+}\{A_b\}$ which is a boundary corrected gradient of the discrete scalar field $\{A_b\}$, $\mathrm{Div}_a^{\gamma^-}\{\boldsymbol{A}_b\}$ which is a boundary corrected divergence of the discrete vector field $\{\boldsymbol{A}_b\}$, and $\mathrm{Lap}_a^{\gamma}(\{B_b\}, \{A_b\})$ which is a boundary corrected Laplacian of the discrete scalar (or vector) field $\{A_b\}$ with discrete diffusion field $\{B_b\}$. They are defined by the following equations:

$$\begin{aligned} \mathbf{Grad}_a^{\gamma^+}\{A_b\} \coloneqq & \frac{\rho_a}{\gamma_a} \sum_{b \in \mathcal{P}} M_b \left(\frac{A_a}{\rho_a^2} + \frac{A_b}{\rho_b^2} \right) \boldsymbol{\nabla} w_{ab} \\ & - \frac{\rho_a}{\gamma_a} \sum_{s \in \mathcal{S}} \left(\frac{A_a}{\rho_a^2} + \frac{A_s}{\rho_s^2} \right) \rho_s \boldsymbol{\nabla}\gamma_{as}, \end{aligned} \tag{3.5}$$

$$\begin{aligned}\mathrm{Div}_a^{\gamma^-}\{\boldsymbol{u}_b\} \coloneqq & -\frac{1}{\gamma_a\rho_a}\sum_{b\in\mathcal{P}} M_b \boldsymbol{u}_{ab}\cdot\boldsymbol{\nabla}w_{ab} \\ & +\frac{1}{\gamma_a}\sum_{s\in\mathcal{S}} \boldsymbol{u}_{as}\cdot\boldsymbol{\nabla}\gamma_{as},\end{aligned} \tag{3.6}$$

$$\begin{aligned}\mathrm{Lap}_a^{\gamma}(\{B_b\},\{A_b\}) \coloneqq & \frac{1}{\gamma_a}\sum_{b\,\in\,\mathcal{P}} V_b 2\overline{B}_{ab}\frac{A_{ab}}{r_{ab}^2}\boldsymbol{x}_{ab}\cdot\boldsymbol{\nabla}w_{ab} \\ & -\frac{1}{\gamma_a}\sum_{s\,\in\,\mathcal{S}}(B_s\boldsymbol{\nabla}A_s + B_a\boldsymbol{\nabla}A_a)\cdot\boldsymbol{\nabla}\gamma_{as}.\end{aligned} \tag{3.7}$$

The factor $\overline{B}_{ab}$ is built with B_a and B_b. In most of the SPH literature, it is taken as their arithmetic mean. However, to ensure continuity of shear flux, i.e. $B_a\boldsymbol{\nabla}A_a\cdot\boldsymbol{x}_{ab} = B_b\boldsymbol{\nabla}A_b\cdot\boldsymbol{x}_{ab}$, it is taken here as their harmonic mean:

$$\overline{B}_{ab} = \frac{2B_aB_b}{B_a+B_b}. \tag{3.8}$$

Furthermore, as a first order approximation, the two boundary terms of Equation 3.7 can be equalised, so that $(B_s\boldsymbol{\nabla}A_s + B_a\boldsymbol{\nabla}A_a)\cdot\boldsymbol{\nabla}\gamma_{as} \approx 2\,|\boldsymbol{\nabla}\gamma_{as}|\,B_s\boldsymbol{\nabla}A_s\cdot\boldsymbol{n}_s$. This is the flux at the boundary, therefore if a Neumann boundary condition is imposed, then it will replace this term and if a Dirichlet condition is imposed then it can be replaced by $-2\,|\boldsymbol{\nabla}\gamma_{as}|\,\overline{B}_{as}A_{as}/(\boldsymbol{x}_{as}\cdot\boldsymbol{n}_s)$, as explained by Ferrand et al. (2013) and Leroy et al. (2014).

3.2.4 Space discretised equations

The equations to be solved are the momentum equation, the equation of motion, the governing equation for γ_a and the continuity equation, which reads:

$$\begin{aligned}\rho_a\frac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t} &= -\mathbf{Grad}_a^{\gamma^+}\{p_b\} + \mathbf{Lap}_a^{\gamma}(\{\mu_b\},\{\boldsymbol{u}_b\}) + \rho_a\boldsymbol{g}, \\ \frac{\mathrm{d}\boldsymbol{x}_a}{\mathrm{d}t} &= \boldsymbol{v_a}, \\ \frac{\mathrm{d}\gamma_a}{\mathrm{d}t} &= \boldsymbol{\nabla}\gamma_{as}\cdot\boldsymbol{v}_{as}, \\ \frac{\mathrm{d}\rho_a}{\mathrm{d}t} &= -\rho_a\mathrm{Div}_a^{\gamma^-}\{\boldsymbol{u}_b\}.\end{aligned} \tag{3.9}$$

It is important to underline that the particle velocity $\boldsymbol{v}_a$ is used to move the particles and update γ_a, while the fluid velocity $\boldsymbol{u}_a$ is used in viscous forces and density evolution.

When time is considered as continuous and in the absence of open-boundaries, the last two lines of Equations (3.9) are fully equivalent to Ferrand et al. (2013):

$$\mathrm{d}(\gamma_a\rho_a) = \mathrm{d}\left(\sum_{b\in\mathcal{P}} M_b w_{ab}\right), \tag{3.10}$$

This formulation allows exact integration with time, and thus will be preferred in the following. However, in presence of inlet or outlet boundaries, the equivalence between Equations (3.9) and Equation (3.10) needs to be modified. The necessary developments are detailed in §3.3.1.

3.2.5 Time stepping without open boundaries

Time stepping scheme used in Ferrand et al. (2013) to solve the system of Equations (3.9) is recalled hereafter:

$$
\begin{aligned}
\boldsymbol{f}_a^n &= \mathbf{Lap}_a^{\gamma}\left(\left\{\mu_b^n\right\},\left\{\boldsymbol{u}_b^n\right\}\right) - \mathbf{Grad}_a^{\gamma^+}\left\{p_b^n\right\} + \rho_a^n \boldsymbol{g},\\
\boldsymbol{u}_a^{n+1} &= \boldsymbol{u}_a^n + \frac{\Delta t}{\rho_a^n}\boldsymbol{f}_a^n,\\
\boldsymbol{v}_a^{n+1} &= \boldsymbol{u}_a^{n+1},\\
\boldsymbol{x}_a^{n+1} &= \boldsymbol{x}_a^n + \Delta t \boldsymbol{v}_a^{n+1},\\
\gamma_a^{n+1} &= \gamma_a^n + \Delta t \sum_{s\in\mathcal{S}} \frac{1}{2}\left(\boldsymbol{\nabla}\gamma_{as}^{n+1} + \boldsymbol{\nabla}\gamma_{as}^n\right)\cdot \boldsymbol{v}_{as}^{n+1},\\
\rho_a^{n+1} &= \frac{1}{\gamma_a^{n+1}}\left[\gamma_a^n\rho_a^n + \sum_{b\in\mathcal{P}} M_b\left(w_{ab}^{n+1} - w_{ab}^n\right)\right],
\end{aligned}
\tag{3.11}
$$

where $p^n = p(\rho^n)$ using Equation (3.1).

Ferrand et al. (2013) considered $\boldsymbol{v}_a = \boldsymbol{u}_a$ since they did not deal with open boundaries, contrary to the present work.

In Section 3.3, this scheme will be adapted to account for ingoing or outgoing mass fluxes at open boundaries.

3.2.6 Volume diffusion correction

In the SPH operators used in Equations (3.9), the pressure and velocities are collocated, i.e. stored at the same points (the particle positions), and therefore require stabilisation. Several authors have developed stabilisation terms for SPH in order to solve this issue. We will refer the reader to the work by Fatehi and Manzari (2011) or to the work by Ferrari et al. (2009) whose work was adapted to Semi-Analytical SPH boundaries by Mayrhofer et al. (2012).

In the present work the stabilisation factor is derived from the pressure-velocity coupling on colocated methods available in the literature for the finite elements mesh-based method, namely Brezzi and Pitkäranta (1984), which was adapted to Semi-Analytical SPH boundaries by Ghaitanellis et al. (2015). The main principle behind this correction is to develop the term in $\boldsymbol{u}_b^{n+1}$ in the continuity equation. In the absence of viscosity, this gives:

$$
\begin{aligned}
\frac{\rho_a^{n+1} - \rho_a^n}{\Delta t} &= -\rho_a^n \mathrm{Div}_a^{\gamma^-}\left\{\boldsymbol{u}_b^n - \frac{\Delta t}{\rho_a^n}\mathbf{Grad}_a^{\gamma^+}\left\{p_b^n\right\} + \frac{\Delta t}{\rho_a^n}\rho_a^n\boldsymbol{g}\right\}\\
&\approx -\rho_a^n \mathrm{Div}_a^{\gamma^-}\left\{\boldsymbol{u}_b^n\right\}\\
&\quad -\rho_a^n \mathrm{Div}_a^{\gamma^-}\left\{-\frac{\Delta t}{\rho_a^n}\mathbf{Grad}_a^{\gamma^+}\left\{\tilde{p}_b^n\right\} + \Delta t\,\mathbf{Grad}_a^{\gamma^+}\left\{\boldsymbol{g}\cdot\boldsymbol{x}_b^n\right\}\right\}.
\end{aligned}
\tag{3.12}
$$

From this reasoning, a diffusion term Δ can be defined as:

$$
\Delta^n = \rho_a^n\left(\mathrm{Lap}_a^{\gamma}\left(\left\{\frac{\Delta t}{\rho_b^n}\right\},\left\{p_b^n\right\}\right) - \mathrm{Lap}_a^{\gamma}\left(\left\{\Delta t\right\},\left\{\boldsymbol{g}\cdot\boldsymbol{x}_b^n\right\}\right)\right).
\tag{3.13}
$$

Note that Δ is not exactly equal to the second term of Equation (3.12) since with SPH operators the Laplacian is not tantamount to the divergence of a gradient. Following Brezzi and

Pitkäranta (1984) this diffusion term will be weighted by an arbitrary factor $\Lambda \in [0; 1]$, and it will be added to the continuity equation. The last part of Equations (3.11) will therefore be rewritten as:

$$\begin{aligned} \tilde{\rho}_a &= \frac{1}{\gamma_a^{n+1}} \left[\gamma_a^n \rho_a^n + \sum_{b \in \mathcal{P}} M_b \left(w_{ab}^{n+1} - w_{ab}^n \right) \right], \\ \frac{\rho_a^{n+1} - \tilde{\rho}_a}{\Delta t} &= \Lambda \tilde{\rho}_a \left(\mathrm{Lap}_a^\gamma \left(\left\{ \frac{\Delta t}{\rho_b} \right\}, \{p_b\} \right) - \mathrm{Lap}_a^\gamma \left(\{\Delta t\}, \{\boldsymbol{g} \cdot \boldsymbol{x}_b\} \right) \right), \end{aligned} \tag{3.14}$$

where $\tilde{p}_b = p(\tilde{\rho}_b)$, and Λ is usually set to 0.1. In addition, the Laplacian operator in Equation (3.14) excludes the boundary elements. This correction is similar to the one developed by Mayrhofer et al. (2012), but it is still valid when the density presents large variations. We prefer the present proposition to the ones existing in the SPH literature because it comes from well established stabilisation methods in the finite volume / finite element methods.

3.2.7 Boundary conditions

Discretising the boundary in terms of segments and vertices gives a method of dealing with the boundaries on the domain, however the method for imposing boundary conditions is not trivial. The approach chosen by Ferrand et al. (2013), and later extended in Leroy et al. (2015), is that Dirichlet boundary conditions will be imposed on the vertices and Neumann boundary conditions will be imposed on the segments. An averaging is then used to find the corresponding values on the segments or vertices.

For example, the shear stress on the walls (calculated either through the no-slip condition or through a turbulence wall function) is set by imposing the fluid velocity on the wall vertices, and the shear stress of the segments is then found by averaging the velocity of the connected vertices.

3.3 Formulation for unified semi-analytical open boundary condition

3.3.1 Time integration of the continuity equation with unified semi-analytical open-boundary conditions

In this Section, the continuity equation will be carefully integrated between two successive iterations. The aim is to integrate exactly all the terms which depend only on particle positions, in order to stop the imposed mass flux from introducing errors on the density at each time step. Otherwise, these errors would lead to density discontinuities for fluid particles near open boundaries, which would in turn result in spurious shock-waves when particles are created or deleted.

While integrating the continuity equation, new boundary terms will naturally appear due to the presence of open boundaries. Their time integration will be performed consistently with the time stepping scheme proposed in Section 3.2.5, so that no artificial density perturbation is induced by the open boundaries.

As stated above, the time-dependent SPH continuity equation is the fourth line of the system (3.9) involving the fluid velocity $\boldsymbol{u}_b$. Let us now rewrite this equation as a total derivative so that the time integration will be exact. To do so, the continuity equation is first rewritten using the divergence-operator definition of Equation (3.6):

$$\begin{aligned} \mathrm{d}\rho_a &= -\rho_a \mathrm{Div}_a^{\gamma^-} \{\boldsymbol{u}_b\} \, \mathrm{d}t \\ &= \frac{1}{\gamma_a} \sum_{b \in \mathcal{P}} M_b \boldsymbol{\nabla} w_{ab} \cdot \boldsymbol{u}_{ab} \mathrm{d}t - \frac{\rho_a}{\gamma_a} \sum_{s \in \mathcal{S}} \boldsymbol{\nabla} \gamma_{as} \cdot \boldsymbol{u}_{as} \mathrm{d}t. \end{aligned} \tag{3.15}$$

To transform the sums in Equation (3.15) into exact total derivatives, the fluid velocity $\boldsymbol{u}_b$ should be replaced by the particle velocity $\boldsymbol{v}_b$. This is possible for all particles except for the sets of vertices and segments which belong to the open boundaries (denoted by $\mathcal{V}^{\mathrm{i/o}}$ and $\mathcal{S}^{\mathrm{i/o}}$, respectively), and thus the density variation reads:

$$\begin{aligned}\mathrm{d}\rho_a = &\frac{1}{\gamma_a}\sum_{b\in\mathcal{P}} M_b \boldsymbol{\nabla} w_{ab}\cdot\boldsymbol{v}_{ab}\mathrm{d}t - \frac{1}{\gamma_a}\sum_{v\in\mathcal{V}^{\mathrm{i/o}}} M_v \boldsymbol{\nabla} w_{av}\cdot(\boldsymbol{u}_v-\boldsymbol{v}_v)\,\mathrm{d}t\\ &-\frac{\rho_a}{\gamma_a}\sum_{s\in\mathcal{S}}\boldsymbol{\nabla}\gamma_{as}\cdot\boldsymbol{v}_{as}\mathrm{d}t + \frac{\rho_a}{\gamma_a}\sum_{s\in\mathcal{S}^{\mathrm{i/o}}}\boldsymbol{\nabla}\gamma_{as}\cdot(\boldsymbol{u}_s-\boldsymbol{v}_s)\,\mathrm{d}t.\end{aligned} \tag{3.16}$$

In order to compute the first and third terms of Equation (3.16), we use the fact that in a Lagrangian frame $\mathrm{d}\boldsymbol{x} = \boldsymbol{v}\mathrm{d}t$, so that:

$$\begin{aligned}\mathrm{d}w_{ab} &= \boldsymbol{\nabla} w_{ab}\cdot\boldsymbol{v}_{ab}\mathrm{d}t,\\ \mathrm{d}\gamma_a &= \sum_{s\in\mathcal{S}}\boldsymbol{\nabla}\gamma_{as}\cdot\boldsymbol{v}_{as}\mathrm{d}t.\end{aligned} \tag{3.17}$$

Therefore Equation (3.16) can be rewritten as:

$$\mathrm{d}(\gamma_a\rho_a) = \gamma_a\mathrm{d}\rho_a + \rho_a\mathrm{d}\gamma_a = \mathrm{d}\left(\sum_{b\in\mathcal{P}} M_b w_{ab}\right) - \gamma_a\delta\rho_a^{\mathrm{i/o}} + \rho_a\delta\gamma_a^{\mathrm{i/o}}, \tag{3.18}$$

where $\delta\rho_a^{\mathrm{i/o}}$ and $\delta\gamma_a^{\mathrm{i/o}}$ are defined by:

$$\delta\rho_a^{\mathrm{i/o}} := \frac{1}{\gamma_a}\sum_{v\in\mathcal{V}^{\mathrm{i/o}}} M_v\boldsymbol{\nabla} w_{av}\cdot(\boldsymbol{u}_v-\boldsymbol{v}_v)\,\mathrm{d}t, \tag{3.19}$$

$$\delta\gamma_a^{\mathrm{i/o}} := \sum_{s\in\mathcal{S}^{\mathrm{i/o}}}\boldsymbol{\nabla}\gamma_{as}\cdot(\boldsymbol{u}_s-\boldsymbol{v}_s)\,\mathrm{d}t. \tag{3.20}$$

It should be noted that Equation (3.18) is a generalized version of Equation (3.10). Furthermore the terms in Equations (D.1) and (D.1) are non-zero because particle and fluid velocities do not coincide at open boundaries, as already pointed out. In addition, for Equation (3.18) to be valid, the particle masses should be constant over the time integration of $\gamma_a\rho_a$. Therefore, the particles masses M_b are kept constant on the time interval $\left]t^n;\, t^{n+1}\right[$ equal to M_b^n. Since the vertex masses will vary in time (see §3.3.3), this implies that these variations will occur after the time integration of the continuity equation.

Let us now integrate the continuity Equation (3.18) from time t^n to $t^{n+1} = t^n + \Delta t$, to have a consistent time marching scheme:

$$\begin{aligned}(\gamma_a\rho_a)^{n+1} - (\gamma_a\rho_a)^n = &\sum_{b\in\mathcal{P}^n}\left(M_b^n w_{ab}^{n+1} - M_b^n w_{ab}^n\right)\\ &-\int_{t^n}^{t^{n+1}}\gamma_a\delta\rho_a^{\mathrm{i/o}} + \int_{t^n}^{t^{n+1}}\rho_a\delta\gamma_a^{\mathrm{i/o}}.\end{aligned} \tag{3.21}$$

Note that the summation is over the particles existing at time n, denoted by $\mathcal{P}^n$. The particle creation/deletion algorithm is described in §3.3.3.

The two time-integrated terms $\int_{t^n}^{t^{n+1}} \gamma_a \delta\rho_a^{\mathrm{i/o}}$ and $\int_{t^n}^{t^{n+1}} \rho_a \delta\gamma_a^{\mathrm{i/o}}$ in Equation (3.21) will be called virtual variation terms, and are the only modification of the time-independent continuity equation proposed by Ferrand et al. (2013) (last line of Equation 3.11).

The key point in the time integration of the virtual variations is that they are expected to maintain exactly a uniform flow field with a constant density. The term $\gamma_a \delta\rho_a^{\mathrm{i/o}}$ represents the variation experienced by $\sum_v M_v w_{av}$, with $v \in \mathcal{V}^{\mathrm{i/o}}$, if the vertex particles v were moved with the velocity $(\boldsymbol{u}_v - \boldsymbol{v}_v)$ while the fluid particle a is fixed. Similarly the term $\delta\gamma_a^{\mathrm{i/o}}$ represents the variation experienced by γ_a if the segments $s \in \mathcal{S}^{\mathrm{i/o}}$ were moved with the velocity $(\boldsymbol{u}_s - \boldsymbol{v}_s)$ while the fluid particle a is fixed.

Consequently the virtual displacement $\delta\boldsymbol{x}_a^{\mathrm{i/o}} := \Delta t \left(\boldsymbol{u}_a^n - \boldsymbol{v}_a^n\right)$ is used to compute the virtual variation terms. Thus the time-integration of $\delta\rho_a^{\mathrm{i/o}}$ and $\delta\gamma_a^{\mathrm{i/o}}$ can be computed consistently with the last two lines of Equation (3.11) as:

$$\int_{t^n}^{t^{n+1}} \gamma_a \delta\rho_a^{\mathrm{i/o}} = \sum_{v \in \mathcal{V}^{\mathrm{i/o}}} M_v^n \Big[w\left(\boldsymbol{x}_{av}^n + \delta\boldsymbol{x}_v^{\mathrm{i/o}}\right) - w\left(\boldsymbol{x}_{av}^n\right) \Big], \tag{3.22}$$

and:

$$\int_{t^n}^{t^{n+1}} \rho_a \delta\gamma_a^{\mathrm{i/o}} = \frac{\rho_a^n}{2} \sum_{s \in \mathcal{S}^{\mathrm{i/o}}} \left[\boldsymbol{\nabla}\gamma_{as}\left(\boldsymbol{x}_{as}^n + \delta\boldsymbol{x}_s^{\mathrm{i/o}}\right) + \boldsymbol{\nabla}\gamma_{as}\left(\boldsymbol{x}_{as}^n\right) \right] \cdot \delta\boldsymbol{x}_s^{\mathrm{i/o}}, \tag{3.23}$$

where $\delta\boldsymbol{x}_s^{\mathrm{i/o}} = \Delta t(\boldsymbol{u}_s^n - \boldsymbol{v}_s^n)$, the velocity of the segments are defined by $\boldsymbol{u}_s = (\boldsymbol{u}_{v_1} + \boldsymbol{u}_{v_2})/2$ and $\delta\boldsymbol{x}_v^{\mathrm{i/o}} = \Delta t(\boldsymbol{u}_v^n - \boldsymbol{v}_v^n)$. In the above notation, v_1 and v_2 are the vertices connected to s.

Equation (3.23) is consistent with the way γ is calculated. If an analytical formulation of γ is used (such as in Leroy et al. (2014)) it should be computed as:

$$\int_{t^n}^{t^{n+1}} \rho_a \delta\gamma_a^{\mathrm{i/o}} = \rho_a^n \sum_{s \in \mathcal{S}^{\mathrm{i/o}}} \left[\gamma_{as}\left(\boldsymbol{x}_{as}^n + \delta\boldsymbol{x}_s^{\mathrm{i/o}}\right) - \gamma_{as}\left(\boldsymbol{x}_{as}^n\right) \right]. \tag{3.24}$$

The proposed continuity Equation (3.21) allows one to take account of ingoing/outgoing mass flux through the last two terms, while other terms depend only on particle positions; Equation (3.21) combined with Equations (3.22) and (3.23) gives:

$$\begin{aligned} \rho_a^{n+1} = \frac{1}{\gamma_a^{n+1}} \Bigg\{ & \gamma_a^n \rho_a^n + \sum_{b \in \mathcal{P}^n} M_b^n \left(w_{ab}^{n+1} - w_{ab}^n \right) \\ & + \sum_{v \in \mathcal{V}^{\mathrm{i/o}}} M_v^n \left[w_{av}^n - w\left(\boldsymbol{x}_{av}^n + \delta\boldsymbol{x}_v^{\mathrm{i/o}}\right) \right] \\ & + \frac{\rho_a^n}{2} \sum_{s \in \mathcal{S}^{\mathrm{i/o}}} \left[\boldsymbol{\nabla}\gamma_{as}\left(\boldsymbol{x}_{as}^n + \delta\boldsymbol{x}_s^{\mathrm{i/o}}\right) + \boldsymbol{\nabla}\gamma_{as}\left(\boldsymbol{x}_{as}^n\right) \right] \cdot \delta\boldsymbol{x}_s^{\mathrm{i/o}} \Bigg\}. \end{aligned} \tag{3.25}$$

For simplicity the summations in the right-hand-side of Equation (3.25) will be noted as $d_a^{\mathrm{i/o}}$, which gives:

$$\rho_a^{n+1} = \frac{1}{\gamma_a^{n+1}} \left(\gamma_a^n \rho_a^n + d_a^{\mathrm{i/o}} \right). \tag{3.26}$$

3.3.2 Time stepping with open boundaries

The time stepping scheme used in Equation (3.11) is therefore adapted to take into account the open boundaries with a volume diffusion correction (Section 3.2.6):

$$
\begin{aligned}
\boldsymbol{f}_a^n &= \mathbf{Lap}_a^{\gamma}\left(\{\mu_b^n\},\{\boldsymbol{u}_b^n\}\right) - \mathbf{Grad}_a^{\gamma^+}\{p_b^n\} + \rho_a^n \boldsymbol{g},\\
\boldsymbol{u}_a^{n+1} &= \boldsymbol{u}_a^n + \frac{\Delta t}{\rho_a^n}\boldsymbol{f}_a^n,\\
\boldsymbol{v}_a^{n+1} &= \boldsymbol{u}_a^{n+1} \qquad \text{for } a \in \mathcal{F},\\
\boldsymbol{x}_a^{n+1} &= \boldsymbol{x}_a^n + \Delta t \boldsymbol{v}_a^{n+1},\\
\gamma_a^{n+1} &= \gamma_a^n + \Delta t \sum_{s\in\mathcal{S}} \frac{1}{2}\left(\boldsymbol{\nabla}\gamma_{as}^{n+1} + \boldsymbol{\nabla}\gamma_{as}^n\right)\cdot \boldsymbol{v}_{as}^{n+1},\\
\tilde{\rho}_a &= \frac{1}{\gamma_a^{n+1}}\left(\gamma_a^n\rho_a^n + d_a^{\mathrm{i/o}}\right),\\
\frac{\rho_a^{n+1}-\tilde{\rho}_a}{\Delta t} &= \Lambda\tilde{\rho}_a\left(\mathrm{Lap}_a^{\gamma}\left(\left\{\frac{\Delta t}{\rho_b}\right\},\{p_b\}\right) - \mathrm{Lap}_a^{\gamma}\left(\{\Delta t\},\{\boldsymbol{g}\cdot\boldsymbol{x}_b\}\right)\right).
\end{aligned}
\tag{3.27}
$$

As a reminder $d_a^{\mathrm{i/o}}$ is defined through Equations (3.25) and (3.26). Furthermore, the constraints on the time step Δt are described in §3.B.4.

3.3.3 Mass update of vertex particles

The method used to create or remove particles needs to be carefully developed in order to keep a correct particle distribution near the open boundaries (this process can be understood as re-meshing). The choice made is to make the vertex particles grow (resp. decrease) over time on an inlet (resp. outlet) boundary. Thus their masses and volumes are not constant. However, the mass evolution should not create any perturbation neither on the density nor on the momentum. Therefore, as stated earlier a constant mass will be set between t^n and t^{n+1} in Equations (3.27). This is ensured by Equations (3.21), (3.22) and (3.23). The masses will therefore be updated outside of the marching scheme.

Nonetheless, the total mass of the system should change smoothly in time. To do so the masses of the vertex particles in $v \in \mathcal{V}^{\mathrm{i/o}}$ take into account the desired mass flux through the open boundary $\dot{M}_v$. This mass flux is computed as a weighted average of the mass fluxes through the segments $s \in \mathcal{N}_{sv}$, where $\mathcal{N}_{sv}$ is the set of all the segments connected to v (usually two segments in 2-D):

$$
\dot{M}_v = \frac{1}{2}\sum_{s\in\mathcal{N}_{sv}^{i/o}} \rho_s S_s(\boldsymbol{u}_s - \boldsymbol{v}_s)\cdot\boldsymbol{n}_s, \qquad \forall v \in \mathcal{V}^{\mathrm{i/o}},
\tag{3.28}
$$

where S_s is the length of a segment. The flux $\dot{M}_v$ is positive for an inlet and negative for an outlet.

The flux $\dot{M}_v$ is used to update the vertex masses. Instinctively one would assume that $M_v^{n+1} = M_v^n + \Delta t \dot{M}_v^n$. However, in the case of an inlet, new fluid particles must be created near the open boundary when the fluid particles go far away from it (see Figure 3.2). For this purpose, each time the vertex particle mass M_v reaches a certain threshold $\theta_v M_{\mathrm{ref}}$, a fluid particle is created at the same position with the reference mass M_{ref} and this mass is subtracted from the vertex particle mass to prevent from any density shock. For the sake of simplicity θ_v is the open angle of the vertex divided by 2π (i.e. $\theta_v = 0.5$ for a plane boundary) and M_{ref} is the mass of a fluid particle.

In the case of an outlet, particles are removed when they cross an open boundary segment. In order to remain consistent the mass of an outgoing fluid particle is redistributed over the open

(a) t_0. (b) t_1. (c) t_2. (d) t_3.

Figure 3.2: Mass evolution and particle creation for an inlet: the mass of vertex particles v is growing due to a positive flux $\boldsymbol{u}_s \cdot \boldsymbol{n}_s$ at the segment s (times t_0 to t_3). When the masses of the inlet vertices reach the defined threshold of $M_{\text{ref}}/2$ (time t_2), new fluid particles are created and the mass of vertex particles is decreased.

boundary vertices connected to this segment. Each of these vertices receives the mass $\beta_{a,v} M_a$, where $\beta_{a,v}$ is a distribution (or weighting) factor.

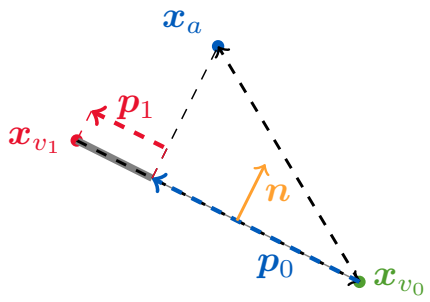


Figure 3.3: Mass distribution process for an outgoing fluid particle a crossing a segment s connecting vertices v_0 and v_1. The dark gray line represents the factor β_{a,v_0}.

The factor $\beta_{a,v}$ is constructed so that most of the mass is distributed to the closest vertex. For a segment s with vertices v_0 and v_1, vector $\boldsymbol{p}_i$ is defined as the vector between vertex v_i and the projection of the particle a on the segment, i.e. $\boldsymbol{p}_i = \boldsymbol{x}_{av_i} - \left(\boldsymbol{x}_{av_i} \cdot \boldsymbol{n}\right) \boldsymbol{n}$ (see Figure 3.3). Assuming that the projection belongs to the segment, then for v_0 and v_1 connected to s β_{a,v_i} is defined as:

$$\begin{aligned} \beta_{a,v_0} &= \frac{\boldsymbol{p}_1 \cdot \boldsymbol{x}_{v_0 v_1}}{\|\boldsymbol{x}_{v_0 v_1}\|^2}, \\ \beta_{a,v_1} &= \frac{\boldsymbol{p}_0 \cdot \boldsymbol{x}_{v_1 v_0}}{\|\boldsymbol{x}_{v_1 v_0}\|^2} = 1 - \beta_{a,v_0}. \end{aligned} \tag{3.29}$$

Otherwise, $\beta_{a,v}$ is set to zero.

Therefore, the masses of vertex particles are updated as follows:

$$M_v^{n+1} = M_v^n + \Delta t \dot{M}_v^n + \delta M_v^n, \qquad \forall v \in \mathcal{V}^{\text{i/o}}, \tag{3.30}$$

where δM_v^n quantifies the mass variation due to particle creation/destruction, i.e. $-M_{\text{ref}}$ each time a fluid particle is created at position v, $+\beta_{a,v} M_a$ each time a particle a crosses a segment connected to v. As stated above, the step described by Equation (3.30) is added at the end of the time stepping scheme (3.27).

In Equation (3.30) only the mass flux positiveness determines whether an open boundary repulses or attracts fluid particles. As illustrated later in the periodic wave test case, i.e. Figure 3.12, it is possible to prescribe time-dependent velocities moving from negative to positive values. Henceforth, the proposed strategy makes it straightforward to treat both inlet and outlet at the same open boundary, depending on space and time.

Free-surface flows can also be treated that way. For example, for the periodic wave shown in Figure 3.12 and later in Section 3.4.3, the mass flux $\dot{M}_v$ at the vertical open boundaries oscillates consistently from positive values to negative values as the velocity is prescribed according to 5th order solution to Stokes wave theory.

3.3.4 Imposing pressure and velocity

The Riemann problem formulation described in 3.A is used to define compatible pressure and velocity fields. The principle is to use a method common in finite volumes, where characteristic waves (λ_{-1}, λ_0 and λ_{+1}) model discontinuities between the exterior state (the boundary conditions) and the interior state (the fluid) through generalized Riemann invariants (GRI).

Imposed velocity

When a velocity field is imposed on a boundary, for instance an incoming free-surface wave velocity profile at an inlet, it is imposed on the open boundary vertices (as it is a Dirichlet boundary condition). Since the imposed velocity field is needed on the open boundary segments, its value will be averaged from the neighbouring vertices:

$$\boldsymbol{u}_s = \frac{1}{|\mathcal{N}_{vs}|} \sum_{v \in \mathcal{N}_{vs}} \boldsymbol{u}_v. \tag{3.31}$$

This value will then be the prescribed velocity field on the exterior state, i.e. $\boldsymbol{u}_{ext} = \boldsymbol{u}_s$. In this case the linearised Riemann problem is used to compute the value of the density (or pressure) fields at the exterior state, and therefore at the boundary segments and vertices.

The GRI of wave λ_{+1} will therefore be used to calculate the external state pressure p_{ext} from the internal state. However the type of discontinuity of the wave λ_{+1} needs to be defined. It is known that since $\lambda_{+1,ext} = \lambda_{+1,2}$, then if $\lambda_{+1,ext} > \lambda_{+1,int}$ the wave is a shock, however since p_{ext} is unknown then c_{ext} cannot be calculated. Therefore assumptions needs to be made regarding the relationship $\lambda_{+1,ext} > \lambda_{+1,int}$. It will be assumed that this condition is verified if $u_{n,ext} > u_{n,int}$, but this assumption will need to be verified once c_{ext} is known. Therefore:

1. If $u_{n,ext} > u_{n,int}$, the λ_{+1} wave is a shock wave. The Rankine-Hugoniot relationships have to be used through the following equation:

$$p_{ext} = p_{int} + \rho_{int} u_{n,int} \left(u_{n,int} - u_{n,ext} \right). \tag{3.32}$$

2. If $u_{n,ext} \leq u_{n,int}$, the λ_{+1} wave is an expansion wave. The corresponding Riemann invariant R_{+1} must must be used (i.e. $R_{+1,ext} = R_{+1,int}$):

$$u_{n,ext} - \psi\left(\rho_{ext}\right) = u_{n,int} - \psi\left(\rho_{int}\right), \tag{3.33}$$

where ψ is defined by Equation (3.68).

Once the pressures are known on the segments (since $p_s = p_{ext}$), then the pressure on the vertices are found by averaging the connected segments:

$$p_v = \frac{1}{|\mathcal{N}_{sv}|} \sum_{s \in \mathcal{N}_{sv}} p_s. \tag{3.34}$$

The densities are once again calculated using Equation (3.1).

As mentioned earlier, if the imposed condition corresponds to an inlet then the tangential velocity components need to be defined, otherwise they will be defined from the interior state.

Imposed pressure (or density)

When a pressure field is imposed on a boundary, for instance hydrostatic pressure at an outlet, it is imposed on the vertices (as it is a Dirichlet boundary condition). Since the imposed pressure field is needed on the open boundary segments, to do so its value will be averaged from the neighbouring vertices:

$$p_s = \frac{1}{|\mathcal{N}_{vs}|} \sum_{v \in \mathcal{N}_{vs}} p_v. \tag{3.35}$$

This value will then be the prescribed pressure field on the exterior state, i.e. $p_{ext} = p_s$. In this case the linearised Riemann problem is used to compute the value of the velocity fields at the exterior state, and therefore at the boundary segment and vertices. The GRI of wave λ_{+1} will therefore be used to calculate the normal velocity component from the internal state. However the type of discontinuity of the wave λ_{+1} also needs to be defined. In this case it will be assumed that this condition is verified if $c_{ext} > c_{int}$, but this assumption will need to be verified once $u_{n,ext}$ is known. Therefore:

1. If $c_{ext} > c_{int}$, the λ_{+1} wave is a shock wave. The Rankine-Hugoniot relationships have to be used through the following equation:

$$u_{n,ext} = \frac{p_{int} - p_{ext}}{\rho_{int} u_{n,int}} + u_{n,int}. \tag{3.36}$$

   Note that because of the presence of $u_{n,int}$ in the denominator of Equation (3.36), when the pressure is imposed on a boundary adjacent to still water this formulation could pose difficulties. This is why, when $u_{n,int} \to 0$ the limit of Equation (3.36) should be considered, i.e.:

$$\lim_{u_{n,int} \to 0} u_{n,ext} = u_{n,int}. \tag{3.37}$$

   Practically, this condition will be imposed when $u_{n,ext} > c_0/10$.

2. If $c_{ext} \leq c_{int}$, the λ_{+1} wave is an expansion wave. The corresponding Riemann invariant R_{+1} must be used (i.e. $R_{+1,ext} = R_{+1,int}$):

$$u_{n,ext} - \psi(\rho_{ext}) = u_{n,int} - \psi(\rho_{int}). \tag{3.38}$$

Once the velocities are known on the segments (since $\boldsymbol{u}_s = \boldsymbol{u}_{ext}$), then the velocities on the vertices are found by averaging the connected segments:

$$\boldsymbol{u}_v = \frac{1}{|\mathcal{N}_{sv}|} \sum_{s \in \mathcal{N}_{sv}} \boldsymbol{u}_s. \tag{3.39}$$

Again, if the imposed condition corresponds to an inlet then the tangential velocity components need to be defined, otherwise they will be defined from the interior state. Finally, the densities are still calculated using Equation (3.1).

Imposed open boundary conditions algorithm

The linearised Riemann problem is used to calculate the open boundary conditions before the time stepping is done. Therefore for an open boundary problem, the calculation steps are listed below:

1. Calculation of the fields at the boundaries, using the method described in Ferrand et al. (2013) for wall boundaries, and the method described in §3.3.4 and 3.3.4 for open boundaries.

2. Time stepping of the fluid particles in the domain using Equations (3.27).

3. Update the mass of the vertex particles using Equation (3.30), create particles if $M_v > \theta_v M_{\text{ref}}$ and delete particles that have crossed an open boundary.

The method for implementing properly this algorithm in a parallel implementation is described in 3.B.

3.4 Numerical tests

For all the simulations, the quintic Wendland kernel with smoothing length $h = 2\Delta r$ will be used.

3.4.1 Non-orthogonal flux on inlet/outlet in a square

The first test case presented is that of a 2-D square fluid domain of size L with only inlet/outlet boundary conditions (see Figure 3.4). The flow is at an angle of $\pi/4$ to the normal to the boundaries; in other words, the velocity imposed at all boundaries is $\boldsymbol{u}_0 = |\boldsymbol{u}_0|/\sqrt{2}(1,1)^T$. It illustrates one of the advantages of present Semi-Analytical inlet/outlet boundary treatment: the absence of buffer layer makes it possible to prescribe a non-orthogonal flow quite easily, regardless of the open boundary shape.

The initial conditions are of constant velocity and density fields with the same values as imposed at the boundary. Particles are initially distributed onto a Cartesian grid, the inter-particle distance Δr and segment size are non-dimensionalised using the problem size, i.e. $\Delta r^+ = \Delta r/L$ for all different tests performed (from $0.8k$ to $41.6k$ particles). Furthermore the speed of sound at rest is set to $c_0 = 10|\boldsymbol{u}_0|$. The fluid viscosity ν is set to $10^{-2}\,\text{m}^2\,\text{s}^{-1}$ and the volume diffusion factor Λ will be set to 0, so that the focus is put on the open boundary conditions.

The evolution of the L_2 errors for both density and velocity is computed as:

$$\varepsilon_\rho = \sqrt{\frac{1}{N_\mathcal{P}} \sum_{a \in \mathcal{P}} \frac{(\rho_a - \rho_0)^2}{\rho_0^2}} \text{ and } \varepsilon_{\boldsymbol{u}} = \sqrt{\frac{1}{N_\mathcal{P}} \sum_{a \in \mathcal{P}} \frac{|\boldsymbol{u}_a - \boldsymbol{u}_0|^2}{|\boldsymbol{u}_0|^2}}, \tag{3.40}$$

where $N_\mathcal{P}$ is the number of fluid and vertex particles.

As a first test the Riemann invariants will not be used and both the velocity and density will be imposed as $\boldsymbol{u}_0$ and ρ_0 respectively. The evolution of the errors is displayed in Figure 3.5 with respect to non-dimensional time $t^+ = t|\boldsymbol{u}_0|/L$. It is therefore shown that the errors are very small; they are mainly linked to the round-off errors. Since the simulations presented are computed on single precision GPU (Graphics Processing Units), the values observed are acceptable.

It should be emphasised that the additional terms in the continuity equation (Equation (3.18)) are essential, as the authors have observed that the simulations without those terms crashed immediately.

As a second test, the velocity are imposed on inlets and the pressure on the outlets together with Riemann invariants. As shown by Figure 3.6, the errors are slightly larger than previously.

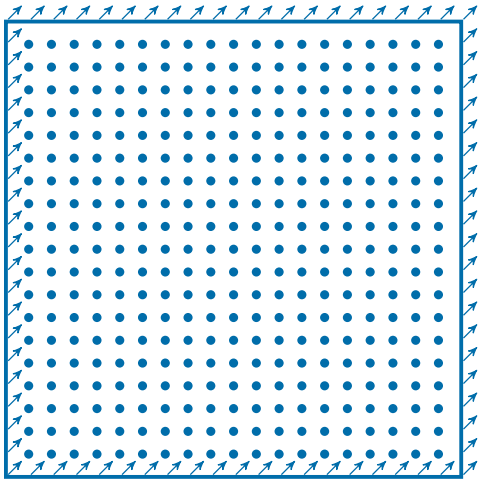


Figure 3.4: Non-orthogonal inlet/outlet in a square.

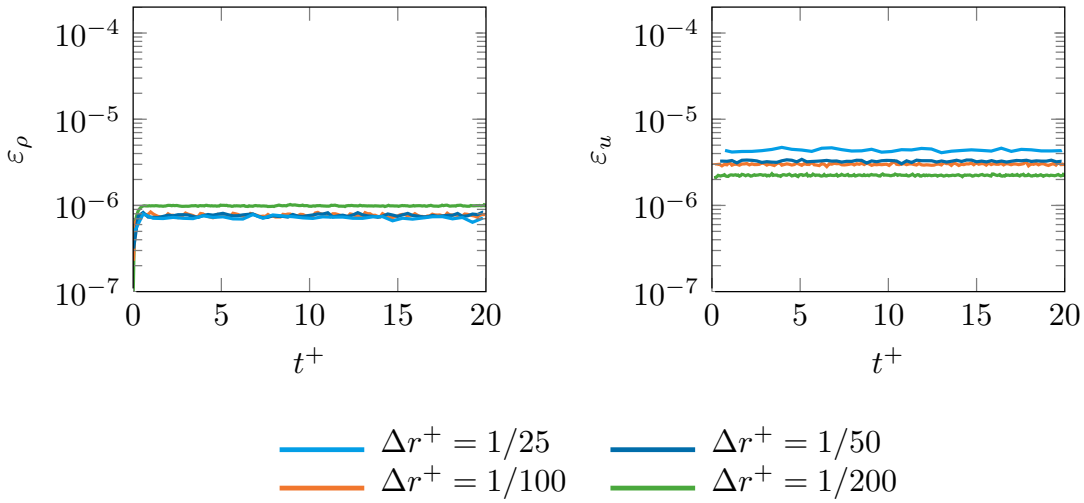


Figure 3.5: Non-orthogonal inlet/outlet in a square. Evolution of the L_2 error in density (left) and velocity (right) for different discretisations without using Riemann invariants.

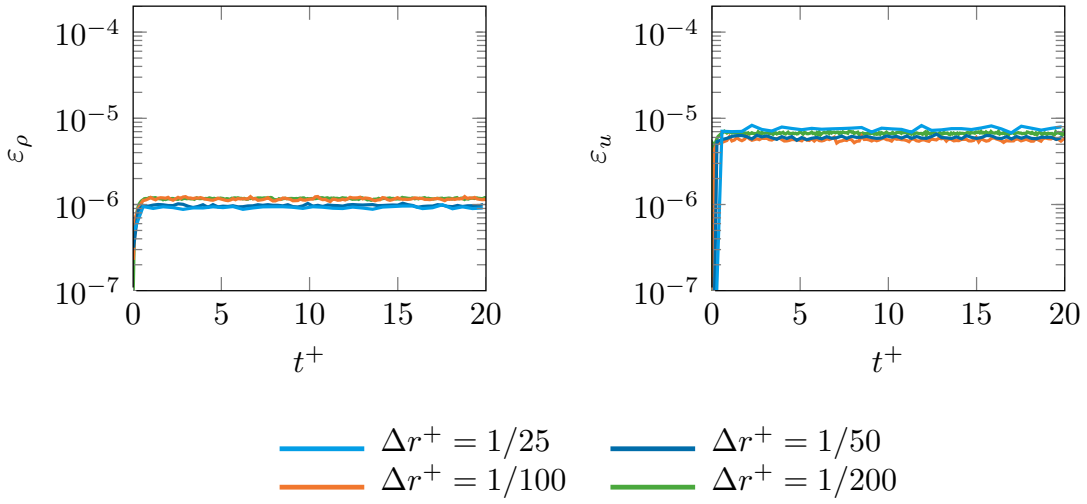


Figure 3.6: Non-orthogonal inlet/outlet in a square. Evolution of the L_2 error in density (left) and velocity (right) for different discretisations using Riemann invariants with imposed velocities on the inlets and imposed pressure on the outlets.

This is because the previous case was over-constrained. Furthermore, when using Riemann invariants at the open boundaries the error is almost independent of the particle size. However the error on the velocity is of an order of magnitude higher than that of the density. This can easily be explained by linearising the Euler equations for the numerical perturbations $\delta\rho$ and δu:

$$\frac{\delta u}{\tau} \sim \frac{c_0^2}{\rho_0}\frac{\delta\rho}{\lambda}, \tag{3.41}$$

$$\frac{\delta\rho}{\tau} \sim \rho_0\frac{\delta u}{\lambda}, \tag{3.42}$$

where τ and λ are time and length scales corresponding to the perturbation propagation.

The order of magnitude of the errors are $\varepsilon_\rho \sim \delta\rho/\rho_0$ and $\varepsilon_{\boldsymbol{u}} \sim \delta u/|\boldsymbol{u}_0|$. Combining the last two equations in order to remove τ and λ yields $\varepsilon_{\boldsymbol{u}} \sim (c_0/|\boldsymbol{u}_0|)\,\varepsilon_\rho$. The error on the velocity is thus amplified by the numerical Mach number, which is taken as 10 as explained above.

3.4.2 Rapidly expanding pipe

In this next test case, the flow from a small pipe (named pipe 1) will enter a large pipe (named pipe 2), creating a sudden expansion of the flow. See Figure 3.7 for details on the geometry. Far from the expansion, the flow should follow the Hagen-Poiseuille equations, which for a flow between two infinite plates is given by:

$$u_x = U_{\max}\left(1 - \frac{z^2}{W^2}\right), \tag{3.43}$$

$$\frac{dP}{dx} = -\frac{2\rho_0\nu U_{\max}}{W^2}, \tag{3.44}$$

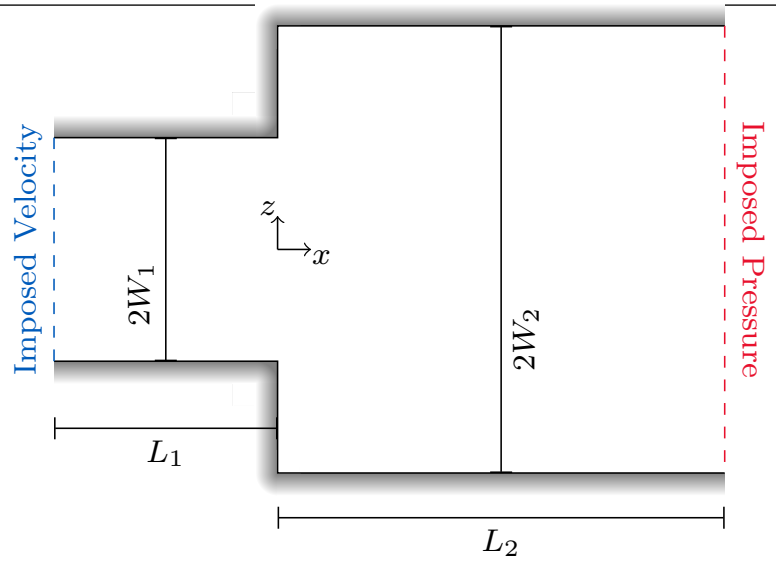


Figure 3.7: Configuration of the rapidly expanding pipe. The origin is placed at the beginning of the second pipe, and on the central axis of the pipes.

where the origin of z is on the centre pipe axis, W is the half width of the pipe and $U_{\max}$ is the maximum velocity. To ensure that the flow stays laminar, this maximum velocity will be defined according to a Reynolds number Re:

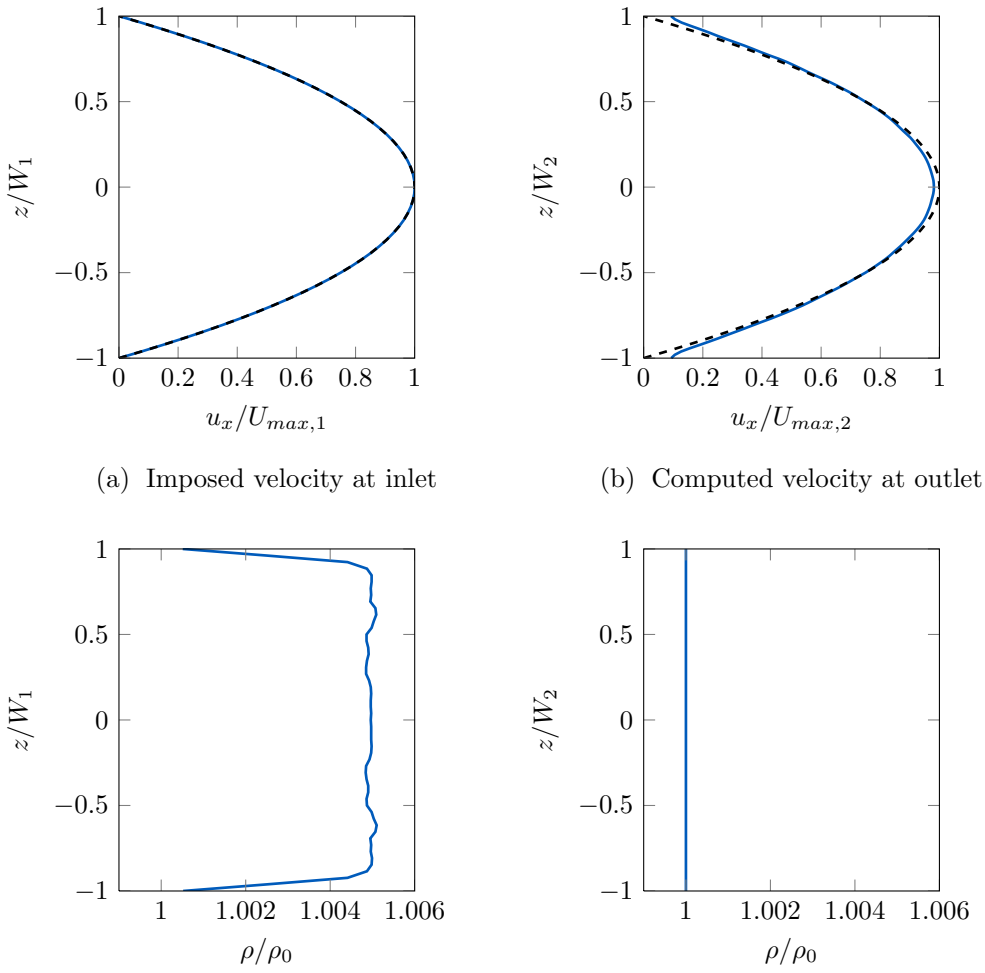
$$U_{\max} = \frac{\nu \mathrm{Re}}{2W}. \tag{3.45}$$

Equation (3.43) will therefore be used to impose a Poiseuille flow velocity profile on the inlet, and a constant pressure will be imposed on the outlet. The geometrical parameters are chosen to correspond to one of the experiments presented by Hammad et al. (1999): $W_1 = 0.13\,\mathrm{m}$, $L_1 = 3W_1$, $W_2 = 2W_1$ and $L_2 = 4W_2$. The particle spacing $\Delta r = W_1/26$ and the physical parameters chosen are $\rho_0 = 1190\,\mathrm{kg\,m^{-3}}$, $\nu = 3.19 \times 10^{-5}\,\mathrm{m^2\,s^{-1}}$ and Re = 20.6. The numerical parameters are $c_0 = 0.03\,\mathrm{m\,s^{-1}}$ and $\Lambda = 0.1$. A background pressure is used to help the flow stabilize.

Figure 3.8 shows the horizontal velocity and density profiles at the inlet and outlet. As can be seen the velocity profile at the outlet is very close to the Poiseuille flow. In addition, the density at the inlet is greater than at the outlet, which is a necessary condition as the pressure is linked to the density through the state Equation (3.1). Finally, integrating the profile in Figure 3.8 to calculate the mass fluxes at the inlet and outlet we get the following time averaged values:

- Mass flux in: $0.523\,\mathrm{kg\,m^{-1}\,s^{-1}}$
- Mass flux out: $0.525\,\mathrm{kg\,m^{-1}\,s^{-1}}$
- Analytical mass flux: $0.521\,\mathrm{kg\,m^{-1}\,s^{-1}}$

The mass flux differences appear because even though the fluid particles have fixed masses throughout the simulation, the open boundary vertices have a variable mass. The mass variations are consistent with the boundary mass flow, and the differences between the inlet and the outlet are a result of the compressible medium which cannot achieve a perfect steady state. The normalised



(a) Imposed velocity at inlet

(b) Computed velocity at outlet

(c) Computed density at inlet

(d) Imposed density at outlet

Figure 3.8: Horizontal velocity and density profiles at the inlet and the outlet of the rapidly expanding pipe. The dashed line shows the analytical solution for a Poiseuille flow, and the solid line the simulation results.

error in the mass flux is therefore equal to 4×10^{-3}, which is acceptable relative to the spatial discretisation, and it does not have a significant impact on the simulation.

In Figure 3.9 the streamlines obtained from the stream function are compared with the experimental results presented in Hammad et al. (1999) and a finite volume solution using code_saturne Archambeau et al. (2004). The calculation of the stream function Ψ from a velocity field can be done using different methods, which can be a source of differences with the experimental values. To analyse the simulation results the following equation was used:

$$\Psi(x,z) = \begin{cases} \int_{\boldsymbol{x}(L_2,W_2)}^{\boldsymbol{x}(x,z)} (u_x dz - u_z dx) & \text{above the central axis,} \\ \int_{\boldsymbol{x}(L_2,-W_2)}^{\boldsymbol{x}(x,z)} (u_x dz - u_z dx) & \text{below the central axis.} \end{cases} \tag{3.46}$$

Amongst the differences between the SPH simulation and the experimental results, the stream line at $\Psi = 0$ does not cross the wall at the same distance from the expansion, meaning that the recirculation zone is smaller in the simulations. This could be because in the state Equation (3.1)

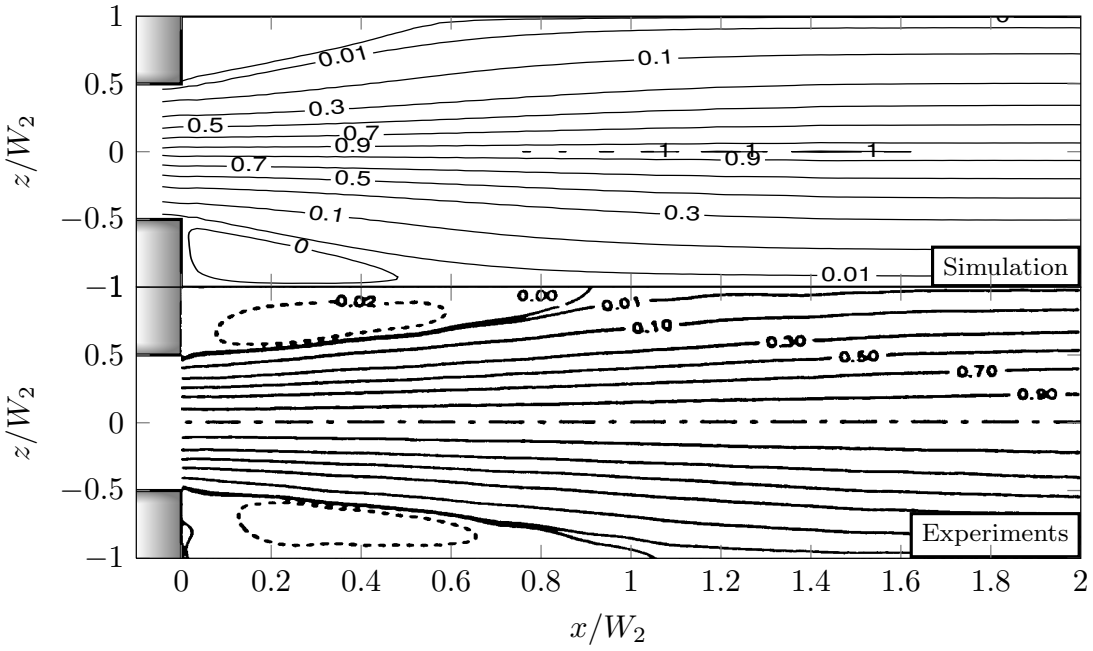


Figure 3.9: Streamlines in the rapidly expanding pipe from the SPH simulated flow (above), experimental results by Hammad et al. (1999) (middle).The values displayed at the streamlines are $\Psi/\Psi(0,0)$.

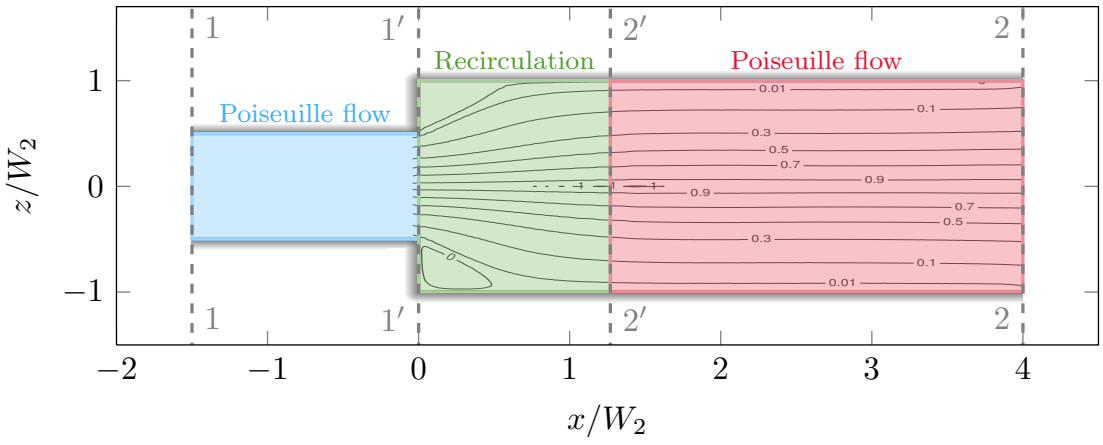


Figure 3.10: The three regions in the rapidly expanding pipe that can be defined to calculate the head loss from the simulated flow.

the value of ξ for diethylene glycol (which was used in the experiment) is not readily available, and therefore $\xi = 7$ was imposed, which is the value for water. This is also true for the simulations using code_saturne, although the incompressible model used in this case allows these results to be closer to the experimental values.

Nonetheless, at the outlet the streamlines of the simulations are close to the experimental results. As a further check of the validity of the simulation, the head loss will be modelled. Looking at the streamlines of the flow, plotted in Figure 3.10, the flow can be separated into 3 regions: a Poiseuille flow inside pipe 1, a recirculation zone created by the sudden expansion and

a Poiseuille flow inside pipe 2. One can estimate the head loss in each Section by (see Hérard and Martin (Preprint: hal-01166478) for more details):

$$\begin{cases} \frac{1}{2}\left(\widehat{U}_1^2 - \widehat{U}_{1'}^2\right) = \frac{1}{\rho_2}\left(P_{1'} - P_1\right), \\ \frac{1}{2}\left(\widehat{U}_{1'}^2 - \widehat{U}_{2'}^2\right) + \frac{1}{\rho_2}\left(P_{1'} - P_{2'}\right) = \frac{1}{2}\left(\widehat{U}_{1'} - \widehat{U}_{2'}\right)^2, \\ \frac{1}{2}\left(\widehat{U}_2^2 - \widehat{U}_{2'}^2\right) = \frac{1}{\rho_2}\left(P_{2'} - P_2\right), \end{cases} \tag{3.47}$$

Where the subscript represents the interface between different zones of the flow. 1 denotes the interface between the inlet and pipe one, $1'$ the interface between the Poiseuille flow of pipe one and the recirculation area, $2'$ the interface between the recirculation area and the Poiseuille flow of pipe two and 2 the interface between pipe two and the outlet (see Figure 3.10). Furthermore $\widehat{U}^2 = 1/(2W)\int_{-W}^{W} U^2 d\widetilde{z}$, $P = 1/(2W)\int_{-W}^{W} p d\widetilde{z}$ and P_0' is the integral along one of the expansion walls.

These three equations can be combined to give:

$$\begin{aligned} \frac{1}{2}\left(\widehat{U}_1^2 - \widehat{U}_2^2\right) + \frac{1}{\rho_2}\left(P_1 - P_2\right) &= \frac{1}{2}\left(\widehat{U}_1 - \widehat{U}_2\right)^2 + \frac{1}{\rho_2}\left(P_1 - P_{1'}\right) \\ &\quad + \frac{1}{\rho_2}\left(P_{2'} - P_2\right) + O\left(\frac{\nu \widehat{U}_1}{W_2}\right). \end{aligned} \tag{3.48}$$

Defining the head of the flow by:

$$H = \frac{\widehat{U}^2}{2g} + \frac{P}{g\rho_2}. \tag{3.49}$$

The theoretical head loss between the inlet and the outlet is thus equal to (neglecting the last term in Equation 3.48):

$$H_1 - H_2 = \frac{1}{2g}\left(\widehat{U}_1 - \widehat{U}_2\right)^2 + \frac{1}{g\rho_2}\left(P_1 - P_{1'}\right) + \frac{1}{g\rho_2}\left(P_2 - P_{2'}\right). \tag{3.50}$$

The expected head loss is therefore equal to the sum of head loss of the Poiseuille flows, and Borda-Carnot head loss in the recirculation zone. Figure 3.11 shows the head along the x-axis. Extracting the simulated head loss and calculating the theoretical head loss gives:

- Head loss from simulations = 5.04×10^{-7} m
- Theoretical head loss = 5.00×10^{-7} m

Therefore, the simulated head loss has less than a 1% difference with the theoretical head loss.

3.4.3 2-D periodic free-surface water wave

The next test case shows that these new open boundary conditions can be used for both inflows and outflows, and that open boundaries can alternate between the two without any difficulties, even with a free surface. To do so, periodic water waves propagating on a flat bed will be imposed.

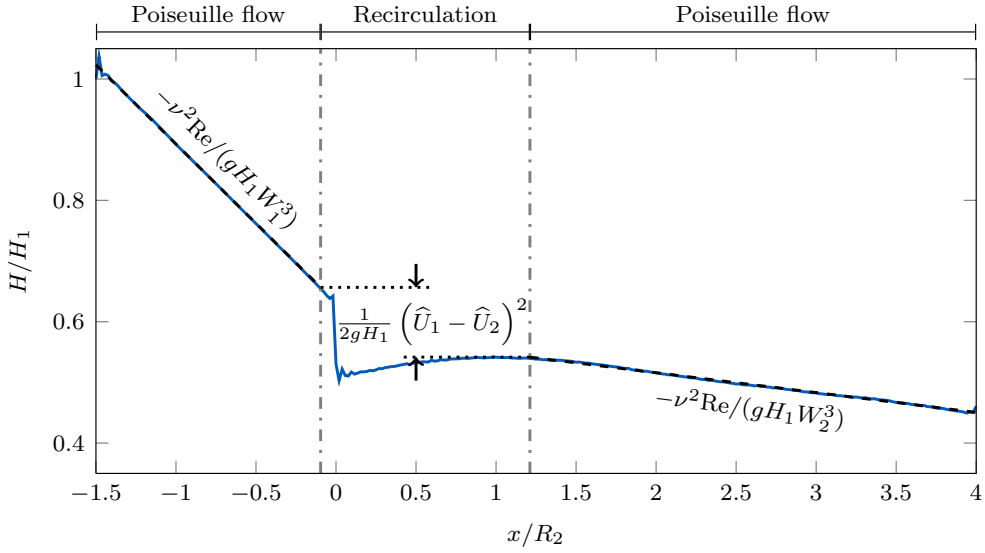


Figure 3.11: Head profile plotted along the x-axis of the rapidly expanding pipe. The blue line is the computed head, the dashed lines show the theoretical head gradient of the Poiseuille flows (from Equations (3.44) and (3.45)) and the arrows around the dotted line show the head loss from the Borda-Carnot equation.

The waves imposed are calculated from the 5$^{\text{th}}$ order solution to Stokes wave theory given by Fenton (1985):

$$k\eta(x,t) = \sum_{i=1}^{5} \epsilon^i \sum_{j=1}^{i} B_{ij} \cos\left[jk\left(x - ct + \frac{\theta}{k}\right)\right], \tag{3.51}$$

$$\overline{U}\left(\frac{k}{g}\right)^{1/2} = C_0 + \epsilon^2 C_2 + \epsilon^4 C_4, \tag{3.52}$$

$$c = \overline{U}_c + \overline{U}, \tag{3.53}$$

$$u_x(x,z,t) = \left(c - \overline{U}\right) + , \\ C_0\left(\frac{g}{k^3}\right)^{1/2} \sum_{i=1}^{5} \epsilon^i \sum_{j=1}^{i} A_{ij} \cosh(jkz)\, jk \cos\left[jk\left(x - ct + \frac{\theta}{k}\right)\right], \tag{3.54}$$

$$u_z(x,z,t) = C_0\left(\frac{g}{k^3}\right)^{1/2} \sum_{i=1}^{5} \epsilon^i \sum_{j=1}^{i} A_{ij} jk \sinh(jkz) \sin\left[jk\left(x - ct + \frac{\theta}{k}\right)\right], \tag{3.55}$$

$$\frac{Rk}{g} = \frac{1}{2}C_0^2 + kD + \epsilon^2 E_2 + \epsilon^4 E_4, \tag{3.56}$$

$$\frac{p(x,z,t)}{\rho_0} = R - gz - \frac{1}{2}\left[(u_x - c)^2 + u_z^2\right], \tag{3.57}$$

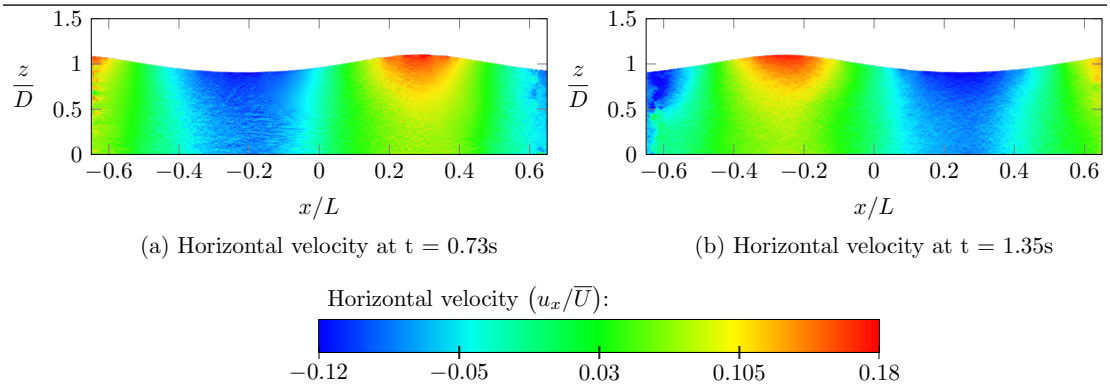
where η is the free surface elevation (with the water depth $h = D + \eta$), A is the wave amplitude (i.e. half of the wave height), D is the mean water depth, g is the acceleration due to gravity, k is the wave number (defined as $k = 2\pi/L$, where L is the wave length), c is the wave velocity, $\overline{U}_c$ is the mean current velocity, $\overline{U}$ is the mean horizontal velocity, θ is a phase constant and R is the Bernoulli constant. The wave period T can be calculated from the wave number and wave velocity, i.e. $T = 2\pi/(kc)$. Furthermore the z-axis has its origin at the bed. The constants are given in Table 3.1.

Table 3.1: Constants for the 5th order solution to Stokes wave theory given by Fenton (1985) and used in Equations (3.51) - (3.57).

$$
\begin{aligned}
A_{11} &= 1/\sinh(kD)\\
A_{22} &= 3S^2/[2(1-S)^2]\\
A_{31} &= (-4-20S+10S^2-13S^3)/[8\sinh(kD)(1-s)^3]\\
A_{33} &= (-2S^2+11S^3)/[8\sinh(kD)(1-s)^3]\\
A_{42} &= (12S-14S^2-264S^3-45S^4-13S^5)/[24(1-S)^5]\\
A_{44} &= (10S^3-174S^4+291S^5+278S^6)/[48(3+2S)(1-S)^5]\\
A_{51} &= (-1184+32S+13232S^2+21712S^3+20940S^4+12554S^5-500S^6\\
&\quad -3341S^7-670S^8)/[64\sinh(kD)(3+2S)(4+S)(1-S)^6]\\
A_{53} &= (4S+105S^2+198S^3-1376S^4-1302S^5-117S^6+58S^7)\\
&\quad /[32\sinh(kD)(3+2S)(1-S)^6]\\
A_{55} &= (-6S^3+272S^4-1552S^5+852S^6+2029S^7+430S^8)\\
&\quad /[64\sinh(kD)(3+2S)(4+S)(1-S)^6]\\
B_{11} &= 1\\
B_{22} &= \coth(kd)(1+2S)/[2(1-S)]\\
B_{31} &= -3(1+3S+3S^2+2S^3)/[8(1-S)^3]\\
B_{33} &= -B_{31}\\
B_{42} &= \coth(kd)(6-26S-182S^2-204S^3-25S^4+26S^5/[6(3+2S)(1-S)^4])\\
B_{44} &= \coth(kd)(24+92S+122S^2+66S^3+67S^4+34S^5)/[24(3+2S)(1-S)^4]\\
B_{51} &= -(B_{53}+B_{55})\\
B_{53} &= 9(132+17S-2216S^2-5897S^3-6292S^4-2687S^5+194S^6+467S^7\\
&\quad +82S^8)/[128(3+2S)(4+S)(1-S)^6]\\
B_{55} &= 5(300+1579S+3176S^2+2949S^3+1188S^4+675S^5+1326S^6+827S^7\\
&\quad +130S^8)/[384(3+2S)(4+S)(1-S)^6]\\
C_0 &= [\tanh(kD)]^{1/2}\\
C_2 &= [\tanh(kD)]^{1/2}(2+7S^2)/[4(1-S)^2]\\
C_4 &= [\tanh(kD)]^{1/2}(4+32S-116S^2-400S^3-71S^4+146S^5)/[32(1-S)^5]\\
E_2 &= \tanh(kD)(2+2S+5S^2)/[4(1-S)^2]\\
E_4 &= \tanh(kD)(8+12S-152S^2-308S^3-42S^4+77S^5)/[32(1-S)^5]
\end{aligned}
$$

Where $S = sech(2kD)$.

For the present test case a depth of 0.5 m, an amplitude of 0.05 m and a wave length of 2.5 m have been chosen. Furthermore there will be no mean current in the flow. The Ursell number for this case is $2AL^2/d^3 = 5 \ll 32\pi^2/3$, which would mean that linear wave theory is applicable.



(a) Horizontal velocity at t = 0.73s

(b) Horizontal velocity at t = 1.35s

Figure 3.12: Propagation of a regular wave on a flat bottom with open boundaries given at 2 instances. A time varying velocity profile is prescribed on both vertical boundaries so they alternate between inlets and outlets.

However, a higher order solution is used, as phase difference can occur after several time periods. Finally, in the simulation the fluid viscosity ν is set to $10^{-6}\,\mathrm{m^2\,s^{-1}}$, the particle spacing Δr is set to one tenth of the amplitude and the numerical parameters are $c_0 = 20\,\mathrm{m\,s^{-1}}$ and $\Lambda = 0.1$. An illustration of this test case can be found in Figure 3.12 where it is shown that the open boundaries alternate between inlets and outlets.

A simulation will be run where the fluid velocities calculated from the analytical solution (Equations 3.51 - 3.57) will be imposed along the normal of the open boundaries, and Riemann invariants will be used to calculate the pressure. This will be compared to the analytical solution and a simulation with periodic conditions. Periodic conditions mean that particles exiting the domain on one side will enter the domain on the other side, but more importantly particles on one side will have as neighbours particles close to the other side, acting as if the domain was truly infinite. This will therefore serve as a reference, because ideal open boundary conditions would give the same values as an extension of the domain filled with fluid and wall particles. Furthermore, it should be noted that if both the pressure and the velocities would be imposed, then the problem would be over-constrained and particles would exit the domain.

Plots of the free surface are then presented in Figure 3.13. The first conclusion is that even after several wave periods the free surface stays very close to the analytical solution for all types of boundaries. Secondly, because of its nature, the simulation that uses periodic conditions needs the domain to be a multiple of the wave length. This is not the case with open boundary conditions (as can be seen from the fact that the green free surface particles extend over a longer range than the blue free surface particles). Finally, the waves propagating through a periodic domain are slightly slower than the analytical solution, which is not true for simulations with open boundary conditions as the analytical solution is imposed on the velocities.

Furthermore, let us define the error on the free surface using the following equation:

$$\varepsilon_\eta = \sqrt{\frac{1}{N_{\mathcal{FS}}} \sum_{a \in \mathcal{FS}} \frac{(\eta_{sim} - \eta)^2}{A^2}}, \tag{3.58}$$

where $\mathcal{FS}$ represents the free surface particles, $N_{\mathcal{FS}}$ is the number of free surface particles, η_{sim} is the free surface elevation of the simulations and η is the analytical solution given in Equation (3.51).

These errors are plotted over time in Figure 3.14, which shows that for these discretisations

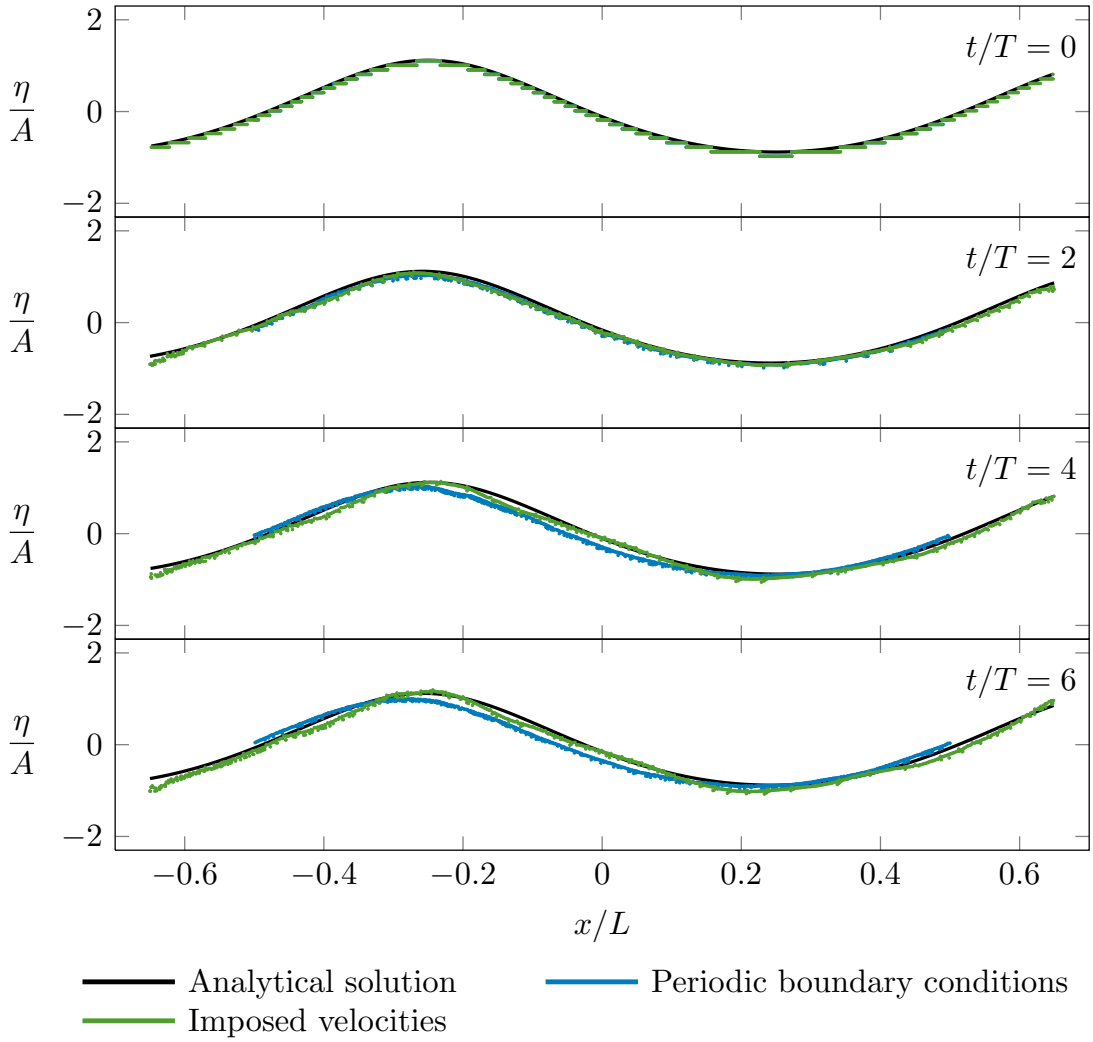


Figure 3.13: Free surface particles for simulations using different boundary conditions compared to the analytical solution of Equations (3.51) - (3.57).

the error (the solid line) is of the order of the particle spacing (the dashed line). Unfortunately, this is not the case for particle size smaller than shown in Figure 3.14 as an instability develops, increasing the error. These instabilities appear as a checker-boarding effect close to the inlets, and work on the volume diffusion term close to open boundaries will be necessary in the future to reduce those errors for very refined simulations.

In all cases the open boundary conditions seem to model properly the propagating waves, and no discrepancies appear between the fluid and imposed free surfaces even after seven wave periods.

3.4.4 2-D solitary wave

The Riemann invariants become particularly useful when the boundary conditions are not well known, or not easy to predict. To illustrate this concept, a test case for the propagation of a solitary wave on a bed with a step, that will cause it to break will be presented. See Figure 3.15.

At the inlet the solitary wave will be imposed from the Korteveg–De Vries solitary wave

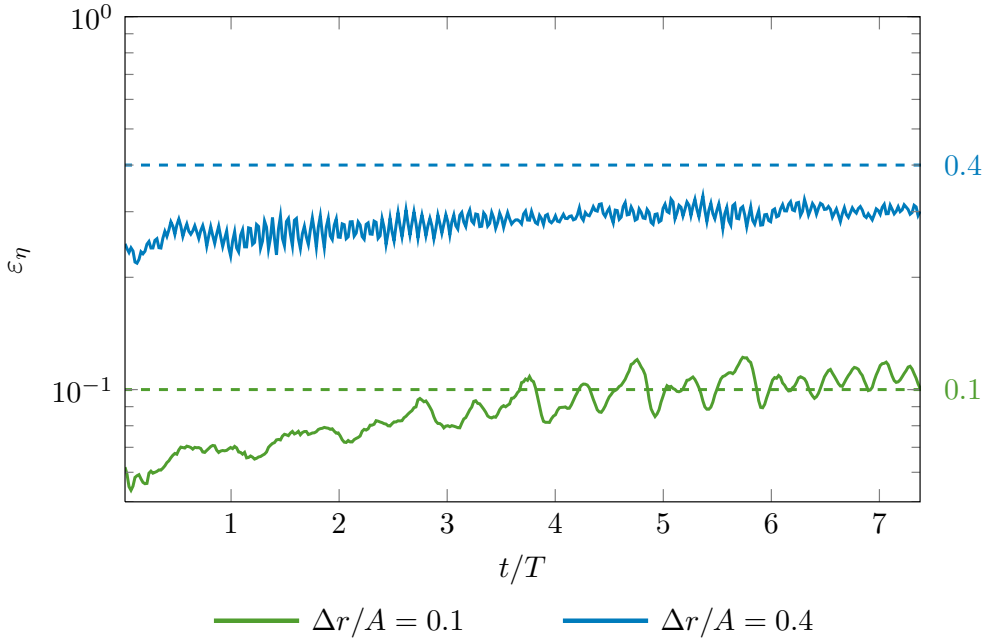


Figure 3.14: Errors on the free surface for simulations with an imposed velocity profile on the open boundaries for two particle spacings Δr. The dashed line shows the two values $\Delta r/A$.

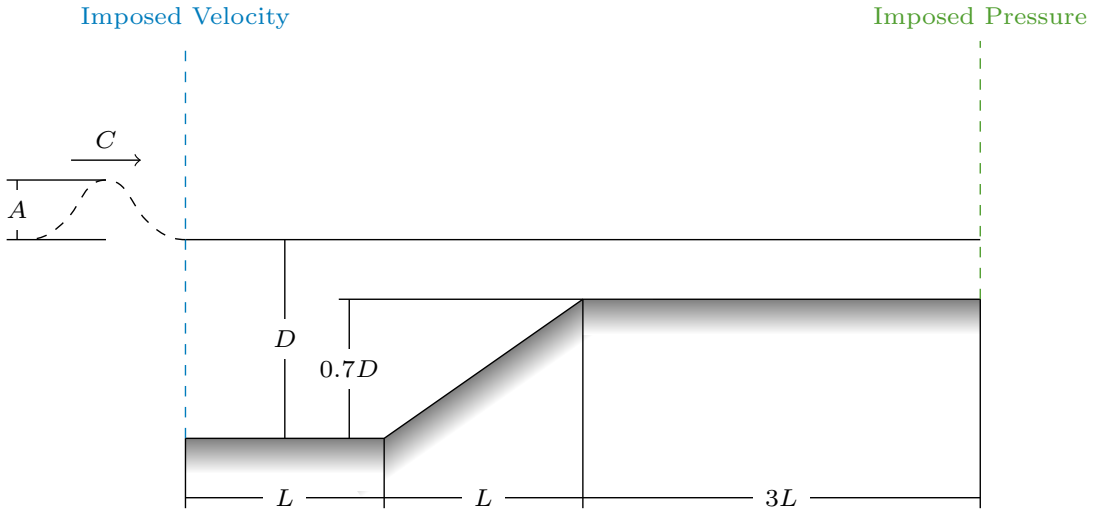


Figure 3.15: Configuration of the 2-D solitary wave on a sloped domain.

equation (see e.g. Dean and Dalrymple, 1991):

$$\eta(t) = A\,\mathrm{sech}^2\left(k(ct - x_0)\right), \tag{3.59a}$$
$$h(t) = D + \eta(t), \tag{3.59b}$$
$$p(z,t) = \rho g(z - h(t)), \tag{3.59c}$$
$$u_x(z,t) = c\frac{\eta(t)}{h(t)}, \tag{3.59d}$$
$$u_z(z,t) = \frac{z}{h(t)}\frac{\partial \eta}{\partial t}(t). \tag{3.59e}$$

R One should bear in mind that Korteveg–De Vries equation is only an approximation of the Euler equations. See e.g. Antonopoulos et al. (2017) for a valuable comparison of the models.

$$\eta(x,t) = A\,\mathrm{sech}^2\left(k(x - ct - x_0)\right), \tag{3.60}$$

where A is the amplitude and x_0 is the wave shifting length. The wave number and celerity are computed as $k = \sqrt{3A/4D^3}$, $c = \sqrt{g(A+D)}$.

At the outlet a hydrostatic pressure corresponding to the reference level D will be imposed and the velocities will be calculated using Riemann invariants.

The geometry chosen for the test case illustrated in Figure 3.15 has a reference depth $D = 0.6m$, a wave amplitude $A = D/2$ and a characteristic length $L = 2.5D$. The wave shifting length is set to ensure that the waves are not in the domain at the initialisation, i.e. $x_0 = -4k$.

The fluid viscosity ν is set to $10^{-6}\,\mathrm{m^2\,s^{-1}}$, the particle spacing will be set to one thirtieth of the amplitude ($\Delta r = A/30$) and the numerical parameters are $c_0 = 20\,\mathrm{m\,s^{-1}}$ and $\Lambda = 0.1$.

The evolution of the solitary wave within the domain at different instances is shown in Figure 3.16. It shows that the step causes the wave to break, creating a highly disturbed flow to exit the domain. Nonetheless, imposing a hydrostatic pressure and using Riemann invariants is enough for the wave to exit the domain without any visible reflection in WCSPH. This shows the high flexibility of these new open boundary conditions.

Finally, it should be noted that the small peak that appears in the last frame of Figure 3.16 appears because when the wave goes over the step, a portion of its energy is reflected back to the inlet, which in turn reflects it back towards the step. This last reflection occurs because after the wave has entered the domain the velocity is imposed as zero, making it essentially a wall.

3.5 Conclusion

The unified semi-analytical boundary condition have been extended to treat complex inlets and outlets without spurious shocks. This was done by allowing the mass of the vertices to vary according to the imposed flow rate, and then creating or deleting particles accordingly. Furthermore, it was necessary to take into account the imposed Eulerian velocity field in order to integrate properly in time the continuity equation, and thus avoid artificial density perturbations near the boundary. Using Riemann invariants it is possible to impose compatible density and velocity fields that can support both Neumann and Dirichlet boundary condition.

These new boundary conditions have been tested on several test cases. Firstly on a simple flow entering and leaving the domain in a non-orthogonal way, on periodic waves where the open boundaries were used as both inlets and outlets and on a solitary waves for which the outlet boundary conditions were unknown. In all cases this formulation of open boundaries in a smoothed particle hydrodynamics framework has proved to be very adaptive to complex confined and free-surface flows.

Finally, the extension to 3-D should be straightforward, but since the boundary elements will be more complex, greater care will be needed to allow fluid particles to enter and leave the domain without creating any artificial perturbations.

Acknowledgements

Most of the GPU fork development of GPUSPH, as well as some of the computations presented here, were run on the Nvidia device offered as Libersky price to Martin Ferrand by the SPHERIC Steering Committee in 2010. The authors are also grateful to Arno Mayrhofer for his help during this work.

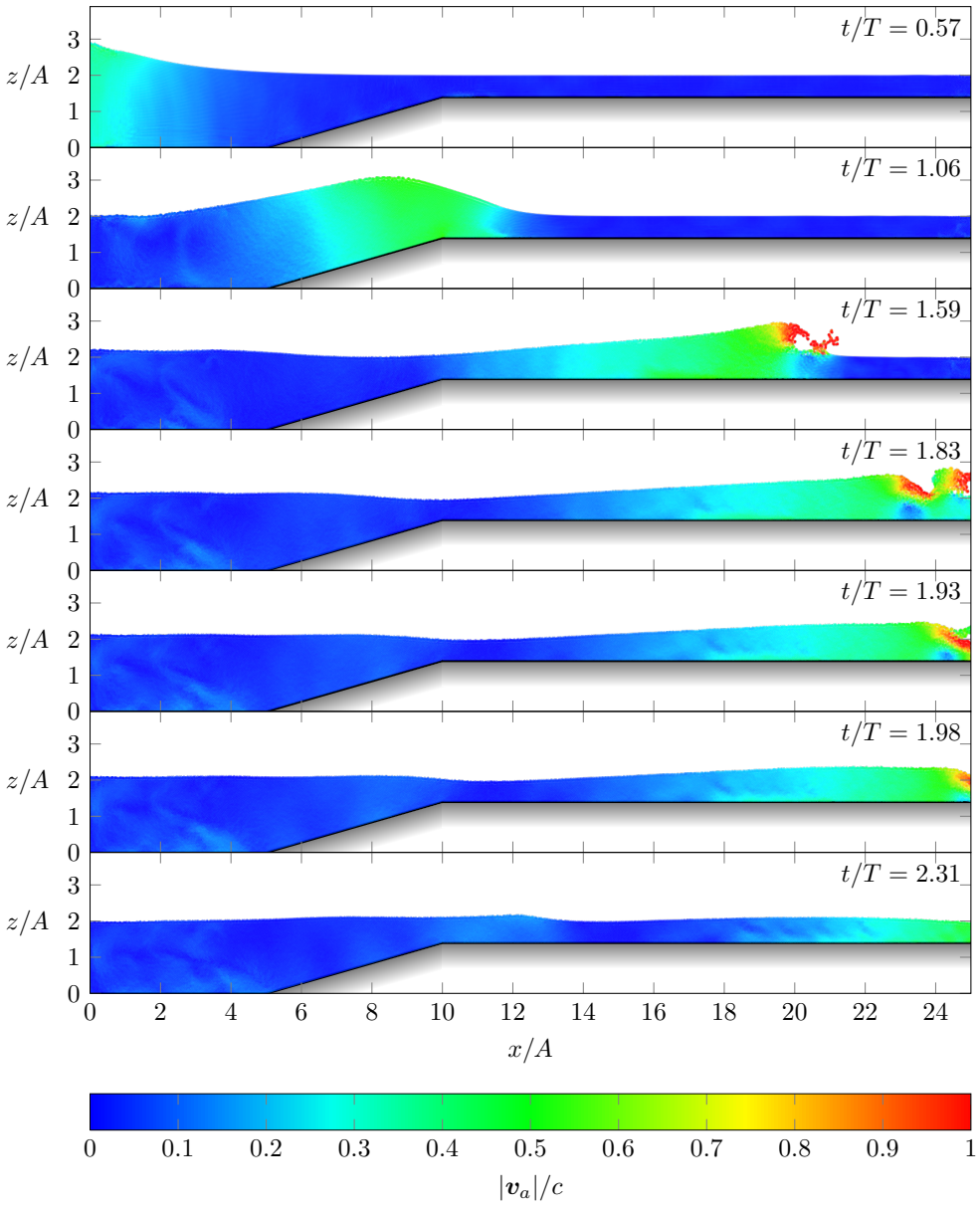


Figure 3.16: Propagation of a solitary wave on a bed with a step. As a reminder, the wave period is $T = 2\pi/(kc)$.

3.A The Riemann problem formulation

In the present approach, the mass flux through an open boundary is calculated at the position of the segments (although it is applied to the connected vertices), therefore the values of the pressure and velocity fields on those segments are required. Since the weakly Compressible SPH method is used, the boundary conditions can be derived using a linearised Riemann problem. This methodology is widely used in the literature for finite volumes, but it is not as common for SPH problems. The approach chosen here follows the work of Blondel et al. (2013), with simplifying assumptions of barotropic fluid and subsonic flow. The latter assumption is justified by the fact that weakly compressible SPH assumes the speed of sound to be significantly larger than the flow velocity.

Let us consider a segment s on an open boundary and work in the local reference frame of space dimension d relative to s, denoted by $(\boldsymbol{n}, \boldsymbol{t}_1, \cdots, \boldsymbol{t}_{d-1})$, $\boldsymbol{n}$ being the local normal unit vector and $\boldsymbol{t}_k$ the tangential unit vectors. In this Section, for simplicity we will drop the label s relative to all fields. The linearised Riemann problem (LRP) can now be written as:

$$\frac{\partial \boldsymbol{Y}}{\partial t} + \mathbf{B}\left(\boldsymbol{Y}\right) \frac{\partial \boldsymbol{Y}}{\partial n} = \mathbf{0}, \tag{3.61}$$

where the unknown vector is defined as:

$$\boldsymbol{Y} := \begin{pmatrix} \rho \\ u_n \\ u_{t_1} \\ \vdots \\ u_{t_{d-1}} \end{pmatrix} = \begin{pmatrix} \rho \\ \boldsymbol{u}\cdot\boldsymbol{n} \\ \boldsymbol{u}\cdot\boldsymbol{t}_1 \\ \vdots \\ \boldsymbol{u}\cdot\boldsymbol{t}_{d-1} \end{pmatrix}, \tag{3.62}$$

and the matrix $\mathbf{B}$ as:

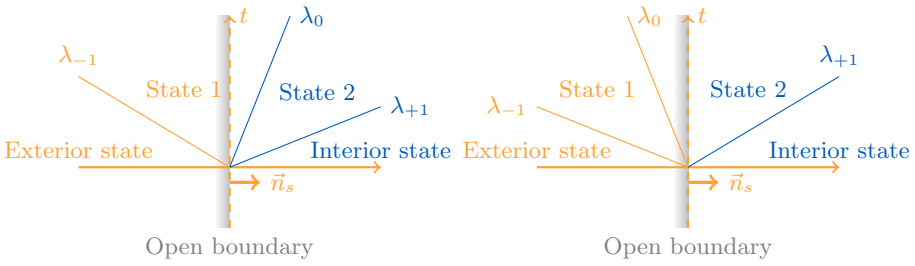
$$\mathbf{B}\left(\boldsymbol{Y}\right) := \begin{pmatrix} u_n & \rho & 0 & \cdots & 0 \\ \dfrac{c^2}{\rho} & u_n & 0 & \cdots & 0 \\ 0 & 0 & u_n & 0 & \cdots \\ \vdots & \vdots & 0 & \ddots & 0 \\ 0 & 0 & \cdots & 0 & u_n \end{pmatrix}. \tag{3.63}$$

Furthermore the speed of sound c is defined from the state Equation (3.1) as:

$$c := \sqrt{\frac{\partial p}{\partial \rho}} = c_0 \left(\frac{\rho}{\rho_0}\right)^{\frac{\xi-1}{2}}. \tag{3.64}$$

The eigenvalues of $\mathbf{B}$ are $\lambda_{-1} := u_n - c$, $\lambda_0 := u_n$ (with a multiplicity $d-1$) and $\lambda_{+1} := u_n + c$.

Along the open boundary there is a discontinuity between the exterior state (upon which we want to impose our values) and the interior state (i.e. the fluid domain). The eigenvalues



(a) Inlet. (b) Outlet.

Figure 3.17: Sketches of the Riemann problems.

of the problem, λ_i, represent the slope of the characteristic curves of the Riemann problem (see Figure 3.17), which are discontinuities between the states on their right and left sides (for example the wave λ_0 is a discontinuity between the data states $\mathbf{Y}_1$ and $\mathbf{Y}_2$). The characteristic waves can therefore be used to link the exterior state and the interior state. To go from one state to another the Generalized Riemann Invariants (GRI) will be used, as these define relations that hold true, for certain waves, across the wave structure. These are calculated from the relations defined by Jeffrey (1976):

$$\frac{\mathrm{d}Y_1}{V^{\lambda}_{r,1}} = \frac{\mathrm{d}Y_2}{V^{\lambda}_{r,2}} = ... = \frac{\mathrm{d}Y_d}{V^{\lambda}_{r,d}}, \tag{3.65}$$

where Y_i is a component of vector $\mathbf{Y}$ and $V^{\lambda}_{r,i}$ is a component of the right eigenvector of $\mathbf{B}$ associated to the eigenvalue λ.

The GRI corresponding to the celerities λ_{-1} and λ_{+1} are respectively denoted by R_{-1} and R_{+1}:

$$R_{-1} := u_n + \psi, \tag{3.66}$$
$$R_{+1} := u_n - \psi, \tag{3.67}$$

where:

$$\psi := \int_{\rho_0}^{\rho} \frac{c}{\tilde{\rho}} \mathrm{d}\tilde{\rho}, \tag{3.68}$$

i.e. $\psi = \dfrac{2c_0}{\xi - 1}\left(\dfrac{\rho}{\rho_0}\right)^{\frac{\xi-1}{2}}$ if $\xi > 1$, $\psi = c_0 \ln \dfrac{\rho}{\rho_0}$ if $\xi = 1$.

As displayed in Figure 3.17, the state of the segment value $\boldsymbol{Y}_s$ is either defined by the first state (see Figure 3.17a) or by the second state (see Figure 3.17b) which respectively correspond to $\lambda_0 = u_n \geq 0$ (ingoing mass flux) and $\lambda_0 = u_n < 0$ (outgoing mass flux). One should bear in mind that λ_{-1} is supposed to be always negative and λ_{+1} always positive as the flow is to be subsonic.

Therefore to link the exterior state a first assumption is made, that the wave λ_{-1} is a "ghost" wave, as it is outside of the domain. This means that the data $\mathbf{Y}_{ext}$ of the exterior state is assumed equal to the data $\mathbf{Y}_1$ of state 1.

Wave λ_0 is known as a contact discontinuity, i.e. $\lambda_{0,1} = \lambda_{0,2}$ Toro (2009). This implies that $u_{n,1} = u_{n,2}$, which is consistent with the GRI relations. Through these relations it is also

considered that $\mathrm{d}\rho = 0$ across λ_0. Therefore $\rho_1 = \rho_2$, but also $p_1 = p_2$ as the pressure is defined by Equation (3.1).

Therefore to link the fluid velocity along the normal of the segment and the pressure, one needs only to find a relation between state 2 and the interior state; i.e. across the characteristic wave λ_{+1} (the tangential velocities will be dealt with later on). In terms of notation, a distinction will be made between this characteristic wave within state 2, $\lambda_{+1,2}$, and its value within the interior state, $\lambda_{+1,int}$. As defined in Toro (2009) the characteristic wave λ_{+1} can belong to three possible types of discontinuity:

1. **Contact wave**
   In this case the characteristics are parallel in both states, this means that the following condition is true:

   $$\lambda_{+1,2} = \lambda_{+1,int}, \tag{3.69}$$

   and the GRI relations hold across the characteristic wave:

   $$R_{+1,2} = R_{+1,int}. \tag{3.70}$$

2. **Expansion (or rarefaction) wave**
   In this case the characteristics are diverging, and the two data states are connected through a *smooth transition*. This case is defined by the following condition:

   $$\lambda_{+1,2} < \lambda_{+1,int}, \tag{3.71}$$

   and the GRI relations hold across the characteristic wave:

   $$R_{+1,2} = R_{+1,int}. \tag{3.72}$$

3. **Shock wave**
   In this case the characteristics are converging towards the wave λ_{+1}, meaning that this case is defined by the following condition:

   $$\lambda_{+1,2} > \lambda_{+1,int}. \tag{3.73}$$

   In this case the GRI relations do not hold across the characteristic wave, and therefore the Rankine-Hugoniot relationships have to be used:

   $$\begin{aligned} \text{Conservation of mass:} \quad & \sigma\left[\rho_2 - \rho_{int}\right] = \rho_2 u_{n,2} - \rho_{int} u_{n,int}, \\ \text{Conservation of momentum:} \quad & \sigma\left[\rho_2 u_{n,2} - \rho_{int} u_{n,int}\right] = p_2 + \rho_2 u_{n,2}^2 - p_{int} - \rho_{int} u_{n,int}^2, \end{aligned} \tag{3.74}$$

   where σ is the shock speed to be computed. Combining Equations (3.74), one gets $\rho_2\rho_{int}\left[u_{n,2} - u_{n,int}\right]^2 = \left[\rho_2 - \rho_{int}\right]\left[p_2 - p_{int}\right]$. If ρ_2 is known, then $u_{n,2}$ can be determined directly through this relation. Otherwise, one may iterate to get ρ_2 from a known value of $u_{n,2}$.

   Note that the published version assumed an infinite value for the shock speed σ.

For the case of the tangential velocities the Riemann invariants relations make it possible to write $\mathrm{d}u_{t_1} = ... = \mathrm{d}u_{t_{d-1}} = 0$ across the wave λ_{+1} (we will assume these relations still hold in the case of a shock). Therefore the tangential velocities are equal between state 2 and the interior state. However, the wave λ_0 acts as a discontinuity.

Therefore if the problem is that of an inlet then the discontinuity prevents a relationship between state 2 and state 1 to be defined for these velocities, all that is known is that $\mathrm{d}u_{t_1} = ... = \mathrm{d}u_{t_{d-1}} = Const$. This means that these velocities need to be defined by the user for inlet open boundaries. However for the case of an outlet then the λ_0 will be assumed to be a "ghost" wave and the exterior tangential velocities will be assumed to be equal to the tangential velocities of state 1 and 2, and therefore equal to the velocities of the interior state.

In mesh-based methods, such as finite volumes, knowing the interior state data is simple as boundary segments are only connected to one element. However in the SPH formulation there is more than one fluid particle connected to a boundary segment. Therefore defining the interior state is non-trivial. The method chosen is to use the SPH interpolations as defined by Ferrand et al. (2013):

$$p_{s,int} = \frac{1}{\alpha_s} \sum_{b \in \mathcal{F}} V_b p_b w_{bs}, \tag{3.75}$$

where, as a reminder, $\mathcal{F}$ contains only fluid particles and α_s is the Shepard filter defined for a segment s by:

$$\alpha_s = \sum_{b \in \mathcal{F}} V_b w_{bs}. \tag{3.76}$$

Similarly, the interior velocity is defined as:

$$\boldsymbol{u}_{s,int} = \frac{1}{\alpha_s} \sum_{b \in \mathcal{F}} V_b \boldsymbol{u}_b w_{bs}. \tag{3.77}$$

Finally, the density will be calculated using Equation (3.1).

3.B Handling varying number of particles in a parallel implementation

Handling a varying number of particles in a parallel implementation can be difficult in a parallel implementation. To allow the algorithm presented here to be easily implemented, several key numerical considerations will be described in the following Sections.

3.B.1 Creating and deleting particles

As mentioned earlier this algorithm has been implemented in a parallel framework, and therefore particle creation/deletion can be non-trivial. Namely, adding an extra particle needs to be an atomic operation (i.e. an operation that will be done sequentially on all processors). The same is true when deleting particles and the mass of the concerned vertex particles.

The implementation tested is based on a branch of the GPUSPH software Hérault et al. (2010, 2014), and therefore most of the memory is handled in the same manner. In addition to the aforementioned problem, all the arrays are of a fixed size at the initialisation. As such, changing the size of any array when a particle is removed or added can be quite expensive, especially when it happens at each iteration. Thus it is suggested to allocate larger arrays. The user has to evaluate

the total number of particles that will be added during the simulation time. Regardless of this number, the loops are only done on the current number of particles at each iteration.

Furthermore it is very important that the masses of the open boundary vertices is kept constant during the stepping algorithm, so that the condition $M_b = M_b^n$ during the open time interval $]t^n, t^{n+1}[$ is maintained (i.e. excluding both t^n and t^{n+1}). The same is true for the set of particles, i.e. during the continuity equation step, the particles that have crossed an open boundary are considered, but not the particles that will be created.

3.B.2 Updating vertices and segments

When dealing with the semi-analytical boundary conditions, it has been established that Dirichlet boundary conditions will be imposed on the vertices, and Neumann boundary conditions on the segments Leroy et al. (2015). This implies that for walls the velocity will be imposed on the vertices, and that velocity on the segments will be the average of the connected vertices. For the pressure the condition $\partial p/\partial n = 0$ will be used to calculate the pressure on the segments, and on the vertices it will be the average of the connected segments.

When dealing with an open boundary, either the velocity or the pressure will be imposed on the vertices and Riemann conditions will be used to calculate the corresponding pressure or velocity on the segments. In all cases the missing values on the segments and vertices will be calculated from the connected vertices or segments (see Sections 3.3.4 and 3.3.4). Furthermore the mass flux will only affect the masses of vertices, even though it uses the densities, velocities and lengths of the connected segments.

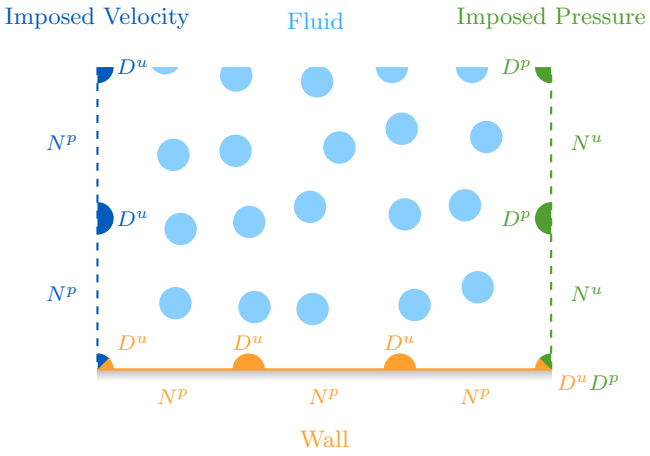


Figure 3.18: Illustration of the different boundary conditions that can be imposed on vertices and segments. The symbol D represents a Dirichlet boundary condition and the symbol N represents a Neumann boundary condition. The superscripts show if these conditions are imposed on the velocity u or on the pressure p.

However, as illustrated by Figure 3.18, dealing with the vertices that are in-between a wall and an open boundary segment is not trivial. For the vertices that are in-between a segment with an imposed velocity and a wall, it is recommended to impose the wall velocity, and to deduce the pressure from the connected segments. For the vertices that are in-between a segment with imposed pressure and a wall, it is recommended to impose the wall velocity and the open boundary pressure. Furthermore, it is recommended to keep the masses of the vertices that are in-between wall segments and open boundaries constant. Therefore, even though these particles will be used to calculate the mass repartition factor β their masses will not evolve. This implies that the open boundary vertices can be considered as an interpolation of the Eulerian fields

imposed.

This means that in the implementation 4 loops would be necessary:

1. Calculate the values that are imposed
2. Update the vertices and segments according to the connected segments or vertices
3. Apply the marching scheme on the fluid particles
4. Update the masses of the open boundary vertices

Furthermore, it is good practice to impose the velocities on the inlets and to impose the pressure on the outlets.

3.B.3 Mass of the vertices

For open boundaries segments it is strongly recommended that their masses stay between $\pm M_{\text{ref}}$. For the vertices that are slightly above the free-surface or near walls, the SPH interpolation can create mass fluxes that will break the condition $-M_0/2 < M_v < M_0/2$. This is particularly critical at the outlets, and therefore it is recommended to clip the masses and for open boundaries with imposed pressures, the mass flux above the free surface is only allowed to be negative. Finally, above the free surface the mass is initialised at 0.

3.B.4 Constraints on the time-step

The marching scheme uses an adaptive time-stepping, where the time-step is calculated using the following condition:

$$\Delta t \leq \frac{0.4h}{c_0}. \tag{3.78}$$

And c_0 is chosen so that the following rule is always true:

$$c_0 \geq 10 \max\left(|\boldsymbol{u}_{\max}|, \sqrt{gD}\right), \tag{3.79}$$

where h is the smoothing length of the kernel w, D is the water depth (in case of free-surface flows) and $|\boldsymbol{u}_{\max}|$ is the norm of the maximum fluid velocity in the domain (i.e. for the fluid particles, but also for the open boundaries).

In addition, since γ_a is calculated through its gradient (see the marching scheme presented in Equations (3.27)), this gives an additional constraint on the time step:

$$\Delta t \leq 0.004 \frac{1}{\max\limits_{a \in \mathcal{F}} \left\{ \max\limits_{s \in \mathcal{S}} \left[|\boldsymbol{\nabla} \gamma_{as} \cdot (\boldsymbol{u}_{as} - \boldsymbol{v}_{as})| \right] \right\}}. \tag{3.80}$$

Additional details about the numerical stability of WCSPH are available in Violeau and Leroy (2014).

Bibliography

Antonopoulos, D., Dougalis, V., Mitsotakis, D., 2017. Error estimates for galerkin approximations of the serre equations. SIAM Journal on Numerical Analysis 55. doi:10.1137/16M1078355.

Archambeau, F., Méchitoua, N., Sakiz, M., 2004. Code_saturne: a finite volume method for the computation of turbulent incompressible flows - industrial applications. International Journal on Finite Volumes 1, 1–62.

Blondel, F., Audebert, B., Pasutto, T., Stanciu, M., 2013. Condensation models and boundary conditions for non-equilibrium wet steam flows. International Journal on Finite Volumes 10, 1–53.

Bouscasse, B., Marrone, S., Colagrossi, A., Di Mascio, A., 2013. Multi-purpose interfaces for coupling SPH with other solvers, in: 8th SPHERIC International Workshop.

Braun, S., Wieth, L., Koch, R., Bauer, H.J., 2015. A Framework for Permeable Boundary Conditions in SPH: Inlet, Outlet, Periodicity, in: 10th SPHERIC International Workshop.

Brezzi, F., Pitkäranta, J., 1984. On the Stabilization of Finite Element Approximations of the Stokes Equations, in: Hackbusch, W. (Ed.), Efficient Solutions of Elliptic Systems. Vieweg+Teubner Verlag. volume 10 of *Notes on Numerical Fluid Mechanics*, pp. 11–19. URL: http://dx.doi.org/10.1007/978-3-663-14169-3_2, doi:10.1007/978-3-663-14169-3_2.

Dean, R.G., Dalrymple, R.A., 1991. Water wave mechanics for engineers and scientists. volume 2 of *Advanced Series on Ocean Engineering*. World Scientific, Singapore.

Fatehi, R., Manzari, M.T., 2011. A remedy for numerical oscillations in weakly compressible smoothed particle hydrodynamics. International Journal for Numerical Methods in Fluids 67, 1100–1114.

Fenton, J.D., 1985. A fith-order Stokes theory for steady waves. Journal of Waterway, Port, Coastal and Ocean Engineering 111, 216–234.

Ferrand, M., Laurence, D.R., Rogers, B.D., Violeau, D., Kassiotis, C., 2013. Unified semi-analytical wall boundary conditions for inviscid, laminar or turbulent flows in the meshless SPH method. International Journal for Numerical Methods in Fluids 71, 446–472. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.3666, doi:10.1002/fld.3666, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.3666.

Ferrari, A., Dumbser, M., Toro, E.F., Armanini, A., 2009. A new 3D parallel SPH scheme for free surface flows. Computers & Fluids 38, 1203–1217.

Fonty, T., 2019. Modélisation de l'entraînement d'air dans la méthode SPH .

Fonty, T., Ferrand, M., Leroy, A., Violeau, D., 2020. Air Entrainment Modeling in the SPH Method: A Two-Phase Mixture Formulation with Open Boundaries. Flow, Turbulence and Combustion , 1573–1987doi:10.1007/s10494-020-00165-7.

Ghaitanellis, A., Violeau, D., Ferrand, M., Leroy, A., Joly, A., 2015. Application of the unified semi-analytical wall boundary conditions to multi-phase SPH, in: 10th SPHERIC International Workshop.

Hammad, K.J., Ötügen, M.V., Arik, E.B., 1999. A PIV study of the laminar axisymmetric sudden expansion flow. Experiments in Fluids 26, 266–272.

Hérard, J.M., Martin, X., Preprint: hal-01166478. An integral approach to compute compressible fluid flows in domains containing obstacles. International Journal on Finite Volumes .

Hérault, A., Bilotta, G., Dalrymple, R.A., 2010. SPH on GPU with CUDA. Journal of Hydraulic Research 48, 74–79.

Hérault, A., Bilotta, G., Dalrymple, R.A., 2014. Achieving the best accuracy in an SPH implementation, in: 9th SPHERIC International Workshop.

Jeffrey, A., 1976. Quasilinear Hyperbolic Systems and Waves. Pitman, London.

Kassiotis, C., Ferrand, M., Violeau, D., Rogers, B.D., Stansby, P.K., Benoit, M., et al., 2011. Coupling SPH with a 1-D Boussinesq-type wave model, in: 6th International SPHERIC Workshop, pp. 241–247.

Kulasegaram, S., Bonet, J., Lewis, R.W., Profit, M., 2004. A variational formulation based contact algorithm for rigid boundaries in two-dimensional SPH applications. Computational Mechanics 33, 316–325.

Kunz, P., Hirschler, M., Huber, M., Nieken, U., 2016. Inflow/outflow with dirichlet boundary conditions for pressure in isph. Journal of Computational Physics 326, 171–187.

Lastiwka, M., Basa, M., Quinlan, N.J., 2009. Permeable and non-reflecting boundary conditions in SPH. International Journal for Numerical Methods inf Fluids , 709–724.

Leroy, A., 2014. A new incompressible SPH model : towards industrial applications. Modeling and Simulation. Ph.D. thesis. Universitée Paris-Est.

Leroy, A., Violeau, D., Ferrand, M., Fratter, L., Joly, A., 2016. A new open boundary formulation for incompressible SPH. Computers & Mathematics with Applications 72, 2417 – 2432. URL: http://www.sciencedirect.com/science/article/pii/S0898122116305107, doi:https://doi.org/10.1016/j.camwa.2016.09.008.

Leroy, A., Violeau, D., Ferrand, M., Joly, A., 2015. Buoyancy modelling with incompressible SPH for laminar and turbulent flows. International Journal for Numerical Methods in Fluids 78, 455–474. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.4025, doi:10.1002/fld.4025, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.4025.

Leroy, A., Violeau, D., Ferrand, M., Kassiotis, C., 2014. Unified semi-analytical wall boundary conditions applied to 2-D incompressible SPH. Journal of Computational Physics 261, 106 – 129. URL: http://www.sciencedirect.com/science/article/pii/S0021999113008437, doi:https://doi.org/10.1016/j.jcp.2013.12.035.

Mahmood, O., Kassiotis, C., Violeau, D., Rogers, B.D., Ferrand, M., 2012. Absorbing inlet/outlet boundary conditions for SPH 2-D turbulent free-surface flows, in: 7th SPHERIC International Workshop.

Mayrhofer, A., Rogers, B.D., Violeau, D., Ferrand, M., 2012. Study of differential operators in the context of the semi-analytical wall boundary conditions, in: 7th SPHERIC International Workshop, pp. 149–156.

Monaghan, J.J., 2005. Smoothed Particle Hydrodynamics. Reports on Progress in Physics 68, 1703–1759.

Tait, P.G., 1888. Report on some of the physical properties of fresh water and sea water. Report on the Voyage of Challenger, Physics and Chemistry 2, 1–76.

Toro, E.F., 2009. Riemann solvers and numerical methods for fluid dynamics. Springer, London.

Vacondio, R., Altomare, C., De Leffe, M., Hu, X., Le Touzé, D., Lind, S., Marongiu, J.C., Marrone, S., Rogers, B.D., Souto-Iglesias, A., 2021. Grand challenges for smoothed particle hydrodynamics numerical schemes. Computational Particle Mechanics 8, 575–588.

Vacondio, R., Rogers, B.D., Stansby, P.K., Mignosa, P., 2011. SPH modeling of shallow flow with open boundaries for practical flood simulation. Journal of Hydraulic Engineering 138, 530–541.

Violeau, D., Leroy, A., 2014. On the maximum time step in weakly compressible SPH. Journal of Computational Physics 256, 388–415.

Wendland, H., 1995. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. Advances in Computational Mathematics 4, 1414–1421.

4. Finite volume arbitrary Lagrangian-Eulerian schemes using dual meshes

Abstract 4.1 For reasons of efficiency and accuracy, water wave propagation is often simulated with potential or inviscid models rather than Navier–Stokes solvers, but for wave-induced flows, such as wave-structure interaction, viscous effects are important under certain conditions. Alternatively, general purpose Navier–Stokes (CFD) models can have limitations when applied to such free-surface problems when dealing with large amplitude waves, run-up, or propagation over long distances. Here we present an arbitrary Lagrangian-Eulerian (ALE) algorithm with special care to the time-stepping and boundary conditions used for the free-surfaces, integrated into code_saturne, and we test its capabilities for modelling a variety of water wave generation and propagation benchmarks, and finally consider interaction with a vertical cylinder. Two variants of the mesh displacement computation are proposed and tested against the discrete geometric conservation law (GCL). The more robust variant, for highly curved or sawtoothed free-surfaces, uses a compatible discrete operator scheme on the dual mesh for solving the mesh displacement, which makes the algorithm valid for any polyhedral mesh. Results for standard wave propagation benchmarks for both variants show that, when care is taken to avoid grids with excessive numerical dissipation, this approach is effective at reproducing wave profiles as well as forces on bodies.

This chapter is a reproduction of Ferrand and Harris (2021) with additional results.

Résumé 4.1 Pour des raisons de performance, la propagation des vagues est souvent simulée en utilisant la théorie des écoulements potentiels, non visqueux, plutôt qu'en utilisant des solveurs Navier–Stokes. Par contre, pour les écoulements induits par les vagues, comme les interactions vagues/structures, les effets visqueux peuvent être importants. Les solveurs Navier–Stokes (*CFD*) généralistes, quant à eux, peuvent avoir des difficultés pour simuler des problèmes à surface libre avec des vagues de grande amplitude, des vagues qui montent sur une plage ou encore leurs propagations sur de longues distances. Nous présentons ici un algorithme arbitrairement lagrangien eulérien (*ALE*) intégré au logiciel code_saturne pour lequel nous détaillons avec attention le schéma en temps et les conditions aux limites de surface libre, et nous testons ses capacités sur une variété de cas tests de génération et de propagation de vagues, ainsi que l'interaction d'une vague avec un cylindre. Deux variantes

pour calculer le déplacement des nœuds du maillage sont présentées et testées (notamment la loi de conservation géométrique discrète (*GCL*)). La version la plus robuste, notamment sur les configurations avec courbure importante ou avec une surface libre en dents de scie, est celle faisant appel aux schémas des opérateurs discrets compatibles (*CDO*) qui résolvent le déplacement du maillage sur un maillage dual (à celui utilisé pour résoudre le fluide). Les résultats démontrent que cette approche est capable de reproduire de façon satisfaisante les profils des vagues et les efforts induits sur les structures.

Ce chapitre est une reproduction de Ferrand and Harris (2021) auquel s'ajoute quelques résultats.

4.1 Introduction

There are many applications to models of water waves, from understanding tsunami hazards, to computing the forces on offshore structures, to understanding the performance of different ship designs. Similarly, there are many different methods for modelling water waves; a recent summary was given by Lin (2008). Due to the vast horizontal scales required to study ocean waves, often a simplified approach is used, such as potential flow, with either nonlinear or linearised free-surface boundary condition, or a depth-averaged approach, such as shallow water or Boussinesq equations. Such specialised models efficiently propagate waves over long distances without excessive numerical dissipation.

A direct approach to the physics would be to solve the full three-dimensional (3-D) Navier–Stokes equations, in a domain covering both the water and air phases. In fact, some of the first water wave models solved the Navier–Stokes equations in this way Chan and Street (1970), using the Marker And Cell (MAC) method (Harlow and Welch, 1965). Many different methods for tracking the free-surface have been developed, but variations of the Volume-Of-Fluid (VOF) technique (Hirt and Nichols, 1981) have become quite popular for coastal engineering, using it for wave modelling (e.g., Jacobsen et al., 2012) and wave-structure interaction (e.g., Chen et al., 2014), as well as many others applications. Such approaches have the advantage of being able to treat breaking waves, which is an important but complex issue ((e.g., Helluy, Philippe et al., 2005)). One major drawback is that they require finer computational domains and tracking the interface without diffusion is a challenge. For a recent discussion of wave propagation benchmarks for VOF codes, see Wroniszewski et al. (2014).

Alternatively, one can use a boundary-fitted domain, and move the computational domain at each time-step (Hirt et al., 1974). Such an arbitrary Lagrangian-Eulerian (ALE) approach will be considered here. This is used in many ocean models (e.g., Telemac3D; Hervouet, 2007), or non-hydrostatic wave models Ma et al. (2012), and these are able to solve such diverse applications as landslide-generated tsunamis and surf-zone wave propagation. Many models that use a boundary-fitted domain, however, are based on sigma-coordinate grids in the vertical, which can have some limitations for dealing with complex grids or immersed bodies. These limitations are removed with the present approach, where a more general mesh movement is solved with specific free-surface condition for the mesh displacement. Although ALE cannot model the details of overturning waves like a VOF approach, such non-hydrostatic wave models are able to incorporate energy dissipation in order to represent breaking effects Zijlema and Stelling (2008), and examples of this will be shown below.

In the context of ALE solvers, there have been many developments (see e.g., Donea et al. (2004)). The sigma-coordinate grid mentioned above corresponds to a special case of the simplest mesh regularisation approach, that of a transfinite mapping. To apply to more general meshes, a common approach that will be discussed further below is a Laplacian smoothing approach (e.g., Benson (1989)) where a Laplace or Poisson equation is solved for the mesh velocity in the interior of the domain. One can consider other node solvers, and there exist more complex, high-order shock-capturing schemes (e.g., Boscheri et al. (2014)). As well, general ALE solvers often consider adaptive meshes, with a remeshing of the domain during the simulation, in order to handle large mesh displacements (e.g., Re et al. (2017)). Such developments however can be time-consuming (due to additional equations to solve), and are unnecessary for ocean wave problems, as we consider waves that are not overturning. As a result, we consider only the essential aspects of integrating an ALE scheme into an existing Navier–Stokes solver and the resulting performance for ocean wave applications. For this, it is important to consider the discrete geometric conservation law (GCL), which states that the volume change of a cell during a timestep should be equal to the volume swept by the cell boundaries during that timestep. The GCL is critical to ensure that the algorithm is robust (e.g., conserves mass and energy) for long

time simulations (e.g. Thomas and Lombard, 1979; Zhang et al., 1993; Farhat et al., 2001; Storti et al., 2012).

As a result, we consider below the numerical details required to develop a robust ALE finite volume algorithm integrated in the open-source CFD solver code_saturne Archambeau et al. (2004) for version 6.1 and newer, with particular attention to the discrete GCL. code_saturne uses collocated finite volumes to solve the Navier–Stokes, Reynolds Averaged Navier–Stokes (RANS), or Large Eddy Simulation (LES) equations, and has been used extensively in the industry, from nuclear thermal hydraulic application (see Gauffre et al., 2020, for a recent application), atmospheric modelling (e.g. Bahlali et al., 2020), electric arc modelling (e.g. Chemartin et al., 2009), fire modelling (Ma et al., 2019), two-phase dispersed flows using hybrid Moment/PDF approach (see Minier, 2016, for a valuable description of this method), ventilation, combustion (e.g., coal, gas, biomass, Dal Secco et al. (2015)) and has also been used previously for wave-current interaction (e.g. Teles et al., 2013; Benoit et al., 2018).

The Chapter is organised as follows. Before passing to the specific case of incompressible flows of constant density used in all applications here, in Section 4.2, we present the governing equations of the finite volume space and time discretisation for fixed meshes in terms that are valid for both incompressible or compressible flows. Then in Section 4.3 we present the ALE algorithm and specify the free-surface boundary condition, with special care on the time stepping scheme to conserve uniform velocity fields. The original use of cell-based, for fluid quantities, and vertex-based solvers (on the dual barycentric mesh), for the mesh displacement, is also presented, and this improves on volume conservation. In order to facilitate very steep waves, a filtering of spurious waves of length smaller than the mesh resolution is also introduced. Finally, in Section 4.4, we consider numerous validation cases. This is first done with flow through a valve, forced by a piston, showing that the velocity field remains accurate with a moving mesh, then by looking at standing waves of a range of amplitudes to verify volume conservation and model stability, a critical point for moving mesh algorithms. Then, satisfied that the model is robust, we consider a number of different applications, including viscous damping of standing waves, solitary wave propagation, wave propagation over a submerged bar, wave forces on a monopile, and the modelling of a novel laboratory setup for long wave generation. This ensemble of test cases demonstrates the ability of this approach to handle a wide variety of test cases of interest to the ocean modelling community.

4.2 Methodology

4.2.1 Governing equations

To describe the governing equations, we begin with the conservative Cauchy momentum equations written in vector notation:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \boldsymbol{u}) &= \dot{m}, \\ \frac{\partial}{\partial t}(\rho \boldsymbol{u}) + \mathbf{div}(\boldsymbol{u} \otimes \rho\, \boldsymbol{u}) &= \mathbf{div}(\boldsymbol{\sigma}) + \boldsymbol{f} + \dot{m}\boldsymbol{u}^{in}, \end{aligned} \tag{4.1}$$

where $\boldsymbol{u}$ represents the fluid velocity, ρ is the density of the fluid, $\boldsymbol{\sigma}$ is the stress field, $\boldsymbol{f}$ is a field of source terms (e.g., gravity, Coriolis, additional head losses), $\dot{m}$ a possible mass source, and $\boldsymbol{u}^{in}$ the velocity of a possible injected fluid. For sake of simplicity, $\boldsymbol{f} = \rho \boldsymbol{g}$ and $\dot{m}$ will be set to zero hereafter.

The stress tensor $\boldsymbol{\sigma}$ is decomposed into an isotropic component, called the pressure p, and

the deviatoric component, $\boldsymbol{\tau}$:

$$\boldsymbol{\sigma} = -p\mathbf{1} + \boldsymbol{\tau}, \tag{4.2}$$

and thus the Cauchy equations are:

$$\begin{aligned} &\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \boldsymbol{u}) = 0, \\ &\frac{\partial}{\partial t}(\rho \boldsymbol{u}) + \mathbf{div}(\boldsymbol{u} \otimes \rho\, \boldsymbol{u}) = -\boldsymbol{\nabla} p + \mathbf{div}\,(\boldsymbol{\tau}). \end{aligned} \tag{4.3}$$

For a Newtonian fluid, the constitutive law connecting the stress tensor to the deviatoric part of the deformation rate tensor, given as $\boldsymbol{S} := \frac{1}{2}\left(\boldsymbol{\nabla}\boldsymbol{u} + \boldsymbol{\nabla}\boldsymbol{u}^T\right)$, is determined as:

$$\boldsymbol{\tau}(\boldsymbol{u}) = \mu\left(\boldsymbol{\nabla}\boldsymbol{u} + \boldsymbol{\nabla}\boldsymbol{u}^T\right) + \left(\kappa - \frac{2}{3}\mu\right)\operatorname{tr}(\boldsymbol{\nabla}\boldsymbol{u})\,\mathbf{1}, \tag{4.4}$$

where κ is the volume viscosity, usually neglected and therefore omitted hereafter in this Chapter.

The combination of terms in the Cauchy momentum equations, Equations (4.1), and a Newtonian fluid, Equation (4.4), gives the Navier–Stokes equations.

When ρ is considered constant, flow is incompressible and the mathematical model is complete, otherwise the system should be supplemented with an energy equation and an equation of state (EOS) between density, pressure and energy must be specified (see e.g. Colas et al., 2019). Turbulent flows can also be considered subtracting the Reynolds stresses (for the RANS approach) or the sub-grid stresses (for the LES approach) from the Cauchy stresses. In this Chapter, only incompressible constant density flows will be considered.

4.2.2 Space and time discretisation

In this section, the time discretisation is detailed on a motionless meshed domain, and some insight of the space discretisation are given. A finite volume scheme where the velocity and pressure are stored at the cell barycentres, $\boldsymbol{x}_c$ (see Fig. 4.1), is used.

Integrating conservative Equations (4.3) both in space and time leads to the definition of the extensive quantities:

$$\begin{aligned} V_c &= \int_{V_c} \mathrm{d}\Omega, \\ M_c &= \int_{V_c} \rho\,\mathrm{d}\Omega, \\ \boldsymbol{Q}_c &= \int_{V_c} \rho\boldsymbol{u}\,\mathrm{d}\Omega, \end{aligned} \tag{4.5}$$

for cell volume V_c, cell mass M_c, and cell momentum $\boldsymbol{Q}_c$. As for the intensive quantities, let ρ_c be considered the cell-averaged density, and let $\boldsymbol{u}_c$ be the density-weighted cell-averaged velocity, defined as:

$$\begin{aligned} \rho_c &= \frac{M_c}{V_c}, \\ \boldsymbol{u}_c &= \frac{\boldsymbol{Q}_c}{M_c}. \end{aligned} \tag{4.6}$$

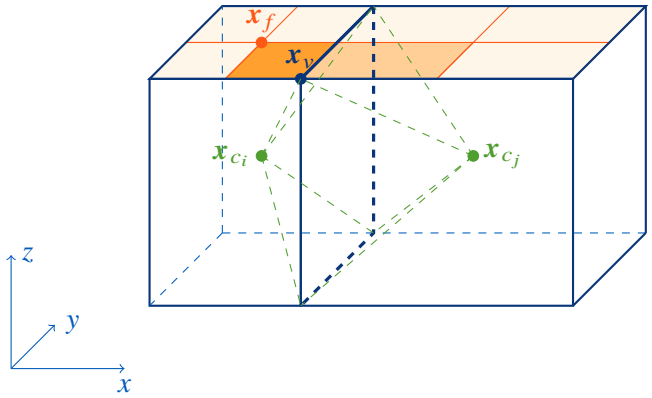


Figure 4.1: Labelling of information used for a mesh. In green, $\boldsymbol{x}_{c_i}$ is the barycentre of one cell c_i whereas $\boldsymbol{x}_{c_j}$ is the one of c_j. Each face f is split into portions of faces associated to a vertex. The portion of face f associated to a vertex v is drawn in solid orange, and is defined by $\boldsymbol{x}_f$, the barycentre of f, the middle of the two edges of f passing through vertex v and $\boldsymbol{x}_v$ itself. The dual barycentric cell around a vertex v is defined by connecting surrounding cell centres, face centres and edge centres.

We denote the exponent $(.)^n$ as the state at time-step t^n, and similarly $(.)^{n+1}$ as the state at time t^{n+1}, which is separated from t^n by a time interval Δt. Time-integrals over Δt are denoted by $(.)|_n^{n+1} := \dfrac{1}{\Delta t}\displaystyle\int_{t^n}^{t^{n+1}} (.)\mathrm{d}t$.

Integrating the mass equation of Equations (4.1) over cell c and time interval Δt and using Gauss' theorem gives:

$$M_c^{n+1} - M_c^n + \Delta t \sum_{f \in \mathcal{F}_c} (\rho\, \boldsymbol{u})_f \big|_n^{n+1} \cdot \boldsymbol{S}_{c>f} = 0, \tag{4.7}$$

where $\mathcal{F}_c$ is the ensemble of all the planar polygonal faces of cell c and $\boldsymbol{S}_{c>f}$ is the *outward* surface vector. Note that the quantity $(.)_f$ coming from the integration of the divergence is defined as the average (in space) of $(.)$ over the face: $(.)_f := \dfrac{1}{S_f}\displaystyle\int_f (.)\mathrm{d}S$, and its expression with respect to the cell values will be given by the space discretisation.

Performing the same computation for the momentum equation of Equations (4.1) gives:

$$\boldsymbol{Q}_c^{n+1} - \boldsymbol{Q}_c^n + \Delta t \sum_{f \in \mathcal{F}_c} (\boldsymbol{u} \otimes \rho\, \boldsymbol{u} - \boldsymbol{\sigma})_f \big|_n^{n+1} \cdot \boldsymbol{S}_{c>f} = \boldsymbol{0}. \tag{4.8}$$

There are many schemes for solving the Navier–Stokes equations with the finite volume method (Versteeg and Malalasekera, 2007). Here, the Navier–Stokes equations are solved in two steps: a prediction step for the velocities from the explicit pressure field, and a correction step, where the continuity equation is used to calculate a change of pressure. The correction step adjusts the mass flux to ensure mass conservation and then update the velocity field. Sub-iterations for these two steps is also possible to perform a Pressure-Implicit with Splitting of Operators (PISO)-like algorithm Issa et al. (1986). These iterations will be denoted below by the superscript k, starting at 1.

For the prediction step, by default, an implicit Euler first-order scheme is used, where the viscous shear stress is time-implicit, the pressure is time-explicit, and the convective term is linearised and semi-implicit as follows:

$$
\begin{aligned}
V_c \left(\rho_c^{n+1,k-1}\widetilde{\boldsymbol{u}}_c^k - \rho_c^n \boldsymbol{u}_c^n\right) & \\
+\Delta t \sum_{f\in\mathcal{F}_c} \widetilde{\boldsymbol{u}}_f^k \left((\rho\,\boldsymbol{u})_f\big|_n^{n+1,k-1} \cdot \boldsymbol{S}_{c>f}\right) & \\
-\Delta t \sum_{f\in\mathcal{F}_c} \left(\boldsymbol{\tau}\left(\widetilde{\boldsymbol{u}}^k\right)\right)_f \cdot \boldsymbol{S}_{c>f} & \\
= -\Delta t \sum_{f\in\mathcal{F}_c} \left(p_f\big|_n^{n+1,k-1}\right) \boldsymbol{S}_{c>f}, & \qquad (4.9)
\end{aligned}
$$

where $\widetilde{\boldsymbol{u}}^k$ is the predicted velocity field to be solved, and the mass flux $(\rho\,\boldsymbol{u})_f\big|_n^{n+1,k-1} \cdot \boldsymbol{S}_{c>f}$ is initialised by $(\rho\,\boldsymbol{u})_f\big|_n^{n+1,0} \cdot \boldsymbol{S}_{c>f} = (\rho\,\boldsymbol{u})_f\big|_{n-1}^{n} \cdot \boldsymbol{S}_{c>f}$, $p|_n^{n+1,0} = p|_{n-1}^n$, $\rho^{n+1,0} = \rho^n$. Different convective schemes for $\widetilde{\boldsymbol{u}}_f^k$ have been implemented, including a pure upwind scheme, a Second-Order Linear Upwind (SOLU), or a centred scheme with an optional slope test limiter. This last approach is the one used for all test cases presented in this paper. The transpose of the velocity gradient and the isotropic term are in fact implicit in time, with an iterative process used to reconstruct terms due to the non-orthogonalities of the mesh if present Archambeau et al. (2004).

The unsteady term of Equation (4.9) is rewritten using the conservation of mass:

$$
\begin{aligned}
V_c\left(\rho_c^{n+1,k-1}\widetilde{\boldsymbol{u}}_c^k - \rho_c^n \boldsymbol{u}_c^n\right) &= V_c \rho_c^n \left(\widetilde{\boldsymbol{u}}_c^k - \boldsymbol{u}_c^n\right) \\
&\quad - \Delta t \sum_{f\in\mathcal{F}_c} \widetilde{\boldsymbol{u}}_c^k \left((\rho\,\boldsymbol{u})_f\big|_n^{n+1,k-1} \cdot \boldsymbol{S}_{c>f}\right). \qquad (4.10)
\end{aligned}
$$

(As mentioned above, although the density is taken to be constant in this paper, the density could be updated through an equation of state, depending on scalars such as temperature or salinity; if so, this modification would be applied here, before the correction step.)

During the correction step, a simplified momentum equation is solved under the mass balance constraint:

$$
\begin{aligned}
\rho^{n+1,k-1}\frac{\boldsymbol{u}^{n+1,k} - \widetilde{\boldsymbol{u}}^k}{\Delta t} + \boldsymbol{\nabla}\phi^k &= \boldsymbol{0}, \\
(\rho\boldsymbol{u})_f\big|_n^{n+1,k} \cdot \boldsymbol{S}_{c>f} &= \left(\rho^{n+1,k-1}\boldsymbol{u}^{n+1,k}\right)_f \cdot \boldsymbol{S}_{c>f}, \qquad (4.11) \\
V_c\left(\rho_c^{n+1,k} - \rho_c^n\right) + \Delta t \sum_{f\in\mathcal{F}_c} \left((\rho\boldsymbol{u})_f\big|_n^{n+1,k}\right) \cdot \boldsymbol{S}_{c>f} &= 0,
\end{aligned}
$$

where $\phi^k = p|_n^{n+1,k} - p|_n^{n+1,k-1}$ is the pressure increment. From Equation (4.11), a Poisson equation is written for ϕ^k:

$$
-\sum_{f\in\mathcal{F}_c}\left(\Delta t\boldsymbol{\nabla}\phi^k\right)_f \cdot \boldsymbol{S}_{c>f} = -\sum_{f\in\mathcal{F}_c}\left(\rho^{n+1,k-1}\widetilde{\boldsymbol{u}}^k\right)_f \cdot \boldsymbol{S}_{c>f} - V_c\frac{\rho_c^{n+1,k} - \rho_c^n}{\Delta t}. \qquad (4.12)
$$

A Two-Point Flux Approximation (TPFA) with non-orthogonal reconstruction is used to discretise the Laplace operator on ϕ^k Eymard et al. (2014), and the Rhie and Chow filter can be added to the right-hand side of Equation (4.12) (see Archambeau et al. (2004) or Chapter 1, Section 1.2.1). See Ferrand et al. (2014) for more details about the reconstruction process. Note that the first line of Equation (4.11) is then used to update the cell velocity field $\boldsymbol{u}_c^{n+1,k}$.

4.3 ALE framework and free-surface condition

Next we consider the ALE formalism, and then present the time-stepping scheme used, showing that volume is conserved when considering mesh displacement in a single dimension.

4.3.1 Governing equations

First it is necessary to formulate the equations governing the conservation of momentum in the domain, $\Omega(t)$, which moves in time. We denote the velocity of mesh displacement field $\boldsymbol{v}(\boldsymbol{x}, t)$, for points $\boldsymbol{x}(\boldsymbol{x}_0, t)$ originally located at $\boldsymbol{x}_0$ and moving in the domain. The mesh velocity of $\Omega(t)$ is thus written:

$$\boldsymbol{v}(\boldsymbol{x}, t) = \frac{\partial \boldsymbol{x}(\boldsymbol{x}_0, t)}{\partial t}. \tag{4.13}$$

We now consider a sub-domain $V_c(t)$ of $\Omega(t)$. This domain corresponds to a cell of the mesh previously described. Using the Leibniz integration rule (or Reynolds transport theorem), we can evaluate the variation of an arbitrary tensor field $\boldsymbol{\Psi}$ in $V_c(t)$ over time (here noted as a tensor of order one, i.e., a vector, but the Leibniz integration rule is true for any order). This is written as:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V_c(t)} \boldsymbol{\Psi}\mathrm{d}V_c = \int_{V_c(t)} \frac{\partial \boldsymbol{\Psi}}{\partial t}\mathrm{d}V_c + \int_{\partial V_c(t)} \boldsymbol{\Psi} \otimes \boldsymbol{v} \cdot \mathbf{d}\boldsymbol{S}. \tag{4.14}$$

It is important to note that $\frac{\mathrm{d}}{\mathrm{d}t}$ represents the total derivative of the integral quantity, that follows the movement of $V_c(t)$.

Applying the Leibniz integral rule, Equation (4.14), to the unity scalar field gives the way cell volumes evolve:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V_c(t)} \mathrm{d}\Omega = \int_{\partial V_c(t)} \boldsymbol{v} \cdot \mathbf{d}\boldsymbol{S}. \tag{4.15}$$

This last equation (in a form integrated in time) is often called the geometrical conservation law (GCL) and will be described in more detail in §4.3.2.

Performing the same calculation for the variation of mass $\frac{\mathrm{d}}{\mathrm{d}t}\int_{V_c(t)} \rho \mathrm{d}\Omega$ in the cell c and using the first line of Equation (4.1) we obtain:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V_c(t)} \rho \mathrm{d}\Omega + \int_{\partial V_c(t)} \rho(\boldsymbol{u} - \boldsymbol{v}) \cdot \mathbf{d}\boldsymbol{S} = 0. \tag{4.16}$$

Applying again the Leibniz integral rule (4.14) to establish the conservation of momentum in the domain $V_c(t)$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V_c(t)} \rho\boldsymbol{u}\mathrm{d}V_c = \int_{V_c(t)} \frac{\partial \rho\boldsymbol{u}}{\partial t}\mathrm{d}V_c + \int_{\partial V_c(t)} \rho\boldsymbol{u} \otimes \boldsymbol{v} \cdot \mathbf{d}\boldsymbol{S}. \tag{4.17}$$

We apply the Cauchy momentum equation to replace $\frac{\partial \rho\boldsymbol{u}}{\partial t}$ of Equation (4.17), which gives us:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V_c(t)} \rho\boldsymbol{u}\mathrm{d}\Omega + \int_{\partial V_c(t)} \boldsymbol{u} \otimes \rho(\boldsymbol{u} - \boldsymbol{v}) \cdot \mathbf{d}\boldsymbol{S} = \int_{\partial V_c(t)} \boldsymbol{\sigma} \cdot \mathbf{d}\boldsymbol{S}. \tag{4.18}$$

4.3.2 Space and time discretisation

Let us start by integrating the volume conservation Equation (4.15) over a time step Δt:

$$V_c^{n+1} - V_c^n = \int_{t^n}^{t^{n+1}} \left(\sum_{f \in \mathcal{F}_c} \int_f \boldsymbol{v} \cdot \mathbf{d}\boldsymbol{S} \right) \mathrm{d}t. \tag{4.19}$$

For calculating the fluxes present in Equation (4.19), we must determine on which mesh we will perform the calculations, whether on the mesh Ω^n, or on Ω^{n+1}, or on several meshes at intermediate times between Ω^n and Ω^{n+1}.

If the mesh displacement is one-dimensional, as for smooth free-surface applications where the movement of the mesh can be limited to the same direction as gravity (typically vertically), without overturning, $\boldsymbol{v} = w\boldsymbol{e}_z$, the mesh velocity flux through cell faces can be discretised exactly as:

$$\begin{aligned} V_c^{n+1} - V_c^n &= \Delta t \sum_{f \in \mathcal{F}_c} \boldsymbol{v}_f \big|_n^{n+1} \cdot \boldsymbol{S}_{c>f}^n \\ &= \sum_{f \in \mathcal{F}_c} \left(\boldsymbol{x}_f^{n+1} - \boldsymbol{x}_f^n \right) \cdot \boldsymbol{S}_{c>f}^n, \end{aligned} \tag{4.20}$$

where $\boldsymbol{x}_f^n$ is the face barycentre at time t^n. This choice will be motivated and justified in Section 4.3.3.

Time integration of the conservation of mass and momentum, Equation (4.16) and (4.18), gives:

$$\begin{aligned} M_c^{n+1} - M_c^n + \sum_{f \in \mathcal{F}_c} \int_{t^n}^{t^{n+1}} (\rho (\boldsymbol{u} - \boldsymbol{v}))_f \cdot \boldsymbol{S}_{c>f} \mathrm{d}t &= 0, \\ \boldsymbol{Q}_c^{n+1} - \boldsymbol{Q}_c^n + \sum_{f \in \mathcal{F}_c} \int_{t^n}^{t^{n+1}} \boldsymbol{u}_f (\rho (\boldsymbol{u} - \boldsymbol{v}))_f \cdot \boldsymbol{S}_{c>f} \mathrm{d}t &= \sum_{f \in \mathcal{F}_c} \int_{t^n}^{t^{n+1}} \boldsymbol{\sigma}_f \cdot \boldsymbol{S}_{c>f} \mathrm{d}t. \end{aligned} \tag{4.21}$$

It is important to remark here that ρ_c^{n+1} and $\boldsymbol{u}_c^{n+1}$ correspond to the mean density and the mean velocity of cell c at its location at time t^{n+1}, and no reconstruction is needed.

At this point one must choose a discretisation in time to resolve the velocity field. As for the discrete volume conservation, Equation (4.20), fluxes are written on the faces at time t^n.

The prediction step for the velocity, which was written for fixed meshes in Eq. (4.9), can now be written on moving meshes as:

$$M_c^n \frac{\widetilde{\boldsymbol{u}}_c^k - \boldsymbol{u}_c^n}{\Delta t} - \sum_{f \in \mathcal{F}_c} \widetilde{\boldsymbol{\tau}}_f^k \cdot \boldsymbol{S}_{c>f}^n + \sum_{f \in \mathcal{F}_c} \left(\widetilde{\boldsymbol{u}}_f^k - \widetilde{\boldsymbol{u}}_c^k \right) (\rho (\boldsymbol{u} - \boldsymbol{v}))_f \big|_n^{n+1,k-1} \cdot \boldsymbol{S}_{c>f}^n = - \sum_{f \in \mathcal{F}_c} \left(p|_n^{n+1,k-1} \right)_f \boldsymbol{S}_{c>f}^n. \tag{4.22}$$

We see that pressure and friction forces are applied on the mesh at time n and mass flux takes face displacement into account. Again, the conservation of mass was used to transform the unsteady term and rearranged with the convective terms.

The correction step of the velocity, Equation (4.11), remains unchanged on mesh at time t^n. At the end of this process, we obtain $\boldsymbol{u}_c^{n+1,k}$, $p_c|_n^{n+1,k}$, and the fluid mass flux $(\rho \boldsymbol{u})_f \big|_n^{n+1,k} \cdot \boldsymbol{S}_{c>f}^n$. Then,

the mesh velocity is computed and the mass flux due to mesh displacement, $(\rho \boldsymbol{v})_f|_n^{n+1,k} \cdot \boldsymbol{S}_{c>f}^n$, is subtracted from it. The total mass flux is used in any transport equation afterwards. We see here that the mesh velocity has to be defined at the nodes of the mesh to move them, and it is also used when calculating the flux $(\rho \boldsymbol{v})_f|_n^{n+1,k-1} \cdot \boldsymbol{S}_{c>f}^n$ at cell faces in Equation (4.22). Face mesh-velocity is deduced from mesh-velocities at the vertices by:

$$\boldsymbol{v}_f|_n^{n+1} = \sum_{v \in \mathcal{V}_f} \frac{\boldsymbol{v}_v|_n^{n+1} \cdot \boldsymbol{S}_{f,v}}{S_f^2} \boldsymbol{S}_{c>f}, \tag{4.23}$$

where $\boldsymbol{S}_{c>f,v}$ is the portion of face f *outward* vector associated to vertex v displayed in Fig. 4.1.

4.3.3 Verification of the discretised geometric conservation law

The geometric conservation law (GCL) states that a variation of volume V_c of a cell c between time-step t^n and t^{n+1} should be equal to the integral over this time interval of the mesh velocity flux through faces of cell c:

$$V_c^{n+1} - V_c^n = \int_{t^n}^{t^{n+1}} \int_{\partial V_c} \boldsymbol{v} \cdot \mathrm{d}\boldsymbol{S} \, \mathrm{d}t. \tag{4.24}$$

The time integration scheme used for the mesh velocity allows one to rewrite the time integral as:

$$\int_{t^n}^{t^{n+1}} \int_{\partial V_c} \boldsymbol{v} \cdot \mathrm{d}\boldsymbol{S} \, \mathrm{d}t = \Delta t \int_{\partial V_c^n} \boldsymbol{v}|_n^{n+1} \cdot \mathrm{d}\boldsymbol{S}^n. \tag{4.25}$$

It is clear that the condition in Equation (4.24) does not hold true if the mesh velocity $\boldsymbol{v}$ contains more than one direction of variation. The integration scheme must be modified if we want to satisfy the GCL for mesh velocities which vary in two or three directions (see Farhat et al., 2001, for a presentation of these types of schemes and their properties).

For one-dimensional mesh movement in z, one sees that $\left(\boldsymbol{x}_f^{n+1} - \boldsymbol{x}_f^n\right) \cdot \boldsymbol{S}_{c>f}^n = \left(z_f^{n+1} - z_f^n\right) S_{c>f,z}^n$, where $S_{c>f,z}$ is the z component of $\boldsymbol{S}$. Hence, volume of cell c can be computed as:

$$V_c = \int_{V_c(t)} \mathrm{div}\ (0,\ 0,\ z)\, \mathrm{d}\Omega \qquad = \sum_{f \in \mathcal{F}_c} z_f S_{c>f,z}, \tag{4.26}$$

and one can prove that $S_{c>f,z}$ is constant over time. Indeed, we can first remark that $\boldsymbol{S}_{c>f}$ can be defined using only surrounding vertices of f, even if the face is warped. Formulae proposed by Carré et al. (2009) for a hexahedron can be extended for any polyhedron. We can remark that $S_{c>f,x}$ can be computed as $S_{c>f,x} = \int_f \mathbf{curl}\, \boldsymbol{r}_x \cdot \mathbf{d}\boldsymbol{S} = \oint_{\partial f} \boldsymbol{r}_x \cdot \mathbf{d}\boldsymbol{l}$ with $\boldsymbol{r}_x = \frac{1}{2}(0,\ -z,\ y)$ such that $\mathbf{curl}\, \boldsymbol{r}_x = (1,\ 0,\ 0)$. Similarly, $S_{c>f,y} = \int_f \mathbf{curl}\, \boldsymbol{r}_y \cdot \mathbf{d}\boldsymbol{S} = \oint_{\partial f} \boldsymbol{r}_y \cdot \mathbf{d}\boldsymbol{l}$ and $S_{c>f,z} = \int_f \mathbf{curl}\, \boldsymbol{r}_z \cdot \mathbf{d}\boldsymbol{S} = \oint_{\partial f} \boldsymbol{r}_z \cdot \mathbf{d}\boldsymbol{l}$ with $\boldsymbol{r}_y = \frac{1}{2}(z,\ 0,\ -x)$ and $\boldsymbol{r}_z = \frac{1}{2}(-y,\ x,\ 0)$ such that $\mathbf{curl}\, \boldsymbol{r}_y = (0,\ 1,\ 0)$ and $\mathbf{curl}\, \boldsymbol{r}_z = (0,\ 0,\ 1)$. So $S_{c>f,z}$ depends only on the x and y components of the vertices of the surrounding edges of face f. Consequently, Equation (4.20) is exactly fulfilled if faces remain planar. If the mesh displacement is 2-D or 3-D, the mesh velocity fluxes must be split into two or three steps to exactly conserve the volume Farhat et al. (2001). Note also that the calculation of $\boldsymbol{S}_{c>f}$ as a curl ensures that for any cell c, $\sum_{f \in \mathcal{F}_c} \boldsymbol{S}_{c>f} = \boldsymbol{0}$ even for warped faces.

Conservation of a uniform velocity field

The time integration scheme chosen has to have good properties to maintain a uniform velocity field. This point is key if we would like to have a stable scheme and verify the maximum principle (see Farhat et al., 2001).

Let us consider the numerical scheme of the prediction step, Equation (4.22), rewritten in velocity increment $\delta \boldsymbol{u} = \widetilde{\boldsymbol{u}}^k - \boldsymbol{u}^n$ for a constant pressure field:

$$\begin{aligned}\frac{M_c^n}{\Delta t}\delta \boldsymbol{u}_c + \sum_{f\in\mathcal{F}_c} \left(\delta \boldsymbol{u}_f - \delta \boldsymbol{u}_c\right) \left(\rho\left(\boldsymbol{u}-\boldsymbol{v}\right)\right)_f\Big|_n^{n+1,k-1} \cdot \boldsymbol{S}_{c>f}^n \\ = -\sum_{f\in\mathcal{F}_c} \left(\boldsymbol{u}_f^n - \boldsymbol{u}_c^n\right) \left(\rho\left(\boldsymbol{u}-\boldsymbol{v}\right)\right)_f\Big|_n^{n+1,k-1} \cdot \boldsymbol{S}_{c>f}^n. \end{aligned} \tag{4.27}$$

We have not considered diffusive terms, but this does not change the conclusions here. From this, it is easy to verify that for a uniform field $\boldsymbol{u}^n$, the right-hand-side of Equation (4.27) is zero. For an upwind convective scheme on the velocity increment, the system to solve is written as:

$$\boldsymbol{EM}\delta \boldsymbol{u}^{n+1} = \boldsymbol{0}, \tag{4.28}$$

with $EM_{cc} = \dfrac{M_c^n}{\Delta t} + \displaystyle\sum_{f\in\mathcal{F}_c} \max\left(-\left(\rho\left(\boldsymbol{u}-\boldsymbol{v}\right)\right)_f\Big|_n^{n+1,k-1} \cdot \boldsymbol{S}_{c>f}^n,\, 0\right)$, and for $c \neq \overline{c}$ and f the face between c and $\overline{c}$,
$EM_{c\overline{c}} = -\max\left(\left(\rho\left(\boldsymbol{u}-\boldsymbol{v}\right)\right)_f\Big|_n^{n+1,k-1} \cdot \boldsymbol{S}_{c>f}^n,\, 0\right)$. One can note that $\boldsymbol{EM}$ is a M-matrix (for all $c \neq \overline{c}$, $EM_{c\overline{c}} \leq 0$ and for all c, $EM_{cc} > \sum_{\overline{c}} |EM_{c\overline{c}}|$) and is therefore invertible.

Just like with a uniform zero velocity, the numerical verification of this assertion is presented in Fig. 4.2, where an initial field $\boldsymbol{u} = \boldsymbol{e}_z 1.0\,\mathrm{m/s}$ is not modified by the mesh displacement imposed for all nodes by:

$$\frac{\boldsymbol{x}-\boldsymbol{x}_0}{L} = 4\sin\left(\frac{2\pi t}{T}\right)\frac{x_0}{L}\left(1-\frac{x_0}{L}\right)\frac{z_0}{L}\left(1-\frac{z_0}{L}\right)\begin{pmatrix}1\\0\\1\end{pmatrix}, \tag{4.29}$$

where $L = 1\,\mathrm{m}$ is also the size of the square computation domain, and $T = 4\,\mathrm{s}$ is the period of the mesh velocity variations.

4.3.4 Vector solving of the mesh velocity

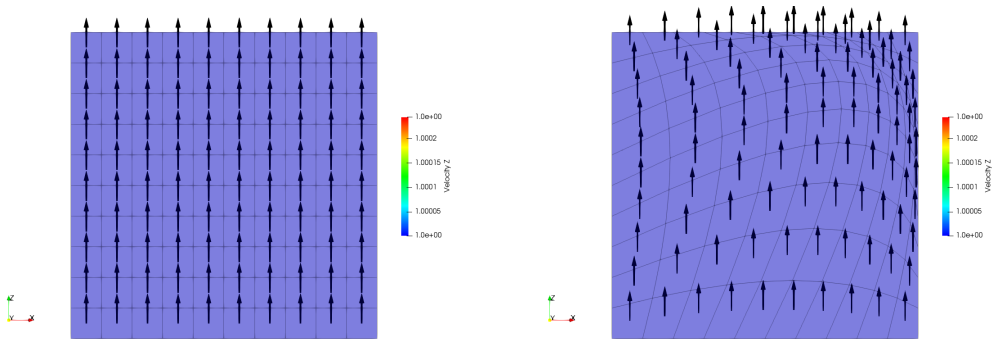
Up to this point, we have considered a mesh velocity only to verify the boundary conditions of the moving mesh, but any arbitrary velocity that satisfies these conditions is acceptable. Here, the mesh velocity is given by solving a Poisson equation:

$$\mathbf{div}\left(\lambda\, \boldsymbol{\nabla}\, \boldsymbol{v}\right) = \boldsymbol{0}, \tag{4.30}$$

$$\boldsymbol{v}|_{\partial\Omega^{imp}(t)} = \text{Imposed velocity}, \tag{4.31}$$

$$\left.\frac{\partial\left(\boldsymbol{v}-\boldsymbol{v}\cdot\boldsymbol{n}\,\boldsymbol{n}\right)}{\partial n}\right|_{\partial\Omega\setminus\partial\Omega^{imp}(t)} = \boldsymbol{0}, \tag{4.32}$$

$$\boldsymbol{v}\cdot\boldsymbol{n}|_{\partial\Omega\setminus\partial\Omega^{imp}(t)} = 0, \tag{4.33}$$



(a) Mesh and velocity field at initial time.

(b) Mesh and velocity field after 1 s.

Figure 4.2: Conservation of a uniform velocity field despite a two-dimensional deformation of the mesh given by Equation (4.29).

where $\partial\Omega^{imp}(t)$ represents the boundary of the domain with either a free-surface or imposed velocity, and where $\partial\Omega \setminus \partial\Omega^{imp}(t)$ is the fixed boundary where we allow the mesh to slide along this boundary. Of course one could also impose the mesh displacement within the domain, or even zero (a fixed mesh), or impose a sliding condition on a free-surface (i.e., to fix a non-zero value of $\boldsymbol{v} \cdot \boldsymbol{n}$ and fix a Neumann condition for the tangential part of the mesh velocity on the boundary).

The tensor $\boldsymbol{\lambda}$ is normally taken to be the identity matrix, but an anisotropic heterogeneous tensor can be chosen to make the mesh more rigid in a particular direction and in a given region. Note that having an anisotropic $\boldsymbol{\lambda}$ induces a strong coupling between the components of the mesh velocity $\boldsymbol{v}$ and therefore requires a coupled solver.

At this stage, it is important to note that it is more natural to solve Equation (4.30) with each Degree of Freedom (DoF) stored at vertices of the mesh. code_saturne legacy schemes are cell-based with the mesh velocity $\boldsymbol{v}_c$ stored at the cell centres. Therefore, once $\boldsymbol{v}_c$ is calculated, nodes displacement must be deduced by interpolation.

code_saturne can make use of algorithms based on the compatible discrete operators (CDO) framework Bonelle and Ern (2014), and DoFs can be either at cell centres (cell-based schemes) or at vertices (vertex-based schemes). A CDO vector vertex-based solver on the dual barycentric mesh is thus used to solve Equation (4.30) and is compared to the legacy approach detailed hereafter.

Free-surface boundary condition

The next step is to make use of the ALE module for modelling a free-surface. This work is similar to what is performed in other models like Telemac3D (Hervouet, 2007), in the sense that the computational domain evolves with the free-surface, but contrary to Telemac3D, fluid velocity and pressure are solved with a cell-based finite volume scheme and not with a vertex based finite element scheme, and domain is meshed using cells of any star-shaped polyhedral type.

The aim of the proposed algorithm is to solve mesh velocity after solving fluid quantities to impose correctly the kinematic condition at the free-surface without any sub-iterations. In addition, the mesh-velocity solver is component-coupled so that sliding conditions are taken into account in an exact manner.

To use the ALE module to simulate a free-surface, we first must specify the free-surface boundary conditions for both the fluid and mesh velocities. For the first one, in the absence of wind, a homogeneous Neumann condition is applied on the fluid as well as a constant pressure

p_0:

$$\left.\frac{\partial \boldsymbol{u}}{\partial n}\right|_{\text{surface}} = \mathbf{0},\ p = p_0. \tag{4.34}$$

It would be possible to take into account wind by providing a shear stress between the air and water (see Hervouet (2007) for an example of this approach). For simulations with turbulence, it is also necessary to impose conditions for the turbulent quantities. Note that for a $k-\epsilon$ model, it is preferable not to impose a homogeneous Neumann condition for these two quantities, as the scale of the vortices (and thus turbulent viscosity) is reduced by the effect of the free-surface (see Celik and Rodi, 1984, for more details).

The aim is to impose a kinematic condition for the mesh velocity:

$$\boldsymbol{v}\cdot\boldsymbol{n}|_{\text{surface}} = \boldsymbol{u}\cdot\boldsymbol{n}|_{\text{surface}}\,. \tag{4.35}$$

For now we assume that the mesh velocity is unidirectional, for example $\boldsymbol{v} = w\boldsymbol{e}_z$. This assumption is done following the remarks in 4.3.3 concerning the GCL.

Since mass flux is known explicitly when solving the mesh velocity, the flux $\boldsymbol{u}\cdot\boldsymbol{n}$ is used as a boundary condition for the mesh velocity, as summarised below in Section 4.3.5.

Legacy approach: node displacements from the mesh velocity

In this section, cell-based mesh velocity $\boldsymbol{v}_c$ is obtained with the vector coupled-solver and will be denoted by the "legacy" version.

We differentiate first the nodes on the external boundaries (called the exterior nodes) and others (called interior nodes). We specify here that the nodes which are on a sliding surface are considered internal, with a specific treatment detailed hereafter. Thus, all nodes belonging to $\partial\Omega^{imp}(t)$ are considered as external, all the others as internal.

Treatment of internal nodes

For an interior node v, we note C_v as the list of cells c which include node v. If $\boldsymbol{x}_c$ is designated the centre of a cell $c \in C_v$, the velocity $\boldsymbol{v}_v$ of v is defined by:

$$\boldsymbol{v}_v = \frac{\displaystyle\sum_{c\in C_v}\frac{1}{V_c}\left(\boldsymbol{v}_c + \boldsymbol{\nabla}\boldsymbol{v}_c\cdot(\boldsymbol{x}_v-\boldsymbol{x}_c)\right)}{\displaystyle\sum_{c\in C_v}\frac{1}{V_c}}. \tag{4.36}$$

Treatment of exterior nodes

For an exterior node v, we set $\mathcal{F}_v$ as the list of external faces where v appears. If $\boldsymbol{x}_f$ is designated the centre of a face $f \in \mathcal{F}_v$ and $\boldsymbol{x}_c$ is the centre of the cell where f appears, we then have a velocity $\boldsymbol{v}_v$ of v defined by:

$$\boldsymbol{v}_v = \frac{\displaystyle\sum_{f\in\mathcal{F}_v}\frac{1}{|S_f|}\left(\boldsymbol{v}_f + \boldsymbol{\nabla}\boldsymbol{v}_{\boldsymbol{x}_c}\cdot\left(\boldsymbol{x}_v-\boldsymbol{x}_f\right)\right)}{\displaystyle\sum_{f\in\mathcal{F}_v}\frac{1}{|S_f|}}. \tag{4.37}$$

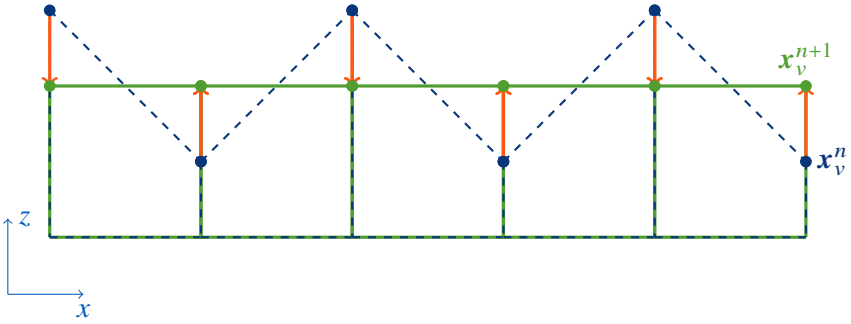


Figure 4.3: Illustration of the effect of the filtering (4.42) on the standing wave test case described in Section 4.4.2 with an initial wavelength of $2\Delta x$ planned in one time step. Without filtering, no fluid nor mesh motion is obtained.

Treatment of internal nodes at a sliding boundary

For an internal node v already treated by Equation (4.36) which also appears on a sliding face f and with an exterior normal vector $\boldsymbol{n}_{c>f}$, we project $\boldsymbol{v}_v$ on the face f as follows:

$$\boldsymbol{v}_v \leftarrow (\mathbf{1} - \boldsymbol{n} \otimes \boldsymbol{n})\, \boldsymbol{v}_v = \boldsymbol{v}_v - (\boldsymbol{v}_v \cdot \boldsymbol{n})\, \boldsymbol{n}. \tag{4.38}$$

CDO approach: node displacements from the mesh velocity

In this section, mesh DoFs of the mesh velocity are directly associated to the nodes and the main advantage is that no interpolation is needed to displace nodes:

$$\boldsymbol{x}_v^{n+1} = \boldsymbol{x}_v^n + \Delta t\ \boldsymbol{v}_v|_n^{n+1}. \tag{4.39}$$

At the same time, boundary conditions on primal boundary faces have to be translated in terms of node mesh-velocity $\boldsymbol{v}_v$ on the dual mesh for the free-surface condition. For a node v at the free-surface, we set $\mathcal{F}_v$ as the list of external free-surface faces where v appears, and for each face f of $\mathcal{F}_v$, c_f is the boundary cell to which f belongs to. The node mesh-velocity is therefore defined as:

$$\boldsymbol{v}_v|_n^{n+1} = \frac{\displaystyle\sum_{f\in\mathcal{F}_v} \frac{\rho \boldsymbol{u}_f|_n^{n+1} \cdot \boldsymbol{n}_{c>f}}{\rho_{c_f}} \boldsymbol{S}_{c>f,v} \cdot \boldsymbol{e}_z}{\displaystyle\sum_{f\in\mathcal{F}_v} \boldsymbol{S}_{c>f,v} \cdot \boldsymbol{e}_z} \boldsymbol{e}_z. \tag{4.40}$$

Equation (4.40) does not ensure local conservation, because the face centre mesh velocity $\boldsymbol{v}_f|_n^{n+1}$, deduced from the nodes mesh velocity by (4.23) will not give back $\frac{\rho \boldsymbol{u}_f|_n^{n+1} \cdot \boldsymbol{n}_{c>f}}{\rho_{c_f}}$. However, global conservation is ensured:

$$\sum_{f\in\text{free-surface}} \rho_{c_f}\ \boldsymbol{v}_f|_n^{n+1} \cdot \boldsymbol{S}_{c>f} = \sum_{f\in\text{free-surface}} \rho \boldsymbol{u}_f|_n^{n+1} \cdot \boldsymbol{S}_{c>f}. \tag{4.41}$$

Repartition of mass fluxes from faces to nodes written in Equation (4.40) can be modified without breaking the global conservation (4.41). Indeed, if we add any affine function of zero mean over

face f, (4.41) still holds. This property is convenient to filter modes such as waves of length $\lambda = 2\Delta x$ where Δx is the horizontal space grid, or to smooth too steep waves before they break (from numerical, rather than physical, reasons). We modify Equation (4.40) by:

$$\boldsymbol{v}_v|_n^{n+1} = \frac{\displaystyle\sum_{f\in\mathcal{F}_v}\left(\frac{\rho \boldsymbol{u}_f|_n^{n+1}\cdot\boldsymbol{n}_{c>f}}{\rho_{c_f}} - \alpha_f\frac{z_v - z_f}{\Delta t}\right)\boldsymbol{S}_{c>f,v}\cdot\boldsymbol{e}_z}{\displaystyle\sum_{f\in\mathcal{F}_v}\boldsymbol{S}_{c>f,v}\cdot\boldsymbol{e}_z}\boldsymbol{e}_z, \tag{4.42}$$

where α_f is a factor equal to 0 or 1 to activate the filter only when necessary. α_f is set to 1 for face f when there is at least one of its nodes which is a local maximum (higher than all the neighbouring face centres) and one of its nodes which is a local minimum (this criterion corresponds to the characterisation of a wavelength $2\Delta x$). α_f is also set to 1 for face f which is too steep (if one edge v_0v_1 is such that Δx is smaller than Δz, where Δx and Δz are defined by:

$$\begin{aligned}\Delta x &= \left\|(\mathbf{1}-\boldsymbol{e}_z\otimes\boldsymbol{e}_z)\cdot\left(\tfrac{\boldsymbol{x}_{v_0}+\boldsymbol{x}_{v_1}}{2}-\boldsymbol{x}_f\right)\right\|,\\ \Delta z &= \left|\left(\tfrac{\boldsymbol{x}_{v_0}+\boldsymbol{x}_{v_1}}{2}-\boldsymbol{x}_f\right)\cdot\boldsymbol{e}_z\right|.\end{aligned} \tag{4.43}$$

Term $\alpha_f\frac{z_v-z_f}{\Delta t}$ acts as a volume-conservative smoother for nodes that have an elevation different from the weighted mean elevation of the surrounding faces. The behavior of the filtering given by Equation (4.42) is illustrated in Fig. 4.3 where the free-surface is initialised with a sawtooth elevation of $2\Delta x$ wavelength, but quiescent domain (with zero fluid velocity). In this case, water is supposed to be at rest and elevation modes cannot be suppressed by the legacy scheme. The filtering is removing it in one step, without changing the potential energy based on the elevation of free-surface faces. Similar types of instabilities have been known since Longuet-Higgins and Cokelet (1976). More sophisticated approaches to removing this issue have been illustrated by Mola et al. (2013).

4.3.5 Summary of the ALE algorithm

The algorithm described above to solve the Navier–Stokes equations on a moving mesh reads:

for time interval n to $n+1$ **do**
  for sub-iteration k **do**
    Solve the predicted velocity $\widetilde{\boldsymbol{u}}_c^{n+1,k}$ on Ω^n;
    Calculate corrected velocity $\boldsymbol{u}_c^{n+1,k}$ on Ω^n and the flux $\boldsymbol{u}_f|_n^{n+1,k}\cdot\boldsymbol{S}_{c>f}^n$;
    if using legacy solver for $\boldsymbol{v}$ **then**
      Solve $\boldsymbol{v}_c|_n^{n+1,k}$ on the primal mesh Ω^n with boundary conditions (4.35);
      Reconstruct velocity from cell centred values at nodes $\boldsymbol{v}_v|_n^{n+1,k}$;
    else if using CDO solver for $\boldsymbol{v}$ **then**
      Calculate $\boldsymbol{v}$ boundary conditions at nodes;
      Solve $\boldsymbol{v}_v|_n^{n+1,k}$ on the original dual mesh Ω^0 with boundary conditions (4.35);
    end if
    Update mesh from Ω^n to $\Omega^{n+1,k}$ using $\boldsymbol{v}_v|_n^{n+1,k}$;
    Calculate total mass flux $\rho(\boldsymbol{u}-\boldsymbol{v})_f|_n^{n+1,k}\cdot\boldsymbol{S}_{c>f}^n$;
  end for
end for

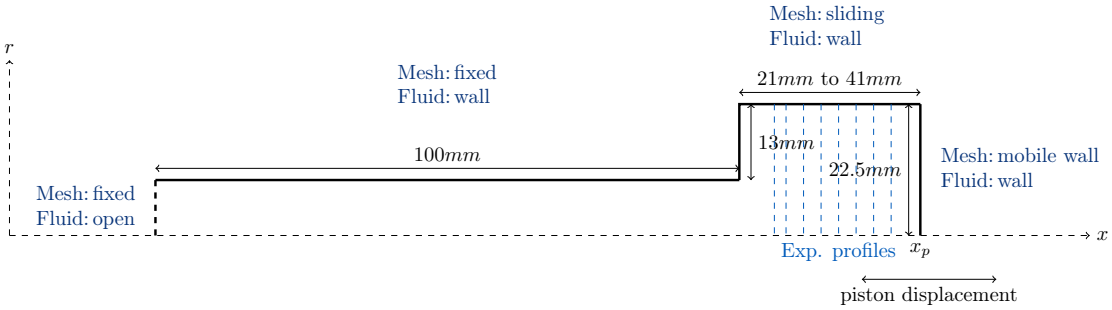


Figure 4.4: Sketch of the valve validation case.

4.4 Verification and validation test cases

In order to validate these developments for modelling wave propagation and wave-structure interaction, we consider a variety of verification and validation test cases, showing the fluid motion generated by a piston in order to validate the ALE module without free-surface condition; standing waves in a closed basin with small and high initial wave amplitudes to verify volume conservation; the viscous damping of waves, in order to validate the proper rate of decay due to viscosity; the propagation of a solitary wave, showing the exact reproduction of a large amplitude wave; the propagation of waves over a submerged bar, showing the ability to handle shoaling; the force on a vertical cylinder in regular waves, showing issues of wave-structure interaction; and finally comparisons against a novel type of wave-maker.

4.4.1 Test case of a 2-D axisymmetric piston

This case consists in studying a fluid flow inside a cylindrical valve (piston) with a sudden expansion. Numerical results are compared to experimental data from Durst et al. (1989). The simulated domain is axisymmetric. The flow is laminar ($Re \simeq 320$), with a density of $\rho = 1000\,\mathrm{kg/m^3}$ and a dynamic viscosity $\mu = 4 \times 10^{-3}\,\mathrm{kg/m/s}$.

Fig. 4.4 describes the geometry and boundary conditions for the Navier-Stokes equations and the mesh velocity. Two different kinds of "mobile wall condition" are implemented in code_saturne:

- imposed velocity of the interface at the centre of each boundary face;
- given displacement of each node of the mobile interface.

When imposing piston motion, it is decomposed into six steps in which either the piston velocity v_p or the piston displacement d_p are prescribed, as summarized in Table 4.1.

The piston extension is maximum at time T_{TDC} (Top Dead Center, $x_p = X_{max} = 0.141\,\mathrm{m}$) and minimum at time T_{BDC} (Bottom Dead Center, $x_p = X_{min} = 0.121\,\mathrm{m}$) given in Eq. (4.44). The maximum velocity during the injection phase is $V_i = 0.006\,22\,\mathrm{m/s}$, and the time during the acceleration phase in which the velocity passes from zero to V_i is given by $T_i = 0.05\,\mathrm{s}$. The minimum velocity during the compression phase is $V_c = 0.0265\,\mathrm{m/s}$, and the time during the acceleration phase in which the velocity passes from zero to $-V_c$ is given by $T_c = 0.18\,\mathrm{s}$ where $X_i = \dfrac{V_i \times T_i}{2}$, $X_c = \dfrac{V_c \times T_c}{2}$. Note that for the maximum expansion time and the complete

Time	Piston velocity	Piston displacement
$0 \leq t \leq T_i$	$v_p = \frac{V_i}{T_i} t$	$d_p = \frac{V_i}{T_i} \frac{t^2}{2}$
$T_i \leq t \leq T_{TDC} - T_i$	$v_p = V_i$	$d_p = X_i + V_i (t - T_i)$
$T_{TDC} - T_i \leq t \leq T_{TDC}$	$v_p = \frac{V_i}{T_i} (T_{TDC} - t)$	$d_p = 2X_i + V_i (T_{TDC} - 2T_i) - \frac{V_i}{T_i} \frac{(T_{TDC} - t)^2}{2}$
$T_{TDC} \leq t \leq T_{TDC} + T_c$	$v_p = -\frac{V_c}{T_c} (t - T_{TDC})$	$d_p = 2X_i + V_i (T_{TDC} - 2T_i) - \frac{V_c}{T_c} \frac{(t - T_{TDC})^2}{2}$
$T_{TDC} + T_c \leq t \leq T_{BDC} - T_c$	$v_p = -V_c$	$d_p = 2X_i + V_i (T_{TDC} - 2T_i) - x_2 - V_c (t - T_{TDC} - T_c)$
$T_{TDC} - T_c \leq t \leq T_{BDC}$	$v_p = -\frac{V_c}{T_c} (T_{BDC} - t)$	$d_p = \frac{V_c}{T_c} \frac{(t - T_{TDC})^2}{2}$

Table 4.1: Piston prescribed velocity and displacement.

compression time the following equations holds:

$$\begin{aligned} T_{TDC} &= \frac{X_{max} - X_{min} + 2X_i}{V_i} \simeq 3.265\,\text{s}, \\ T_{BDC} &= T_{TDC} + \frac{X_{max} - X_{min} + 2X_c}{V_c} \simeq 4.200\,\text{s}. \end{aligned} \tag{4.44}$$

Axial velocity profiles and radial velocity profiles are respectively displayed in Fig. 4.6 and in Fig. 4.7 for the compression step, and in Fig. 4.8 and in Fig. 4.9 for the injection step. The sub-figures correspond to the different abscissae.

First, one can notice that same results are obtained with the CDO and legacy solver for the mesh velocity, and from imposing mesh displacement or directly the mesh velocity. This is expected because the code transforms the mesh displacement boundary condition into a mesh velocity boundary condition to compute it. One can also verify that the incoming and outgoing mass flux should be equal to the imposed mass flux due to the valve displacement. As the flow is incompressible, the two fluxes displayed in Fig. 4.5 must be equal.

4.4.2 Standing wave in a closed basin

This case is initialised by a free-surface with a sinusoidal form with a wavelength λ. We consider a basin with length $L = 400\,\text{m}$, and average depth $h_0 = 10\,\text{m}$ and we fix $\lambda = \frac{L}{2}$. The domain is two-dimensional (*i.e*, one cell thick) and the mesh used is composed of 10 cells in the z direction and 400 in the x direction.

In order to fix a period of the wave close to 10 s, we take the acceleration due to gravity to be $g = 40\,\text{m/s}^2$.

(Alternatively, we could start from a nonlinear standing wave solution instead of a pure sinusoidal initial condition, but our goal here is not to look at the nonlinear wave dynamics, which we will consider in the next cases, but simply to evaluate the volume conservation and stability.)

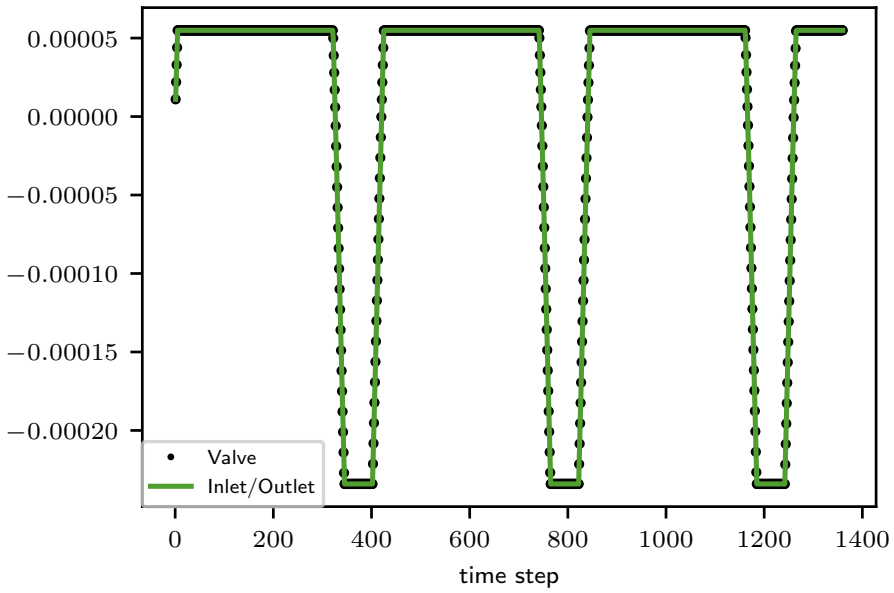


Figure 4.5: Comparison of mass flux computed at the open boundary (green) and the value imposed by the piston, represented in black dots.

The free-surface deforms over time, and with an initial amplitude A, one or several (particularly for large amplitude initial conditions) modes become visible. To evaluate the numerical algorithm used, we begin by studying a wave which is quasi-linear.

Linear standing wave

When $\frac{A}{h_0} \ll 1$, linear wave theory can be used to describe the free-surface, $z_{surf}\,(x,\,t)$, which is given by:

$$z_{surf}\,(x,\,t) = A\cos\left(\frac{2\pi x}{\lambda}\right)\cos\left(\frac{2\pi t}{T}\right), \tag{4.45}$$

where the period of the wave is given by the celerity c, as $T = \frac{\lambda}{c}$, with the dispersion relation giving:

$$c_0^2 = g\frac{\lambda}{2\pi}\tanh\left(\frac{2\pi h_0}{\lambda}\right). \tag{4.46}$$

In fact we consider a shallow water case, where $h \ll \lambda$, so we can approximate this as $c_0 \simeq \sqrt{g h_0}$. We obtain precisely:

$$c_0 = \sqrt{g\frac{\lambda}{2\pi}\tanh\left(\frac{2\pi h_0}{\lambda}\right)} \simeq 19.681\ \mathrm{m/s}, \tag{4.47}$$

giving $T_0 = \frac{\lambda}{c_0} \simeq 10.162\,\mathrm{s}$.

We choose to start with a case with wave amplitude $A = 0.025\,\mathrm{m}$. The simulation of one hundred periods is shown in Figure 4.10 where we can observe the evolution of the wave height in time at the point $x = 0$ (the centre of the domain).

The period obtained numerically is $T_{num} = 10.163 \pm 0.001\,\mathrm{s}$ (the range obtained is taken to be the time-step divided by the number of periods observed). This is therefore a good match for the theoretical T_0, which we see in Fig. 4.10, $T = T_0 \pm 0.025\,\mathrm{s}$.

We note that in Fig. 4.10, that for a relatively long duration, the wave-height remains nearly constant which indicates that the model is not losing significant amounts of volume nor energy.

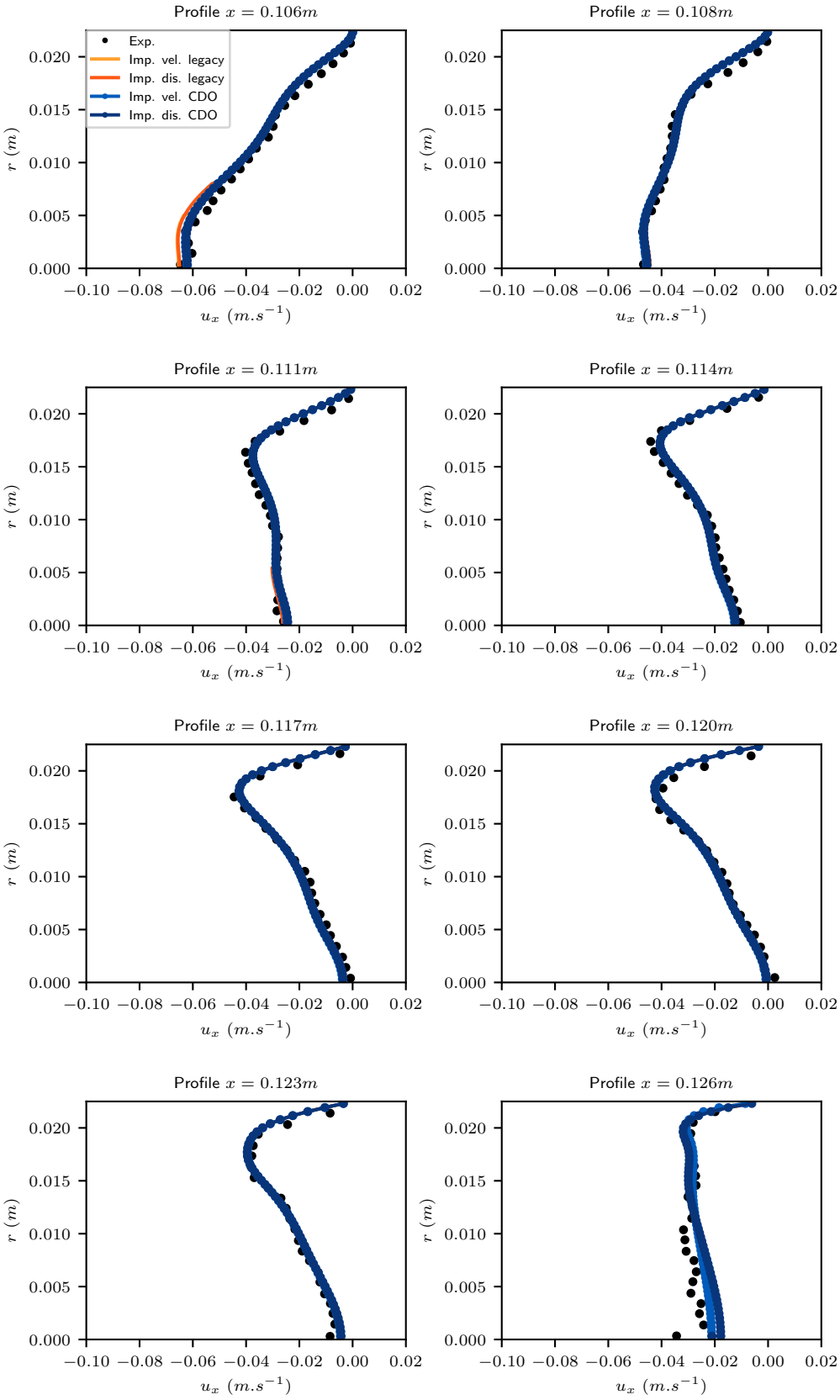


Figure 4.6: Axial velocity profiles at different positions x for the compression step ($x_p = 0.127$ m).

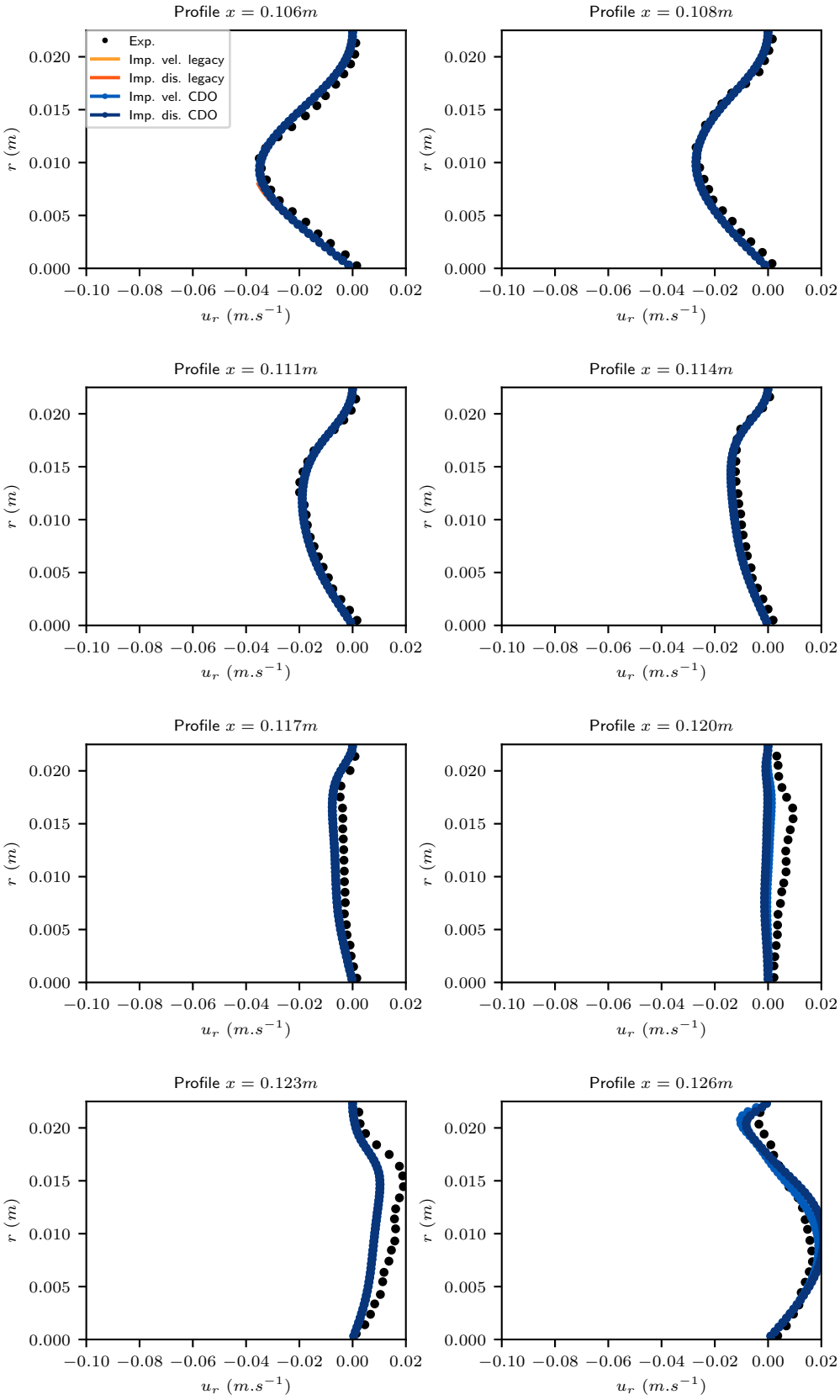


Figure 4.7: Radial velocity profiles at different positions x for the compression step ($x_p = 0.127$ m).

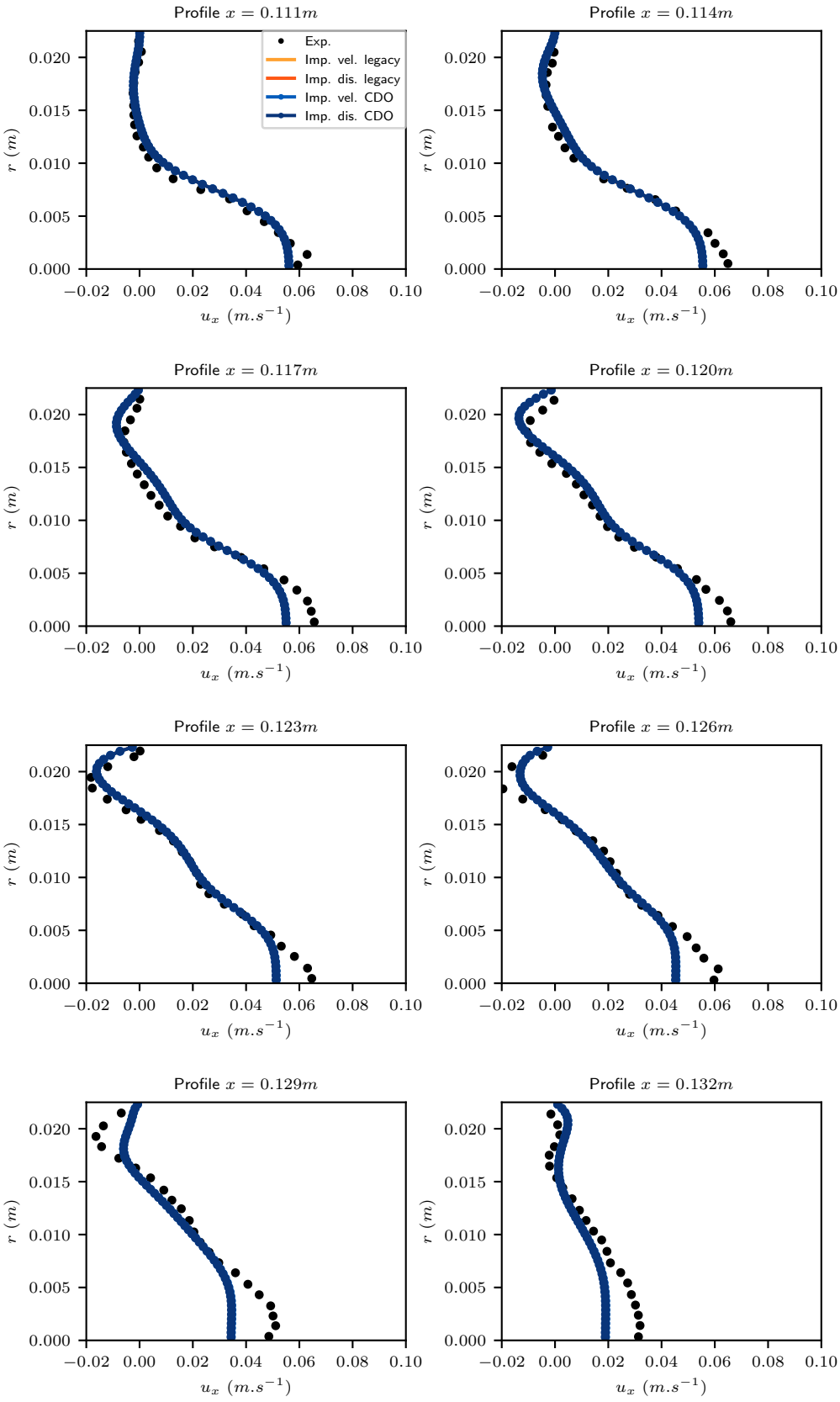


Figure 4.8: Axial velocity profiles at different positions x for the injection step ($x_p = 0.135$ m).

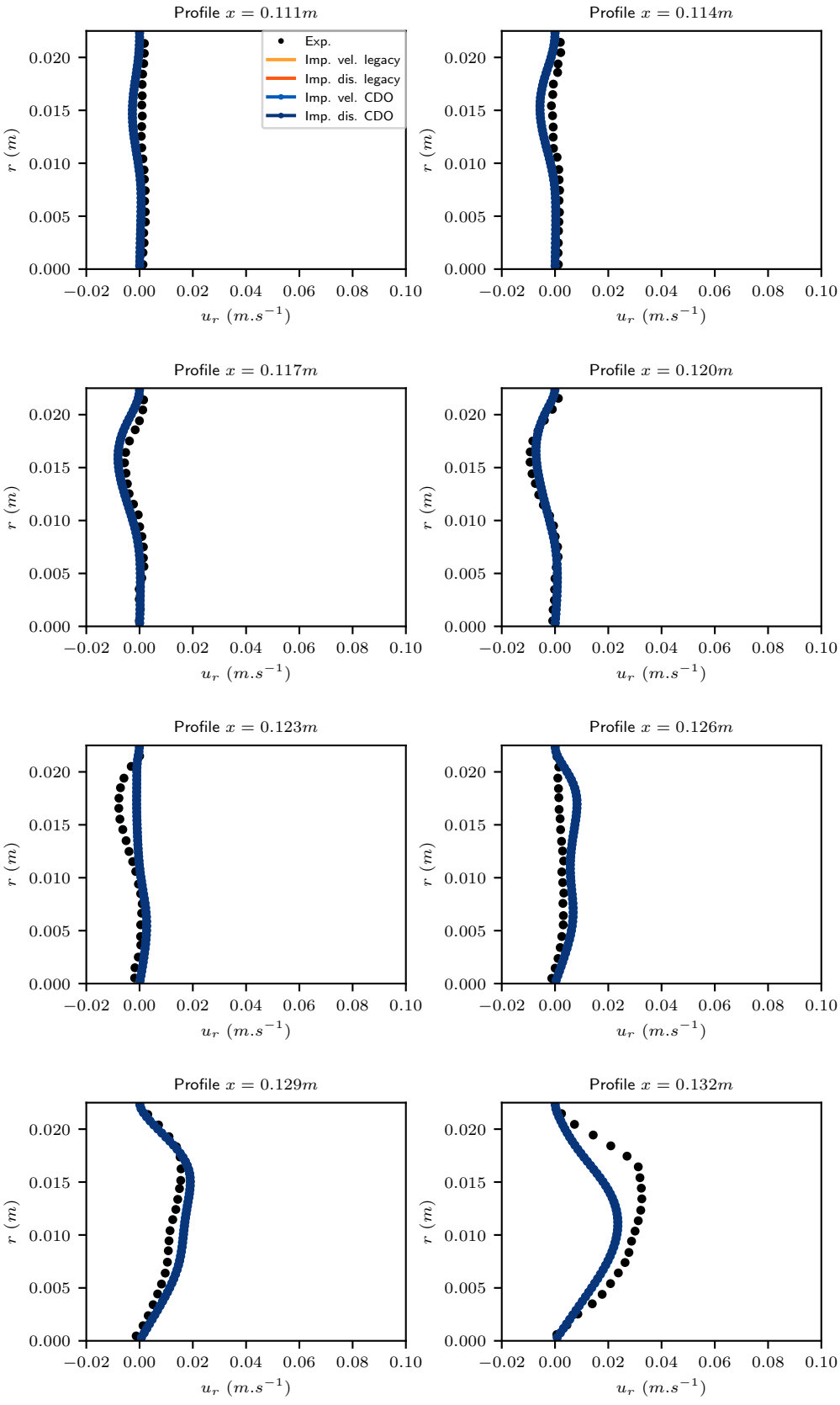
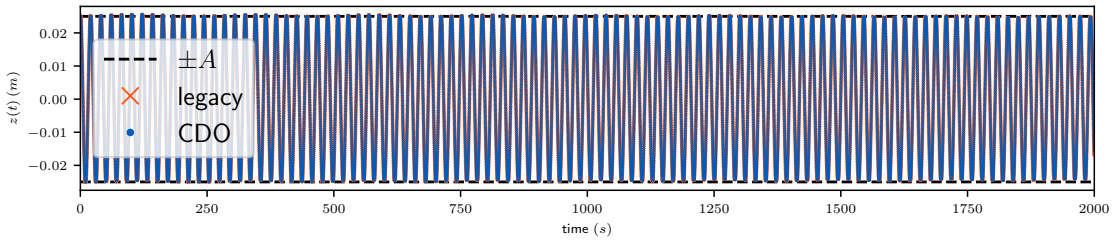
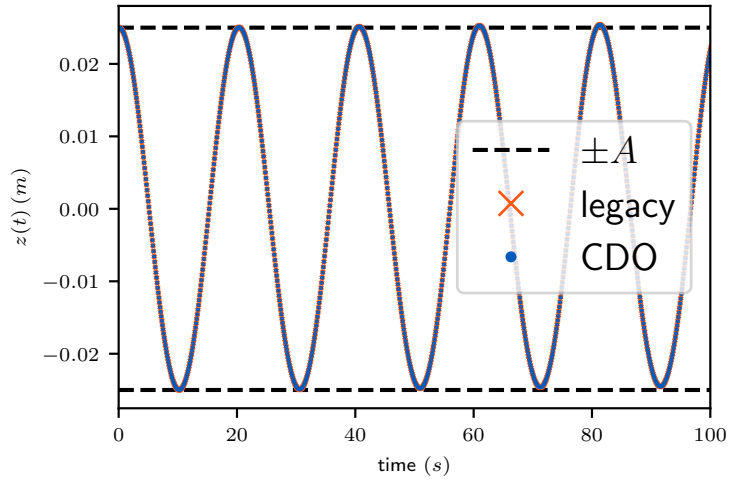


Figure 4.9: Radial velocity profiles at different positions x for the injection step ($x_p = 0.135$ m).



(a) 100 periods.



(b) Zoom over 5 periods.

Figure 4.10: Time series of wave elevation at the centre point in the case of a standing wave. The initial wave amplitude is given as $A = 0.025\,\text{m}$ for an average water depth of $h_0 = 10\,\text{m}$.

Nonlinear standing waves and the conservation of volume

When the amplitude A of the initial sinusoidal deformation is increased, linear theory is no longer valid, and other modes of oscillation can appear. The goal here is not to study the other modes, but to determine the validity of the numerical developments presented here. As the initial amplitude increases from $A = 0.025\,\mathrm{m}$ up to $A = 2.5\,\mathrm{m}$ (a case that was numerically unstable in earlier versions of code_saturne), we can consider the relative variation in volume over time.

We recall that from §4.3.3, this volume numerically does not necessarily remain constant using the legacy approach because of the interpolations between nodes and cell centres cause differences in how the mesh and fluid velocity are considered. On the contrary, the CDO approach shows a good total volume conservation over time. The error remains under the precision used to solve the mass equation.

In Figure 4.12, we see that the relative variation in volume for all amplitudes with the CDO approach does not exceed 10^{-9}, which is smaller than the mass solver precision (10^{-8} is the tolerance used here for the pressure solver). As expected, for larger wave amplitudes with the legacy approach, the loss of volume becomes larger, but this variation in volume seems to stabilise after an initial adjustment period. For example, for an amplitude of 2.5 m, the quasi-total loss of volume is made at the beginning of the simulation, when the wave of 2.5 m breaks into a number of other wavelengths.

The most non-linear case can also be used to see the effect of the filtering introduced with the CDO solver (see Figure 4.11).

4.4.3 Viscous damping of waves

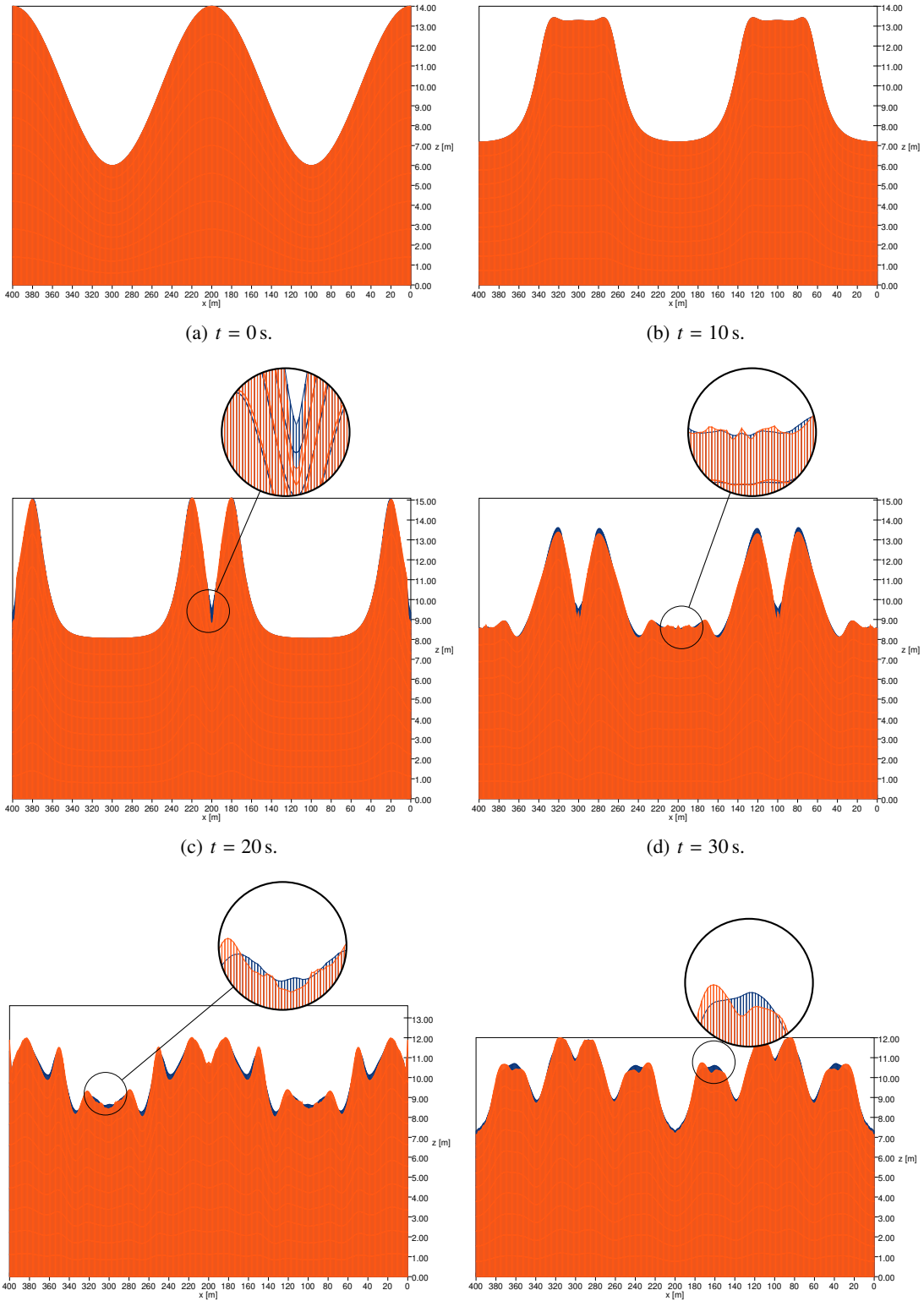
To verify that the application of these ALE schemes has not affected the ability of the algorithm to properly model viscous effects, we next consider a standing wave, similar to the previous section, but with viscosity added. We consider a wave which normally would be considered a deep water case, and then can use the solution from Lamb (1945) to describe the viscous decay and also the extension to arbitrary depths as in Antuono and Colagrossi (2013).

The domain we consider is again two-dimensional, 128×64 cells (in the x- and z-directions, respectively, with a width of $L = 2\,\mathrm{m}$ and depth $h_0 = 1\,\mathrm{m}$. The bottom boundary-condition is no-slip, but the side boundaries are free-slip. For this test case, we take $\rho = 1\,\mathrm{kg\,m^{-3}}$ and $g = 1\,\mathrm{m\,s^{-2}}$, to make all variables non-dimensional. A time-step of $\sqrt{\pi}/50\,\mathrm{s}$ was used, which corresponds to $\frac{1}{100}^{th}$ of the period of oscillation for an inviscid linear wave. Instead of having an initial free-surface elevation, here we impose on an undisturbed domain the velocity field of a standing Airy wave solution:

$$\boldsymbol{u}\,(\boldsymbol{x}, t = 0) = \begin{pmatrix} \omega A \sin kx \dfrac{\cosh\left(k(z + h_0)\right)}{\sinh\left(kh_0\right)} \\ 0 \\ -\omega A \cos kx \dfrac{\sinh\left(k(z + h_0)\right)}{\sinh\left(kh_0\right)} \end{pmatrix}. \tag{4.48}$$

Taking $kh_0 = \pi$, and a wave amplitude of $A = 0.005h_0$, the dispersion relationship, $\omega^2 = gk \tanh kh_0$, reduces to the deep-water case. After, we examine the evolution of the kinetic energy, $\mathcal{E}_K(t) = \int_\Omega ||\boldsymbol{u}||^2/2\mathrm{d}\Omega$, compared to the expected exponential decay, where $\mathcal{E}_K(t) \propto e^{-\beta\sqrt{\frac{g}{h_0}}t}\mathcal{E}_K(0)$ for a coefficient β.

In deep water, the viscous dissipation of waves due to the viscosity of the fluid has been studied since the work of Lamb (1945), who predicted $\beta = 4\pi^2/Re$. The exact rate of decay is



(a) $t = 0\,\mathrm{s}$. (b) $t = 10\,\mathrm{s}$.

(c) $t = 20\,\mathrm{s}$. (d) $t = 30\,\mathrm{s}$.

(e) $t = 40\,\mathrm{s}$. (f) $t = 50\,\mathrm{s}$.

Figure 4.11: Snapshots of free-surface for the filtered (in blue) and the non-filtered (in orange) mesh-velocity boundary condition for the standing wave test case with an initial $A = 4\,\mathrm{m}$ and an average depth of $h_0 = 10\,\mathrm{m}$; a close-up with three times magnification is added to emphasise the differences, mainly next to the peaks.

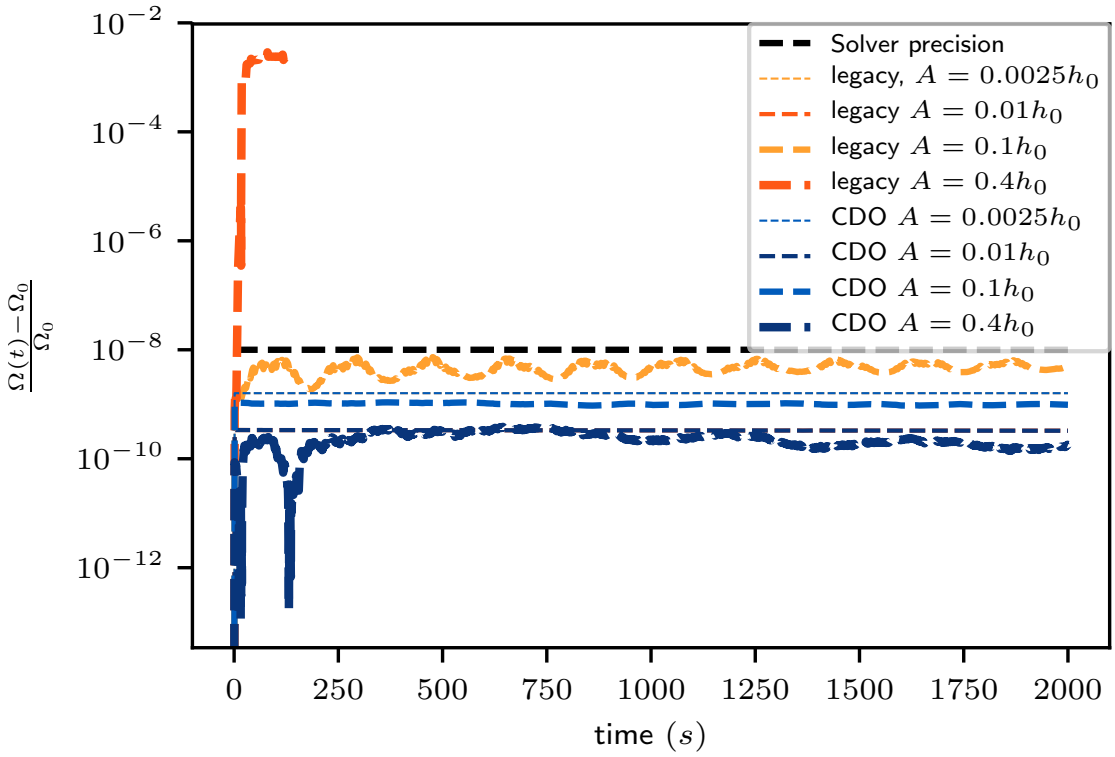


Figure 4.12: Relative change in volume in terms of time for the standing wave test case with different initial wave heights, ranging from $A = 0.0025h_0$ to $A = 0.4h_0$ with an average depth of $h_0 = 10\,\text{m}$. Dashed line shows the precision of the solver for pressure (10^{-8}).

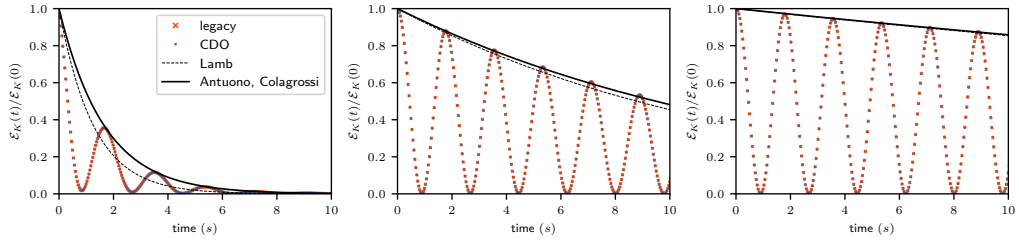


Figure 4.13: Evolution of the kinetic energy for $Re = 50$ (left), $Re = 500$ (centre), and $Re = 2500$ (right), showing the decay of energy as compared to the theoretical solution from Lamb (1945) and Antuono and Colagrossi (2013).

slightly different in this case, because for low Reynolds numbers (where $Re = h_0\sqrt{g h_0}/\nu$), this classic solution does not take into account the bottom boundary layer, and even though for inviscid waves, for wave-numbers $k h_0 \geq \pi$, we are in the deep-water regime, a more complete solution was provided by Antuono and Colagrossi (2013), who solved the linearised Navier–Stokes solution that included the effect of the bottom boundary layer, thus $\beta = 4\pi^2 Re^{-1} - 2\sqrt{2}\pi^{11/4} Re^{-1.5}$, and showed results that more closely match what is obtained here (Fig. 4.13). In this case, little difference is seen between legacy and CDO solvers, as the amplitude of motion is very small.

4.4.4 Solitary wave

In addition to handling a moving free-surface, for ocean engineering applications it is important to be able to generate waves, particularly steep waves. One form of this which can easily be studied mathematically is a solitary wave, where the wave translates over constant depth without changing shape.

Here we consider a domain $L = 40\,\mathrm{m}$ long, a depth $h_0 = 1\,\mathrm{m}$, and a wave with moderately large relative height $A/h_0 = 0.6$. Viscosity is reduced to a negligible value, so any changes to the wave shape in the domain should be due entirely to numerical errors. Again we consider a non-dimensional case, with $\rho = 1\,\mathrm{kg\,m^{-3}}$, $g = 1\,\mathrm{m\,s^{-2}}$, $h_0 = 1\,\mathrm{m}$.

On the wave-maker (left) boundary, the solution of the Euler equations is approximated using a highly accurate solution of the Korteweg–De Vries equation proposed by Dutykh and Clamond (2014) to impose the horizontal and vertical velocity fields as a boundary condition. In order to start with a wave tank initially at rest, the wave is shifted in time so that the wave crest passes $x = 0\,\mathrm{m}$ at $t = 8.0\,\mathrm{s}$.

On the other (right) side, an Orlanski-type radiative outlet is imposed, where all variables (i.e., velocity and pressure) are assumed to propagate at the theoretical celerity of the wave; see Orlanski (1976) or the code_saturne theory documentation Team (2021) for more detail. Note that this outlet condition is only useful for this case, where the solitary wave-speed is known. For more general cases where different waves propagate at different speeds, other damping methods are required, as discussed in the next application.

We are able to reproduce a solitary wave elevation within a few percent at a gauge in the middle of our domain (Fig. 4.14). The reflection is of similar magnitude, and likely due partly to an inaccurate wave celerity (i.e., the speed of the numerically modelled wave is not the same as the theoretical value, taken to be the convective outlet velocity).

In order to understand the errors associated with wave propagation, we consider three different grids, with 200×5 cells ($\Delta t = 0.01\sqrt{\frac{h_0}{g}}$), 400×10 cells ($\Delta t = 0.005\sqrt{\frac{h_0}{g}}$), and 800×20 cells ($\Delta t = 0.0025\sqrt{\frac{h_0}{g}}$). With $\Delta x = \Delta z = 0.1 h_0$, we get an arrival time (and thus wave-speed) error of less than 1% for both legacy and CDO solvers.

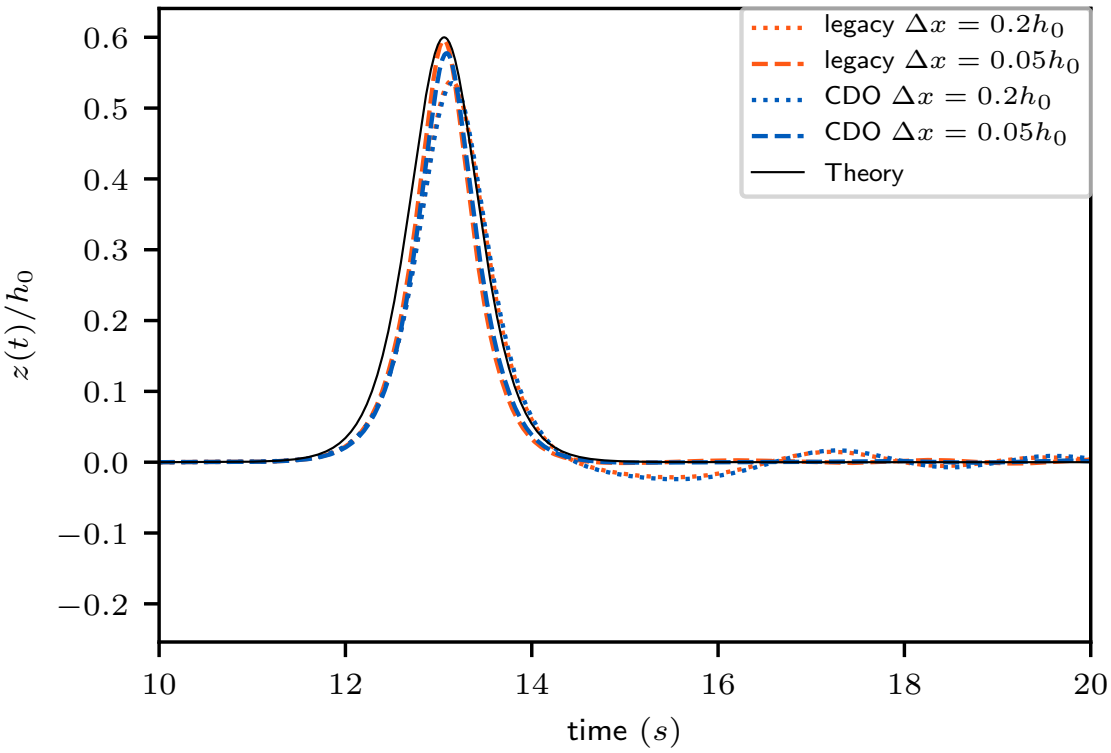


Figure 4.14: Wave elevation at $x = 20\,\text{m}$ over time for different discretisations (dotted: $\Delta x = \Delta z = 0.2h_0$; dashed: $\Delta x = \Delta z = 0.05h_0$), compared to the theoretical solution (solid line).

We also see a near linear convergence rate for the wave amplitude (Table 4.2), which is expected for the low-order method used here.

4.4.5 Wave propagation over a submerged bar

A common test of non-hydrostatic wave models is propagation over a submerged bar, which is beyond the range of a shallow-water assumption, or even some Boussinesq models (e.g., the $O(\mu)^2$ model of Wei et al. (1995)). We consider here the experimental results of Beji and Battjes (1993), comparing against time-series measured at various wave gauges.

In this test case, 17500 hexahedral cells are used to discretise the domain which is physically 30 m long, with a depth of 0.4 m, with a bar of minimum depth 0.1 m. For this case, a $k - \epsilon$ turbulence model will be used, in part to demonstrate that it does not negatively impact the propagation of the waves in non-breaking conditions.

Waves are absorbed on the right side through a surface pressure proportional to the vertical velocity at the surface, similar to Grilli and Horrillo (1997). Here, we impose a surface absorbing pressure, $p_{abs}(x,t)$ on the free-surface, which smoothly increases from zero inside the domain, up to some maximum, depending on the vertical velocity, u_z, at the free-surface:

$$p_{abs}(x,t) = \begin{cases} C_{AB}(\rho u_z \sqrt{gh}) \left(\frac{x - x_{AB}}{h}\right)^2 & x > x_{AB} \\ 0 & x \leq x_{AB} \end{cases}, \tag{4.49}$$

where here $x_{AB} = 20\,\text{m}$ and $C_{AB} = 0.003$ (chosen by trial-and-error).

On the left side, the velocity is imposed as a boundary condition, using second-order Stokes wave theory, similar to Ma et al. (2012), to ensure mass conservation, with a wave period of 2.5 s.

Scheme	$\frac{\Delta x}{h_0} = \frac{\Delta z}{h_0}$	Crest	Rel. error	Arrival time	Rel. error	Reflection	Rel. error
Legacy	1/5	0.536	-10.75%	16.31	1.94%	0.064	10.67%
	1/10	0.573	-4.51%	16.09	0.53%	0.062	10.36%
	1/20	0.594	-1.05%	16.00	0.01%	0.064	10.58%
CDO	1/5	0.540	-9.93%	16.28	1.75%	0.037	6.25%
	1/10	0.570	-5.00%	16.10	0.62%	0.037	6.22%
	1/20	0.583	-2.92%	16.06	0.34%	0.045	7.52%

Table 4.2: Errors in wave-elevation at $x = 20\,\text{m}$ for solitary wave propagation test case at different grid resolutions.

Two different test conditions are considered; one for non-breaking waves, and one with plunging waves. The incident wave parameters were chosen to best represent the signal recorded at the first wave gauge, at the toe of the submerged bar, in order to best match experimental conditions and imprecision in the physical wavemaker.

For the non-breaking condition, a 2.9 cm incident wave-height was considered in the basin. Looking at the wave elevation over time at both this wave-gauge (WG1, at $x = 6\,\text{m}$), as well as at the front and back of the shallowest region of the submerged bar (WG3 and WG5, at $x = 12\,\text{m}$ and $x = 14\,\text{m}$), we see that the waves are well-represented (Fig. 4.15). For this case, the differences between legacy and CDO solvers are negligible.

For the plunging wave condition, 5.4 cm incident wave is considered for the same setup. Like in the previous case, we are able to verify that the incident wave elevation at the toe of the bar is well represented by the code, and while the peak elevation is overpredicted, the overall form of the time-series is also captured over the bar (Fig. 4.16). Initially this is unexpected, as with the ALE approach we do not consider an overturning wave, but others (e.g., Zijlema and Stelling, 2008; Bradford, 2011; Ma et al., 2012) have observed that non-hydrostatic models are able to reproduce correctly some aspects of breaking waves. Notably, we see here that the turbulent kinetic energy and dissipation rate show that the turbulence model is most active exactly in the region where breaking waves are expected (Fig. 4.17), though clearly the exact dynamics of the free-surface are not captured.

Note that a homogeneous Neumann condition is used at the free-surface, but as the scale of the vortices (and thus turbulent viscosity) is reduced by the effect of the free-surface, in general a more sophisticated free-surface boundary condition is needed for the turbulent quantities (see Celik and Rodi, 1984, for more details).

One criticism of general-purpose CFD models for the applications being considered here may be their computational speed in comparison with specialised wave propagation models. As the submerged bar test case has frequently been considered for validation and demonstrating the dispersive properties of wave models, this can be used as a basis of comparison. For this we use the 3D code NHWAVE Ma et al. (2012), with an identical grid and identical starting timestep of $1/256^{th}$ of one wave period. NHWAVE uses an adaptive timestep in function of CFL number, but this does not vary significantly during this test case, so the two models should require a

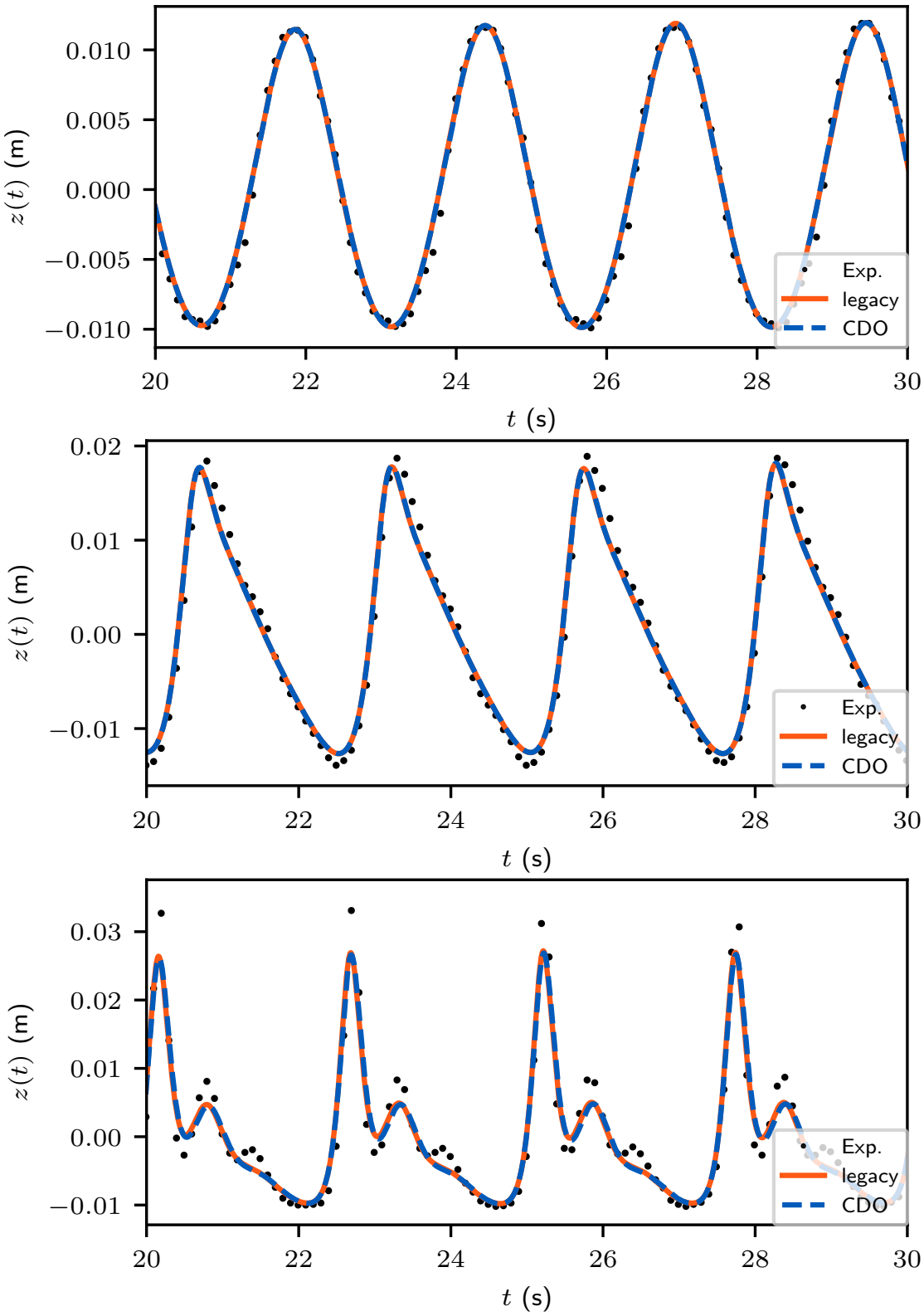


Figure 4.15: Wave elevation time-series at the toe of a submerged bar (at $x = 6\,\text{m}$) and at the front (at $x = 12\,\text{m}$) and back (at $x = 14\,\text{m}$) of the top of the bar for non-breaking wave conditions, compared to Beji and Battjes (1993).

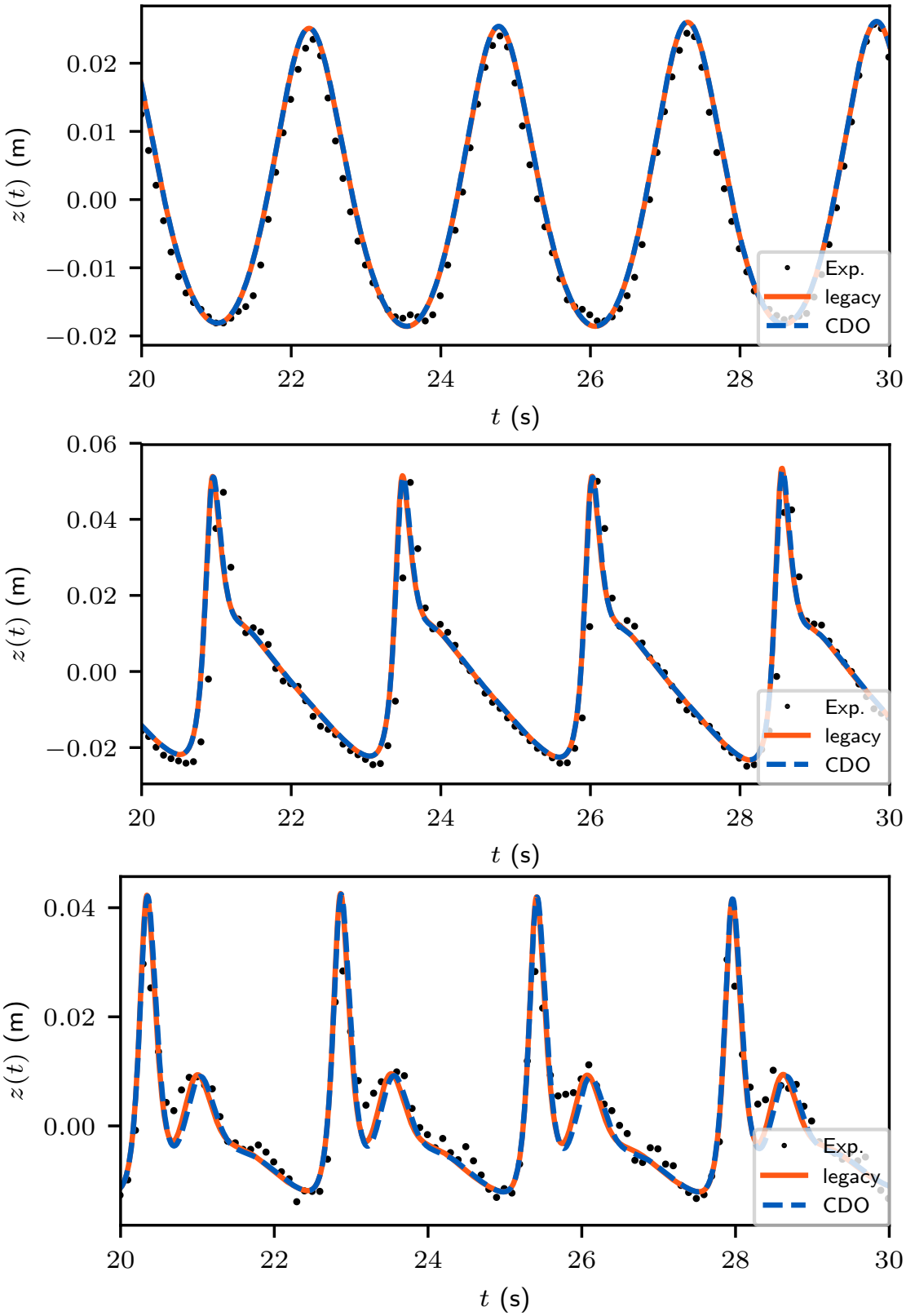


Figure 4.16: Wave elevation time-series at the toe of a submerged bar (at $x = 6\,\text{m}$) and at the front (at $x = 12\,\text{m}$) and back (at $x = 14\,\text{m}$) of the top of the bar for plunging wave conditions, compared to Beji and Battjes (1993).

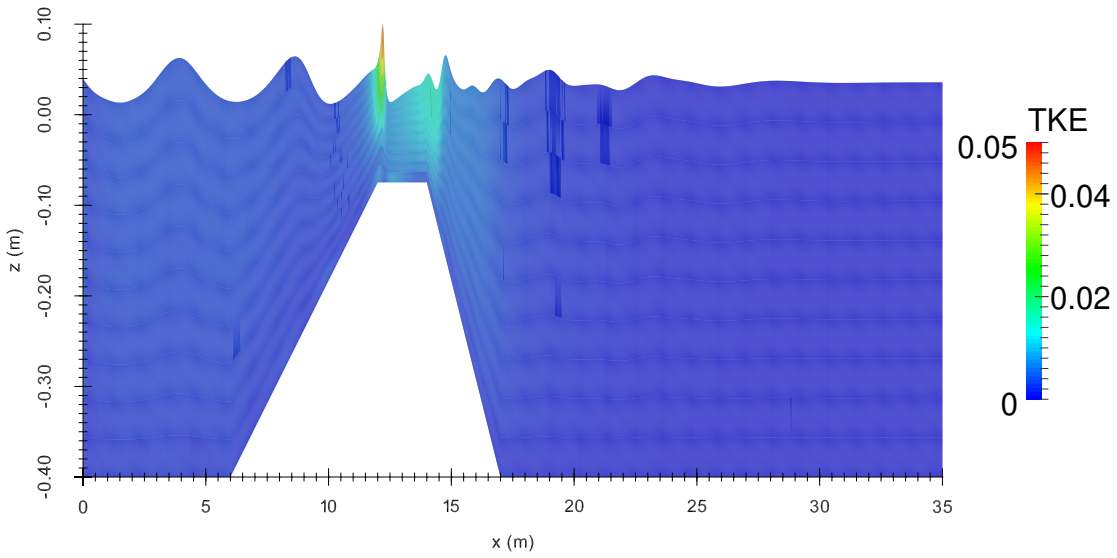


Figure 4.17: Turbulent kinetic energy (m^2/s^2) over a submerged bar under plunging wave conditions, corresponding to the experimental test case of Beji and Battjes (1993), at $t = 31.2594$ s.

similar amount of computational effort. To minimize variations between the tests, results were obtained on a single Intel Core `i7` processor, with 645.0 s for code_saturne with the legacy scheme of code_saturne (and 2680.0 s with the unoptimised conjugate gradient linear solvers for CDO scheme) versus 559.0 s for NHWAVE. This is not meant to be an exact comparison between the two codes, as NHWAVE can benefit from higher-order approaches which are not present in code_saturne, and code_saturne can be used on types of grids which are not available with NHWAVE, but merely to show that code_saturne can be used to provide a reasonable performance.

4.4.6 Forces on a vertical cylinder in regular waves

A common application of wave-models is the understanding of forces on bodies. Here we consider the nonlinear forces exerted on a vertical bottom-mounted cylinder (i.e., a monopile) in regular waves. Huseby and Grue (2000) conducted a range of experiments which has commonly been used for benchmarking numerical tools (Ferrant, 2000; Shao and Faltinsen, 2014) for the forces of periodic forces on a cylinder of radius $R = 3$ cm, in a depth of 0.6 m, at a range of wave conditions. Here we consider the case with $kR = 0.245$, and a wave steepness of $kA = 0.10$. For this application, no turbulence model was used, but a viscosity of 1×10^{-5} m^2/s was applied, slightly higher than the molecular viscosity, in order to stabilize the model and account for some viscous effects.

The grid used consists of 77406 hexahedral cells in a domain of width 0.5 m and length 3.5 m, with the cylinder positioned in the centre of the domain, with a time-step of $T/100$. Second-order Stokes waves are used to impose the incoming waves, and the same damping setup as the previous case (Equation 4.49) is re-used here, with $x_{damp} = 2.5$ m and $C_{damp} = 0.25$. The simulations are performed for 10 wave periods T, with a quasi-steady state being reached for $t > 7T$. A short-time Fourier transform is used to evaluate the convergence of the integrated force on the cylinder in the x-direction, $f(t)$ (Fig. 4.18):

$$F_{(m)}(t) = \frac{2}{T} \int_t^{t+T} F(t)\, e^{im\omega\tau} \mathrm{d}\tau, \tag{4.50}$$

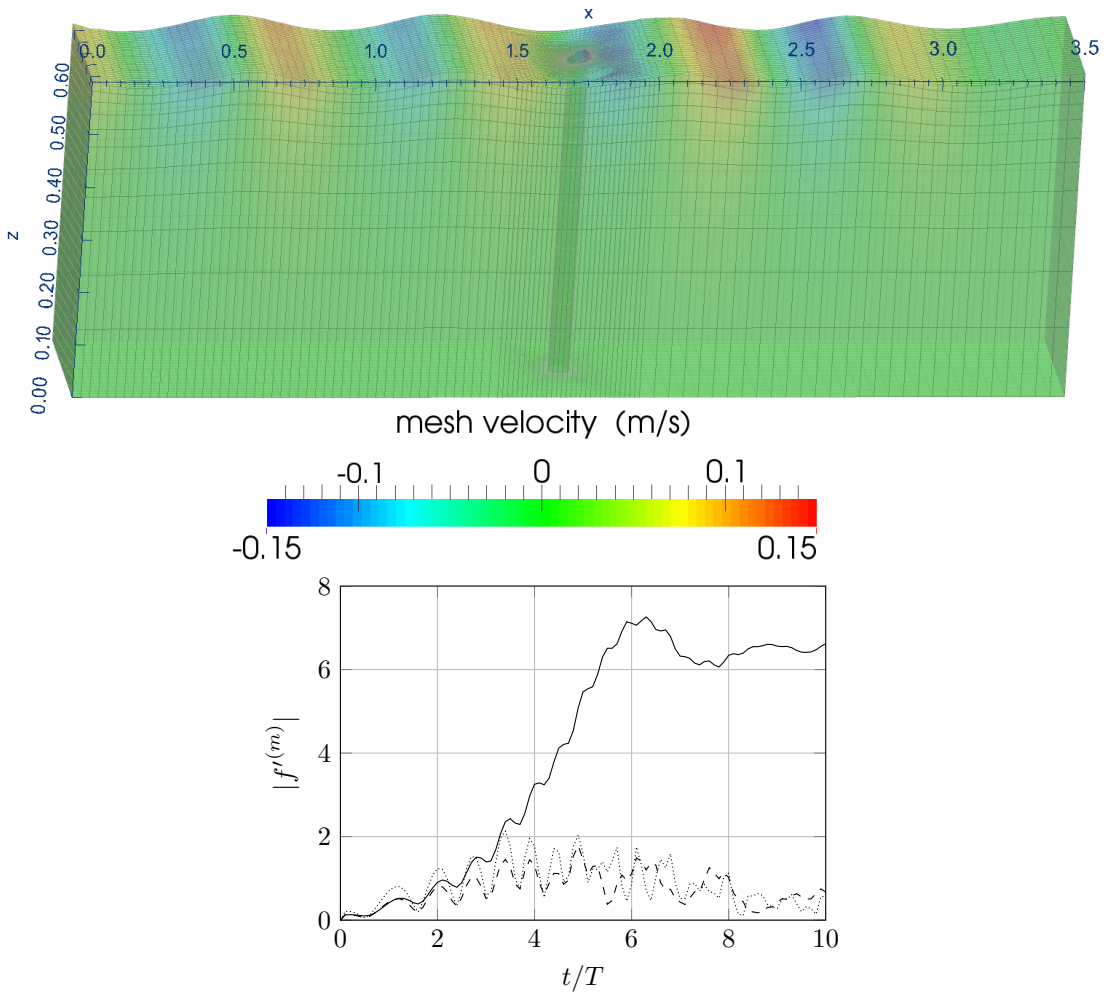


Figure 4.18: A vertical cylinder in regular waves (upper panel, at $t = 10T$), and the corresponding variation in time of the first three harmonics of the horizontal force, as evaluated by a short-time Fourier transform.

and we see that the computed force compares well to the experimentally determined values (Table 4.3), using data from four periods, after a spin-up time of 7 periods has passed.

Further study of different amplitudes showed that local instabilities near the monopile are found at higher incident wave-heights, likely due to local wave breaking or an importance of local viscous effects that require a finer mesh than considered here. Here the legacy solver was used for the presented calculations, though little change was seen with the CDO solver.

4.4.7 Bottom-tilting wave-maker

In the ocean, tsunamis exhibit extremely long wavelengths that can be difficult to realistically replicate in a physical wave tank or basin, which are normally designed to handle conditions representative of wind-waves. While there has been extensive studies with solitary waves, it is known that solitary waves do not correctly represent the waves which are actually experienced in a tsunami, as shown by Madsen et al. (2008). As a result, Lu et al. (2017) have developed a bottom-tilting wave maker, testing on a small wave tank, 2 m long (see Fig. 4.19a; i.e., $L = 1$ m). This provides a validation case where one of the boundaries is thus moving.

This 1 m long flap is either starting from a lowered position, moving up, or starting from a

	code_saturne	Exp.
$\lvert F_1\rvert/(\rho g A R^2)$	6.42	6.45
$\lvert F_2\rvert/(\rho g A^2 R)$	0.41	0.40
$\lvert F_3\rvert/(\rho g A^3)$	0.29	0.37

Table 4.3: Harmonic components of the surge force on a monopile in deep water, with $kR = 0.245$ and wave amplitude $kA = 0.1$, comparing with the experiments of Huseby and Grue (2000).

raised position, moving down, and stopping when the bed is flat. One can then propagate the waves over a long time, reflecting between two walls, even from a very small tank. Numerically, this is described such that the depth, $h(x,t)$, is:

$$h(x,t) = \begin{cases} h_0 & x > L \\ & \text{or } t > \frac{bL}{\sqrt{gh_0}}, \\ h_0 + \frac{L-x}{L}(ah_0)[1 - \frac{t\sqrt{gh_0}}{Lb}] & x < L \\ & \text{and } t < \frac{bL}{\sqrt{gh_0}}, \end{cases} \tag{4.51}$$

where the prescribed motion has a non-dimensional amplitude a and time b. For validation, we consider the results for a resting water depth of h_0=5 cm, with a duration $b = 1.4$, and an amplitude $a = \pm 0.8$.

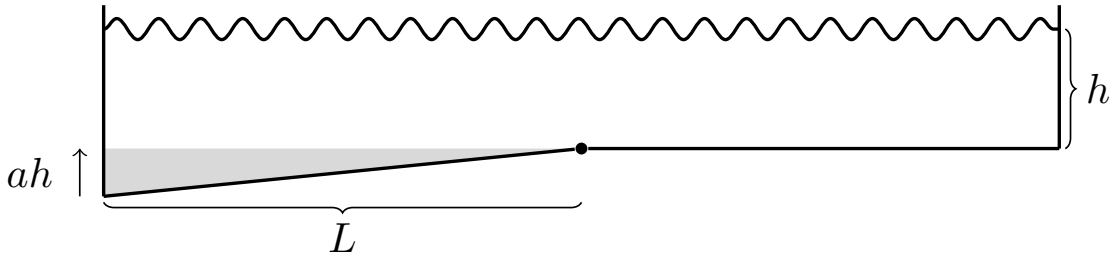
The grid is discretised with 5 vertical levels and 100 cells in the lengthwise direction and 11 cells in the spanwise direction (i.e., $\Delta x = 2$ cm, $\Delta y = 1$ cm, and $\Delta z \approx 1$ cm), and a time-step of 0.009 s is used. It is assumed that all boundaries are smooth walls. The flap motion itself is imposed by fixing the mesh velocity.

The wave elevation was measured at a gauge at the centre of the tank (i.e., directly over the hinge), and we are able to reproduce the experimental values to a reasonable degree of accuracy, for both upward (Fig. 4.19b) and downward (Fig. 4.19c) motion.

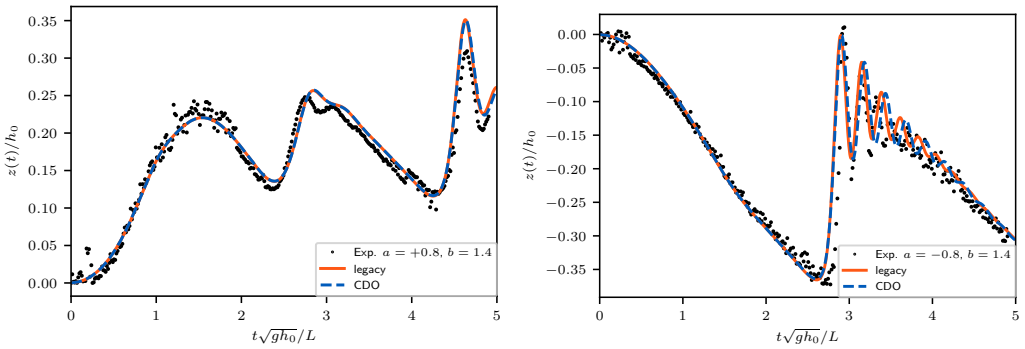
4.5 Summary

Here we present recent developments of the ALE module of code_saturne for the simulation of free-surface flows, and then the results of applying the resulting model to ocean waves. Although certainly there are limitations to using a finite volume ALE model such as code_saturne in terms of wave steepness, we can see that for certain circumstances, understanding the limitations of numerical dissipation and lower-order elements, a general CFD solver not specialised for water wave propagation can produce useful results in reasonable amount of time. Validation tests were presented for multiple setups, showing that the model is able to exactly conserve mass with the vertex-based solver on the dual barycentric mesh, correctly conserve energy, propagate waves, and compute forces on structures.

In future work, comparisons will be made between the ALE module and the recently released VOF module, which will enable the complex modelling of breaking waves, but at an increased



(a) Bottom-tilting wave-maker schematic.



(b) Upward initial movement.

(c) Downward initial movement.

Figure 4.19: Bottom-tilting wave-maker schematic, and the wave profiles that are generated by an initial movement, compared to the experiments of Lu et al. (2017).

computational cost. In order to offset this increase in computational time, a coupling between code_saturne and far-field inviscid models will be considered.

Bibliography

Antuono, M., Colagrossi, A., 2013. The damping of viscous gravity waves. Wave Motion 50, 197–209.

Archambeau, F., Méchitoua, N., Sakiz, M., 2004. Code_saturne: a finite volume method for the computation of turbulent incompressible flows - industrial applications. International Journal on Finite Volumes 1, 1–62.

Bahlali, M.L., Henry, C., Carissimo, B., 2020. On the well-mixed condition and consistency issues in hybrid Eulerian/Lagrangian stochastic models of dispersion. Boundary-Layer Meteorology 174, 275–296.

Beji, S., Battjes, J.A., 1993. Experimental investigation of wave-propagation over a bar. Coastal Engineering 19, 151–162.

Benoit, M., Dias, F., Herterich, J., Scolan, Y.M., 2018. Un cas-test discriminant pour la simulation de la propagation et du run-up de trains de vagues de type tsunami, in: Proceedings of 16th Journées de l'Hydrodynamique, p. 14.

Benson, D.J., 1989. An efficient, accurate, simple ALE method for nonlinear finite element programs. Computer Methods in Applied Mechanics and Engineering 72, 305–350.

Bonelle, J., Ern, A., 2014. Analysis of Compatible Discrete Operator schemes for elliptic problems on polyhedral meshes. ESAIM Mathematical Modelling and Numerical Analysis 48, 553–581.

Boscheri, W., Balsara, D.S., Dumbser, M., 2014. Lagrangian ADER-WENO finite volume schemes on unstructured triangular meshes based on genuinely multidimensional HLL Riemann solvers. Journal of Computational Physics 267, 112–138.

Bradford, S.F., 2011. Nonhydrostatic model for surf zone simulation. Journal of waterway, port, coastal, and ocean engineering 137, 163–174.

Carré, G., Del Pino, S., Després, B., Labourasse, E., 2009. A cell-centered lagrangian hydrodynamics scheme on general unstructured meshes in arbitrary dimension. Journal of Computational Physics 228, 5160–5183.

Celik, I., Rodi, W., 1984. Simulation of free-surface effects in turbulent channel flows. Physiochemical Hydrodynamics 5, 217–227.

Chan, R., Street, R., 1970. Computer study of finite-amplitude water waves. Journal of Computational Physics 6, 68–94.

Chemartin, L., Lalande, P., Montreuil, E., Delalondre, C., Cheron, B., Lago, F., 2009. Three dimensional simulation of a DC free burning arc. Application to lightning physics. Atmospheric Research 91, 371–380.

Chen, L., Zang, J., Hills, A., Morgan, G., Plummer, A., 2014. Numerical investigation of wave-structure interaction using OpenFOAM. Ocean Engineering 88, 91–109.

Colas, C., Ferrand, M., Hérard, J.M., Latché, J.C., Le Coupanec, E., 2019. An Implicit Integral Formulation to Model Inviscid Fluid Flows in Obstructed Media. Computers & Fluids 188, 136–163.

Dal Secco, S., Juan, O., Louis-Louisy, M., Lucas, J.Y., Plion, P., Porcheron, L., 2015. Using a genetic algorithm and CFD to identify low NOx configurations in an industrial boiler. Fuel 158, 672–683.

Donea, J., Huerta, A., Ponthot, J.P., guez Ferran, A.R., 2004. Arbitrary Lagrangian-Eulerian Methods, in: Stein, E., de Borst, R., Hughes, T.J.R. (Eds.), Encyclopedia of Computational Mechanics. John Wiley & Sons. chapter 14, pp. 413–437.

Durst, F., Maxworth, T., Pereira, J.C.F., 1989. Piston-driven, unsteady separation at a sudden expansion in a tube: Flow visualization and LDA measurements. Physics of Fluids A 1, 1249.

Dutykh, D., Clamond, D., 2014. Efficient computation of steady solitary gravity waves. Wave Motion 51, 86–99.

Eymard, R., Gallouët, T., Guichard, C., Herbin, R., Masson, R., 2014. TP or not TP, that is the question. Computational Geosciences 18, 285–296. doi:10.1007/s10596-013-9392-9.

Farhat, C., Geuzaine, P., Grandmont, C., 2001. The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids. Journal of Computational Physics 174, 669–694.

Ferrand, M., Fontaine, J., Angelini, O., 2014. An anisotropic diffusion finite volume algorithm using a small stencil, in: Finite Volumes for Complex Applications VII-Elliptic, Parabolic and Hyperbolic Problems, pp. 577–585.

Ferrand, M., Harris, J.C., 2021. Finite volume arbitrary lagrangian-eulerian schemes using dual meshes for ocean wave applications. Computers & Fluids 219, 104860.

Ferrant, P., 2000. Fully nonlinear interactions of long-crested wave packets with a three-dimensional body, in: 22nd Symposium on Naval Hydrodynamics, pp. 403–416.

Gauffre, M.C., Benhamadouche, S., Badel, P.B., 2020. Wall-Modeled Large Eddy Simulation of the Flow Through PWR Fuel Assemblies at Re_H = 66 000 - Validation on CALIFS Experimental Setup. Nuclear Technology 206, 255–265.

Grilli, S.T., Horrillo, J., 1997. Numerical generation and absorption of fully nonlinear periodic waves. Journal of Engineering Mechanics 123, 1060–1069.

Harlow, F.H., Welch, J.E., 1965. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. Physics of Fluids 8, 2182–2189.

Helluy, Philippe, Golay, Frédéric, Caltagirone, Jean-Paul, Lubin, Pierre, Vincent, Stéphane, Drevard, Deborah, Marcer, Richard, Fraunié, Philippe, Seguin, Nicolas, Grilli, Stephan, Lesage, Anne-Cécile, Dervieux, Alain, Allain, Olivier, 2005. Numerical simulations of wave breaking. ESAIM: M2AN 39, 591–607. URL: https://doi.org/10.1051/m2an:2005024, doi:10.1051/m2an:2005024.

Hervouet, J.M., 2007. Hydrodynamics of free surface flows: modelling with the finite element method. Wiley and Sons, Ltd.

Hirt, C.W., Amsden, A.A., Cook, J.L., 1974. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. Journal of Computational Physics 14, 227–253.

Hirt, C.W., Nichols, B.D., 1981. Volume of fluid (VOF) method for the dynamics of free boundaries. Journal of Computational Physics 39, 201–225.

Huseby, M., Grue, J., 2000. An experimental investigation of higher-harmonic wave forces on a vertical cylinder. Journal of Fluid Mechanics 414, 75–103.

Issa, R.I., Gosman, A., Watkins, A., 1986. The computation of compressible and incompressible recirculating flows by a non-iterative implicit scheme. Journal of Computational Physics 62, 66–82.

Jacobsen, N.G., Fuhrman, D.R., Fredsøe, J., 2012. A wave generation toolbox for the open-source CFD library: OpenFOAM. International Journal for Numerical Methods in Fluids 70, 1073–1088.

Lamb, H., 1945. Hydrodynamics. Dover Books.

Lin, P., 2008. Numerical modeling of water waves. CRC Press.

Longuet-Higgins, M.S., Cokelet, E., 1976. The deformation of steep surface waves on water – I. A numerical method of computation. Proceedings of the Royal Society A 350, 1–26.

Lu, H., Park, Y.S., Cho, Y.S., 2017. Modeling of long waves generated by bottom-tilting wave maker. Coastal Engineering 122, 1–9.

Ma, G., Shi, F., Kirby, J.T., 2012. Shock-capturing non-hydrostatic model for fully dispersive surface wave processes. Ocean Modelling 43-44, 22–35.

Ma, L., Nmira, F., Consalvi, J., 2019. Verification and validation of a variable–density solver for fire safety applications. Numerical Heat Transfer, Part B: Fundamentals 76, 107–129.

Madsen, P.A., Fuhrman, D.R., Schäffer, H.A., 2008. On the solitary wave paradigm for tsunamis. Journal of Geophysical Research 113, C12012.

Minier, J.P., 2016. Statistical descriptions of polydisperse turbulent two-phase flows. Physics Reports 665, 1–122.

Mola, A., Heltai, L., DeSimone, A., 2013. A stable and adaptive semi-Lagrangian potential model for unsteady and nonlinear ship-wave interactions. Engineering Analysis with Boundary Elements 37, 128 – 143. URL: http://www.sciencedirect.com/science/article/pii/S0955799712001907, doi:https://doi.org/10.1016/j.enganabound.2012.09.005.

Orlanski, I., 1976. A simple boundary condition for unbounded hyperbolic flows. Journal of Computational Physics 21, 251 – 269. doi:https://doi.org/10.1016/0021-9991(76)90023-1.

Re, B., Dobryzynski, C., Guardone, A., 2017. An interpolation-free ALE scheme for unsteady inviscid flows computations with large boundary displacements over three-dimensional adaptive grids. Journal of Computational Physics 340, 26–54.

Shao, Y.L., Faltinsen, O.M., 2014. A harmonic polynomial cell (HPC) method for 3D Laplace equation with application in marine hydrodynamics. Journal of Computational Physics 274, 312–332.

Storti, M.A., Garelli, L., Paz, R.R., 2012. A finite element formulation satisfying the discrete geometric conservation law based on averaged Jacobians. International Journal for Numerical Methods in Fluids 69, 1872–1890. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.2669, doi:10.1002/fld.2669, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.2669.

Team, D., 2021. code_saturne 7.0 Theory Guide. EDF R&D. URL: https://code-saturne.org/cms/sites/default/files/docs/7.0/theory.pdf.

Teles, M.J., Pires-Silva, A.A., Benoit, M., 2013. Numerical modelling of wave current interactions at a local scale. Ocean Modelling 68, 72–87.

Thomas, P.D., Lombard, C.K., 1979. Geometric conservation law and its application to flow computations on moving grids. AIAA Journal 17, 1030–1037. doi:10.2514/3.61273, arXiv:https://doi.org/10.2514/3.61273.

Versteeg, H.K., Malalasekera, W., 2007. An introduction to computational fluid dynamics the finite volume method. 2^{nd} ed., Pearson Education Ltd., Harlow (GB) New York Boston [etc.]. URL: http://www.sudoc.fr/11762473X.

Wei, G., Kirby, J.T., Grilli, S.T., Subramanya, R., 1995. A fully nonlinear Boussinesq model for surface waves. Part 1. Highly nonlinear unsteady waves. Journal of Fluid Mechanics 294, 71–92.

Wroniszewski, P.A., Verschaeve, J.C.G., Pedersen, G.K., 2014. Benchmarking of Navier-Stokes codes for free surface simulations by means of a solitary wave. Coastal Engineering 91, 1–17.

Zhang, H., Reggio, M., Trépanier, J., Camarero, R., 1993. Discrete form of the GCL for moving meshes and its implementation in CFD schemes. Computers & Fluids 22, 9 – 23. URL: http://www.sciencedirect.com/science/article/pii/004579309390003R, doi:https://doi.org/10.1016/0045-7930(93)90003-R.

Zijlema, M., Stelling, G.S., 2008. Efficient computation of surf zone waves using the nonlinear shallow water equations with non-hydrostatic pressure. Coastal Engineering 55, 780–790.

5. Conclusion

This manuscript first introduced some basics for studying the fluid motion in presence of a free-surface. As a material surface, a free-surface is naturally studied using Lagrangian approach. Balance equations written in conservative form lead to discrete local balance on finite volume. These balances can be performed on moving volumes with an arbitrary Euler / Lagrange view, and a classical prediction – correction algorithm is proposed to time discretised the set of equations. The SPH method in its Lagrangian form is introduced and derived from the Lagrange equations. One of the key issue of the SPH method is to deal with (Eulerian) wall boundary conditions or open (Eulerian) inlet – outlet boundaries.

Chapter 2 introduced an approach to deal with walls with the SPH method which is both simple and robust. The simplicity lies in the manner we compute the Eulerian renormalisation term γ_a with an integration in time which only requires the analytical computation of its gradient $\boldsymbol{\nabla}\gamma_a$. The robustness is due to the integration in time of the continuity equation which makes the density field depend only on the particles' positions. This allows long time simulations with a relatively larger time step and is a major advantage for conservation properties. The definition of new boundary corrected differential operators (*gradient*, *Laplacian*) allows one to fix boundary conditions and fluxes on the pressure field, the wall shear stress but also any scalar fields such as k and ϵ in a turbulence model.

Chapter 3 extended the semi-analytical approach in the SPH method to derive boundary terms for unsteady open boundaries which can be of complex shape. This is performed using the following main ingredients: divergence operator is boundary corrected, mass of boundary particles is made time-variable and they are used to create new fluid particles for entering flow and Riemann problem is solved to correctly impose the state on the boundary. A special care is also given to the time integration of the mass equation.

Boundary condition presented in this manuscript can be extended to 3-D, and one of the main challenge is to efficiently compute the renormalisation factor gradient. An algorithm is proposed in Appendix D.4. An extension to axisymmetric formulation is also proposed in Appendix D.5 and gives encouraging results compared to the really few algorithms proposed in the literature.

Finally, the finite volume method in its ALE formulation is applied to (gentle) free-surfaces, using an original mixing between a cell-based scheme for fluid conservation laws and a vertex-based CDO scheme to compute nodes displacements. A special care to free-surface conditions at the discrete level is given and the dual scheme is used with advantages to filter short-waves modes.

Some insights are given to the space discretisation of the cell-based Poisson problem in Appendix A and extension of the present algorithms to second time-order is proposed in Appendix C.

A. An anisotropic diffusion finite volume algorithm using a small stencil

Abstract A.1 This chapter presents a finite volume algorithm to solve anisotropic heterogeneous diffusion equations within the open source CFD software code_saturne. This algorithm has the advantage to use a small stencil composed of face neighbouring cells only, which makes it easy to parallelise. The resolution is performed through an iterative process (fixed point Picard algorithm). Second order convergence in space is numerically obtained on various analytical test-cases and mesh sequences of the FVCA6 benchmark and the results are compared to the barycentric version of the SUSHI scheme Eymard et al. (2010).

This chapter is a reproduction of Ferrand et al. (2014). A description of a conjugate-gradient type algorithm to solve the process is added.

Résumé A.1 Ce chapitre présente un algorithme de volumes finis pour résoudre des problèmes de diffusion hétérogène anisotrope. Cet algorithme est intégré au logiciel open-source code_saturne. Il présente l'avantage de ne faire appel qu'à un petit stencil composé uniquement, pour une cellule donnée, des cellules voisines par les faces, ce qui le rend facilement parallélisable (comme le reste des algorithmes historiques de code_saturne). La résolution du problème est faite au travers d'un algorithme de type point fixe de Picard. Une convergence d'ordre deux en espace est obtenue numériquement sur quelques cas tests et différentes séries de maillages du *benchmark FVCA6* (ces résultats sont comparés au schéma SUSHI barycentrique Eymard et al. (2010)).

Ce chapitre est une reproduction de Ferrand et al. (2014) auquel s'ajoute la description de l'algorithme de type gradient conjugué que l'on peut activer pour résoudre le processus itératif de façon accélérée.

A.1 Introduction

Several discretisation schemes for anisotropic, heterogeneous diffusion problems are presented in the literature, especially with the finite volume method (see Eymard et al. (2000), or the 2-D Svyatskiy (2008) and 3-D Eymard et al. (2011) benchmarks). In this Appendix, we propose a two-point flux approximation (TPFA) scheme for non-Cartesian grids within an industrial code called code_saturne (see Team (2021) for more information), applied to various steps in the core

solver (diffusion part of scalar transport equation with Generalised Gradient Diffusion Hypothesis (GGDH) Dehoux et al. (2017), projection step of the predictor-corrector Navier-Stokes solver in presence of head-losses, etc.). Therefore, the proposed scheme is aimed to be highly parallelised, with a ghost-cells technique, and should use the smallest stencil possible to be run on large meshes.

In this Appendix, the space discretisation is presented in Section A.2, then the scheme is derived from the flux continuity property, the resolution is performed through an iterative process. The obtained results are compared to the barycentric version of the SUSHI scheme Eymard et al. (2010) and provided in Section A.3.

The tensor product denoted by $\otimes$ makes a tensor of order $n+m$ from two tensors of order n and m. The dot product denoted by $\cdot$ makes a tensor of order $n-1+m-1$ from two tensors of order n et m ($n>0$, $m>0$). This product is identified to the canonic dot product on vectors for two first order tensors, to the canonic matrix/vector product for a second order tensor and a first order tensor.

A.2 Space discretisation of the anisotropic heterogeneous diffusion equation

In this chapter, 0^{th}-order tensors (identified to scalar fields) are not bold, whereas first-order tensors (identified to vector fields) and second-order tensors (identified to matrix fields) are bold.

The studied Poisson equation reads:

$$\begin{cases} -\text{div}\,(\boldsymbol{K}\cdot\mathbf{grad}\psi) = S_\psi & \text{on } \Omega, \\ \psi = \psi_d & \text{on } \mathcal{D}\subset\partial\Omega, \\ (\boldsymbol{K}\cdot\mathbf{grad}\psi)\cdot\boldsymbol{n} = Q_d & \text{on } \mathcal{N}=\partial\Omega\setminus\mathcal{D}. \end{cases} \tag{A.1}$$

where we denote by $\partial\Omega=\overline{\Omega}\setminus\Omega$ the boundary of the domain Ω, an open bounded connected polyhedral subset of $\mathbb{R}^d$ (d is the space dimension), ψ is the scalar field defined on the domain Ω and belongs to $H^1(\Omega)$, $\boldsymbol{K}$ is the tensor diffusivity field, assumed to be symmetric positive definite on the whole domain, continuous by part and limited, and $S_\psi\in L^2(\Omega)$ a source term. The boundary conditions on the field ψ are composed of Dirichlet conditions on $\mathcal{D}$, and of Neumann conditions on $\mathcal{N}$. $\mathcal{D}$ and $\mathcal{N}$ are partitions of the boundary $\partial\Omega$.

Space discretisation

The domain is discretised into cells V_c on which K is supposed to be constant. The barycentre of a cell c (respectively of a cell $\overline{c}$) is denoted by $\boldsymbol{x}_c$ (respectively by $\boldsymbol{x}_{\overline{c}}$). Two cells c and $\overline{c}$ are said to be neighbours if they share a face noted $f_{c|\overline{c}}$ of centre $\boldsymbol{x}_f$. The intersection between $f_{c|\overline{c}}$ and the vector $\boldsymbol{x}_{\overline{c}}-\boldsymbol{x}_c$ is $\boldsymbol{x}_o$. The unit normal vector to the face $f_{c|\overline{c}}$ oriented from c to $\overline{c}$ is $\boldsymbol{n}_{c>\overline{c}}$, the surface of $f_{c|\overline{c}}$ is $S_{f_{c|\overline{c}}}$, and we define $\boldsymbol{S}_{c>\overline{c}}=S_{f_{c|\overline{c}}}\boldsymbol{n}_{c>\overline{c}}$. The set of all interior faces of c is denoted $\mathcal{F}_c^{int}$. Finally, $\boldsymbol{x}'_c$ (respectively $\boldsymbol{x}'_{\overline{c}}$) is the orthogonal projection of $\boldsymbol{x}_c$ (respectively of $\boldsymbol{x}_{\overline{c}}$) with respect to the face $f_{c|\overline{c}}$.

Faces shared by one and only one cell c are said to be boundary faces and denoted f_b. $\boldsymbol{n}_{c>f_b}$ is the outward normal to the face f_b, and $\boldsymbol{S}_{c>f_b}$ is the normal vector to the face which norm is the surface. The set of boundary faces of c is denoted by $\mathcal{F}_c^{ext}$, whereas $\mathcal{F}_c$ is the set of all faces of c. Finally, $\boldsymbol{x}'_c$ is the orthogonal projection of $\boldsymbol{x}_c$ with respect to face normal of f_b. All the geometric definitions are recalled on Figure A.1(a) and on Figure A.1(b). The cell mean of ψ in c is defined by $\psi_c := \dfrac{1}{V_c}\displaystyle\int_{V_c}\psi\,\mathrm{d}\Omega$.

code_saturne uses a finite volume scheme where the solved variables are stored at cell centres: i.e. $\psi_{\boldsymbol{x}_c} = \psi_c$. The discretised field ψ is assumed to be affine on each cell. Therefore, for every point $\boldsymbol{x}_c''$ of a cell V_c, the following relationship reads:

$$\psi_{\boldsymbol{x}_c''} = \psi_c + \mathbf{grad}_c\psi \cdot \left(\boldsymbol{x}_c'' - \boldsymbol{x}_c\right), \tag{A.2}$$

where $\mathbf{grad}_c\psi$ is the ψ gradient, constant within each cell. Face quantities are defined by: $\psi_f = \frac{1}{S_f}\int_{S_f}\psi \mathrm{d}S$. Thanks to the affinity of ψ, we have $\psi_f = \psi_{\boldsymbol{x}_f}$. Integrating Equation (A.1) over a cell V_c gives:

$$-\sum_{f\in\mathcal{F}_c}(\boldsymbol{K}\cdot\mathbf{grad}\psi)_f\cdot\boldsymbol{S}_{c>f} = V_c S_{\psi_c}. \tag{A.3}$$

Two-point flux approximation scheme

The aim of this Section is to write a two-point flux as for isotropic diffusion problem on orthogonal meshes (see Eymard et al. (2014) for a description of TPFA schemes). One can notice that for the face $f_{c|\overline{c}}$, the diffusive flux seen by cell c can be written as:

$$\boxed{(\boldsymbol{K}_c\cdot\mathbf{grad}\psi)\cdot\boldsymbol{n}_{c>\overline{c}} = (\mathbf{grad}\psi)\cdot\left(\boldsymbol{K}_c^T\cdot\boldsymbol{n}_{c>\overline{c}}\right).} \tag{A.4}$$

The tensor $\boldsymbol{K}$ is symmetric, therefore $\boldsymbol{K}^T = \boldsymbol{K}$. The Equation (A.4) indicates that the direction $\boldsymbol{K}_c\cdot\boldsymbol{n}_{c>\overline{c}}$ is optimal to discretise the flux. Let $\boldsymbol{x}_c'$ (resp. $\boldsymbol{x}_{\overline{c}}''$) be a point on the line passing through $\boldsymbol{x}_f$ with direction vector $\boldsymbol{K}_c\cdot\boldsymbol{n}_{c>\overline{c}}$ (resp. $\boldsymbol{K}_{\overline{c}}\cdot\boldsymbol{n}_{c>\overline{c}}$). The diffusive flux seen from cell c is then approximated by:

$$(\boldsymbol{K}\cdot\mathbf{grad}\psi)_{\boldsymbol{x}_f}\cdot\boldsymbol{n}_{c>\overline{c}} = (\mathbf{grad}\psi)_{\boldsymbol{x}_f}\cdot(\boldsymbol{K}_c\cdot\boldsymbol{n}_{c>\overline{c}}) \simeq \frac{\|\boldsymbol{K}_c\cdot\boldsymbol{n}_{c>\overline{c}}\|}{\overline{\boldsymbol{x}_c''\boldsymbol{x}_f}}\left(\psi_{\boldsymbol{x}_f} - \psi_{\boldsymbol{x}_c''}\right), \tag{A.5}$$

where $\overline{\boldsymbol{x}_c''\boldsymbol{x}_f} := \left(\boldsymbol{x}_f - \boldsymbol{x}_c''\right)\cdot\frac{(\boldsymbol{K}_c\cdot\boldsymbol{n}_{c>\overline{c}})}{\|\boldsymbol{K}_c\cdot\boldsymbol{n}_{c>\overline{c}}\|}$ is the algebraic distance between $\boldsymbol{x}_c''$ and $\boldsymbol{x}_f$. The flux seen from cell $\overline{c}$ is approximated by:

$$(\boldsymbol{K}\cdot\mathbf{grad}\psi)_{\boldsymbol{x}_f}\cdot\boldsymbol{n}_{c>\overline{c}} = (\mathbf{grad}\psi)_{\boldsymbol{x}_f}\cdot(\boldsymbol{K}\cdot\boldsymbol{n}_{c>\overline{c}}) \simeq \frac{\|\boldsymbol{K}_{\overline{c}}\cdot\boldsymbol{n}_{c>\overline{c}}\|}{\overline{\boldsymbol{x}_f\boldsymbol{x}_{\overline{c}}''}}\left(\psi_{\boldsymbol{x}_{\overline{c}}''} - \psi_{\boldsymbol{x}_f}\right), \tag{A.6}$$

where $\overline{\boldsymbol{x}_f\boldsymbol{x}_{\overline{c}}''} := \left(\boldsymbol{x}_{\overline{c}}'' - \boldsymbol{x}_f\right)\cdot\frac{(\boldsymbol{K}_{\overline{c}}\cdot\boldsymbol{n}_{c>\overline{c}})}{\|\boldsymbol{K}_{\overline{c}}\cdot\boldsymbol{n}_{c>\overline{c}}\|}$ is the algebraic distance between $\boldsymbol{x}_f$ and $\boldsymbol{x}_{\overline{c}}''$. Enforcing the continuity of the fluxes (A.5) and (A.6) yields:

$$\psi_{\boldsymbol{x}_f} = \left(\frac{\overline{\boldsymbol{x}_c''\boldsymbol{x}_f}}{\|\boldsymbol{K}_c\cdot\boldsymbol{n}_{c>\overline{c}}\|} + \frac{\overline{\boldsymbol{x}_f\boldsymbol{x}_{\overline{c}}''}}{\|\boldsymbol{K}_{\overline{c}}\cdot\boldsymbol{n}_{c>\overline{c}}\|}\right)^{-1}\left(\psi_{\boldsymbol{x}_c''}\frac{\overline{\boldsymbol{x}_f\boldsymbol{x}_{\overline{c}}''}}{\|\boldsymbol{K}_{\overline{c}}\cdot\boldsymbol{n}_{c>\overline{c}}\|} + \psi_{\boldsymbol{x}_{\overline{c}}''}\frac{\overline{\boldsymbol{x}_c''\boldsymbol{x}_f}}{\|\boldsymbol{K}_c\cdot\boldsymbol{n}_{c>\overline{c}}\|}\right). \tag{A.7}$$

Therefore $\psi_{\boldsymbol{x}_f}$ is the weighted harmonic mean depending on $\boldsymbol{K}\cdot\boldsymbol{n}$ in cells c and $\overline{c}$. Injecting (A.7) in (A.5) gives a two-point formula for the flux through face $f_{c|\overline{c}}$:

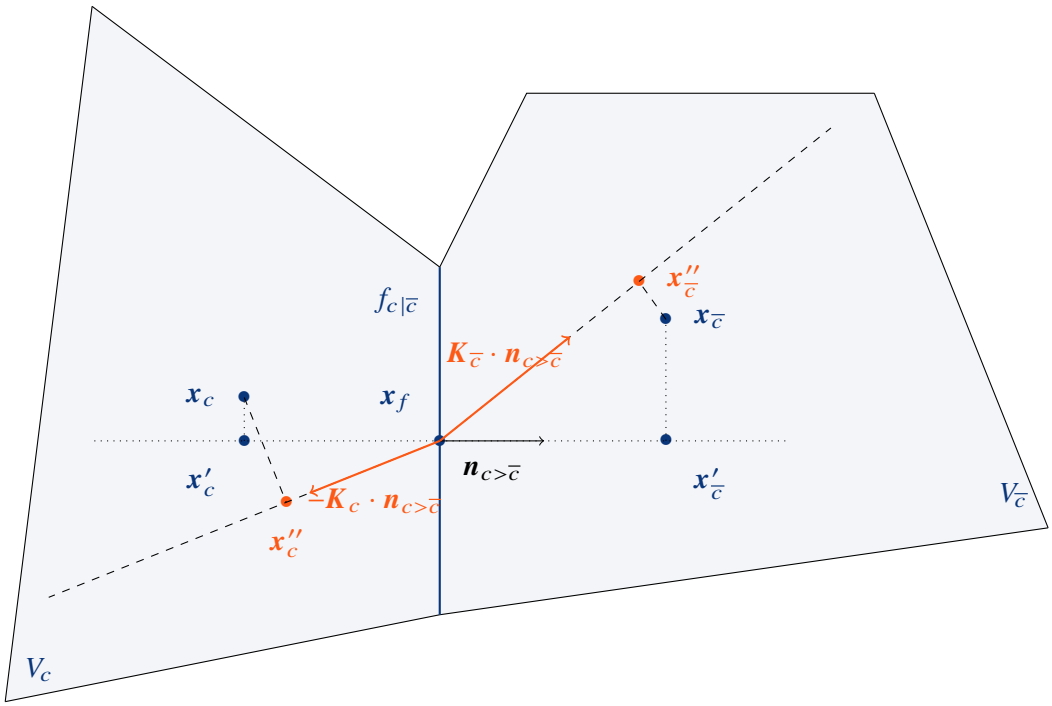
$$-(\boldsymbol{K}\cdot\mathbf{grad}\psi)_{f_{c|\overline{c}}}\cdot\boldsymbol{n}_{c>\overline{c}} = \frac{K_{f_{c|\overline{c}}}}{\overline{\boldsymbol{x}_c''\boldsymbol{x}_{\overline{c}}''}}\left(\psi_{\boldsymbol{x}_c''} - \psi_{\boldsymbol{x}_{\overline{c}}''}\right),$$

where the equivalent scalar face diffusivity $K_{f_{c|\overline{c}}}$ is defined by:

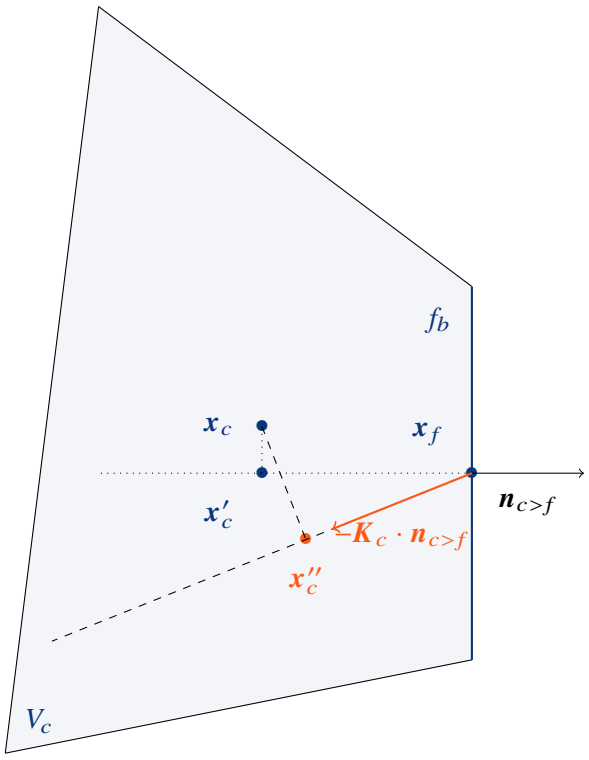
$$\frac{K_{f_{c|\overline{c}}}}{\overline{\boldsymbol{x}_c''\boldsymbol{x}_{\overline{c}}''}} = \left(\frac{\overline{\boldsymbol{x}_c''\boldsymbol{x}_f}}{\|\boldsymbol{K}_c\cdot\boldsymbol{n}_{c>\overline{c}}\|} + \frac{\overline{\boldsymbol{x}_f\boldsymbol{x}_{\overline{c}}''}}{\|\boldsymbol{K}_{\overline{c}}\cdot\boldsymbol{n}_{c>\overline{c}}\|}\right)^{-1}. \tag{A.8}$$

Besides, boundary terms are discretised as:

$$-(\boldsymbol{K}\cdot\mathbf{grad}\psi)_{f_b}\cdot\boldsymbol{n}_{c>f_b} \quad = \quad A_\psi^f + B_\psi^f\psi_{\boldsymbol{x}_c''}.$$



(a) Interior face.



(b) Boundary face.

Figure A.1: Sketch displaying interior and exterior faces with reconstruction points x''_c and $x''_{\overline{c}}$.

Non-orthogonalities reconstruction and iterative solving

As the field values of ψ at $\boldsymbol{x}_c''$ and $\boldsymbol{x}_{\overline{c}}''$ are not degrees of freedom of the discretised field, they are written in terms of ψ_c and $\psi_{\overline{c}}$ using the cell gradient through (A.2):

$$\begin{aligned} \psi_{\boldsymbol{x}_c''} &= \psi_{\boldsymbol{x}_c} + \mathbf{grad}_c\psi \cdot \left(\boldsymbol{x}_c'' - \boldsymbol{x}_c\right), \\ \psi_{\boldsymbol{x}_{\overline{c}}''} &= \psi_{\boldsymbol{x}_{\overline{c}}} + \mathbf{grad}_{\overline{c}}\psi \cdot \left(\boldsymbol{x}_{\overline{c}}'' - \boldsymbol{x}_{\overline{c}}\right). \end{aligned} \tag{A.9}$$

One should note that the choices for $\boldsymbol{x}_c''$ and $\boldsymbol{x}_{\overline{c}}''$ are still arbitrary, the only requirement is that $\boldsymbol{x}_c'' - \boldsymbol{x}_f$ should be collinear to $\boldsymbol{K}_c \cdot \boldsymbol{n}_{c>\overline{c}}$ and $\boldsymbol{x}_{\overline{c}}'' - \boldsymbol{x}_f$ should be collinear to $\boldsymbol{K}_{\overline{c}} \cdot \boldsymbol{n}_{c>\overline{c}}$. Eventually, inserting (A.9) in Equation (A.3) reads:

$$\begin{aligned} &\sum_{f_{c|\overline{c}} \in \mathcal{F}_c^{int}} \frac{K_{f_{c|\overline{c}}}}{\overline{\boldsymbol{x}_c''\boldsymbol{x}_{\overline{c}}''}} \left(\psi_c - \psi_{\overline{c}} + \left(\mathbf{grad}_c\psi \cdot \left(\boldsymbol{x}_c'' - \boldsymbol{x}_c\right) - \mathbf{grad}_{\overline{c}}\psi \cdot \left(\boldsymbol{x}_{\overline{c}}'' - \boldsymbol{x}_{\overline{c}}\right)\right)\right) \\ + &\sum_{f_b \in \mathcal{F}_c^{ext}} \left(A_\psi^f + B_\psi^f \left(\psi_c + \mathbf{grad}_c\psi \cdot \left(\boldsymbol{x}_c'' - \boldsymbol{x}_c\right)\right)\right) - V_c f_c \quad = \quad E_c(\psi) = 0, \end{aligned} \tag{A.10}$$

where $\mathbf{E}(.)$ is the affine operator to be solved and ψ be the vector containing the degree of freedom (vector of size N_{cel}) of the discretised field ψ: $\psi = \left[\psi_1, \cdots, \psi_i, \cdots, \psi_{N_{cel}}\right]$.

Equation (A.10) is solved using a Picard fixed-point. This approach is widely used in industrial CFD codes because it gives a linear system with good invertibility properties and with low memory consumption due to the small stencil. The problem (A.10) can be rewritten as: find ψ so that $\mathbf{E}(\psi) = \mathbf{0}$. Let $\left(\psi^k\right)_{k\in\mathbb{N}}$ be a series initialised at $\mathbf{0}$ defined by:

$$\begin{cases} \mathbf{EM}\left(\delta\psi^{k+1}\right) &= \mathbf{E}\left(\psi^k\right), \\ \psi^{k+1} &= \psi^k + \delta\psi^{k+1}, \end{cases} \tag{A.11}$$

where $\mathbf{EM}(\cdot)$ is the linear operator built with non-reconstructed fluxes:

$$EM_c\left(\delta\psi^{k+1}\right) = \sum_{f_{c|\overline{c}} \in \mathcal{F}_c^{int}} \frac{K_{f_{c|\overline{c}}}}{\overline{\boldsymbol{x}_c''\boldsymbol{x}_{\overline{c}}''}} \left(\delta\psi_c - \delta\psi_{\overline{c}}\right) + \sum_{f_b \in \mathcal{F}_c^{ext}} B_\psi^f \delta\psi_c. \tag{A.12}$$

$\mathbf{EM}$ can be considered as an approximation of the Hessian matrix of the affine operator[1] $\mathbf{E}(\cdot)$.

The incremental vector is computed at each sub-iteration solving a linear system of size $N_{cel} \times N_{cel}$. The sparse matrix $\mathbf{EM}$ is composed of a diagonal of size N_{cel} and an extra-diagonal of size N_{fac} (number of interior faces), The linear system (A.11) is rewritten into:

$$\underbrace{\begin{pmatrix} D_1 & & X_{ij} \\ & \ddots & \\ X_{ji} & & D_{N_{cel}} \end{pmatrix}}_{\text{Size } N_{cel}} \begin{pmatrix} \delta\psi_1^{k+1} \\ \vdots \\ \delta\psi_{N_{cel}}^{k+1} \end{pmatrix} = \begin{pmatrix} E\left(\psi^k\right)_1 \\ \vdots \\ E\left(\psi^k\right)_{N_{cel}} \end{pmatrix}. \tag{A.13}$$

[1] Operator $\mathbf{E}(\cdot)$ is affine and not linear because of the right-hand-side or when boundary conditions are non-zero. It will not be affine anymore if convection is taken into account, limiters or slope tests are non-linear.

R **EM** is a $M-$matrix: each diagonal component is strictly positive ($\forall c,\ D_c > 0$), each extra-diagonal component is negative and the opposite of the sum of the extra-diagonal terms of the same line is inferior to the diagonal term of the line ($\forall c,\ D_c \geq -\sum_{\overline{c}} X_{c\overline{c}}$).

Thus, **EM** is invertible and its inverse is positive definite. In all the numerical test cases, a conjugate gradient combined with an algebraic multi-grid algorithm is used to solve (A.11).

R A conjugate-gradient type algorithm can be developed to improve the convergence speed of the iterative process (A.11). Indeed, noting α such that the update of ψ^{k+1} is now:

$$\psi^{k+1} = \psi^k + \alpha\delta\psi^{k+1}, \tag{A.14}$$

an optimal value for α can be derived, minimising the L^2 norm $\left\|\mathbf{E}\left(\psi^{k+1}\right)\right\|$. Noting $\mathbf{A}\left(.\right)$ the linear part of $\mathbf{E}\left(.\right)$ and $\mathbf{B} = \mathbf{R}\left(\mathbf{0}\right)$, we obtain:

$$\alpha = -\frac{\left\langle \mathbf{A}\left(\delta\psi^{k+1}\right),\ \mathbf{E}\left(\psi^k\right)\right\rangle}{\left\|\mathbf{A}\left(\delta\psi^{k+1}\right)\right\|^2}. \tag{A.15}$$

Note that this only requires to store the vector of size N_{cel} $\mathbf{E}\left(\psi^k\right)$ and to compute (and store) the vector $\mathbf{A}\left(\delta\psi^{k+1}\right)$. This can be improved again storing the last two increments of the iterative process $\delta\psi^{k+1}$ and $\delta\psi^k$ and replacing (A.14) by:

$$\psi^{k+1} = \psi^k + \alpha\delta\psi^{k+1} + \beta\delta\psi^k. \tag{A.16}$$

Minimising again $\left\|\mathbf{E}\left(\psi^{k+1}\right)\right\|$ gives optimal values for α and β that are:

$$\begin{aligned}
\beta &= \frac{\left\langle \mathbf{A}\left(\delta\psi^{k+1}\right),\ \mathbf{A}\left(\delta\psi^k\right)\right\rangle \left\langle \mathbf{A}\left(\delta\psi^{k+1}\right),\ \mathbf{E}\left(\psi^k\right)\right\rangle - \left\|\mathbf{A}\left(\delta\psi^{k+1}\right)\right\|^2 \left\langle \mathbf{A}\left(\delta\psi^k\right),\ \mathbf{E}\left(\psi^k\right)\right\rangle}{\left\|\mathbf{A}\left(\delta\psi^{k+1}\right)\right\|^2 \left\|\mathbf{A}\left(\delta\psi^k\right)\right\|^2 - \left\langle \mathbf{A}\left(\delta\psi^{k+1}\right),\ \mathbf{A}\left(\delta\psi^k\right)\right\rangle^2}, \\
\alpha &= -\frac{\left\langle \mathbf{A}\left(\delta\psi^{k+1}\right),\ \mathbf{E}\left(\psi^k\right)\right\rangle + \beta\left\langle \mathbf{A}\left(\delta\psi^{k+1}\right),\ \mathbf{A}\left(\delta\psi^k\right)\right\rangle}{\left\|\mathbf{A}\left(\delta\psi^{k+1}\right)\right\|^2}.
\end{aligned} \tag{A.17}$$

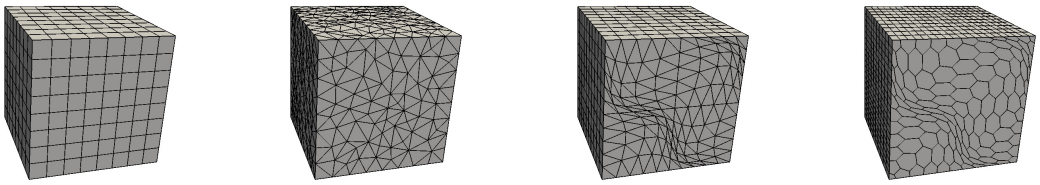
The denominator of β is zero if and only if $\mathbf{A}\left(\delta\psi^{k+1}\right)$ and $\mathbf{A}\left(\delta\psi^k\right)$ are collinear. In this case, β is set to 0. Note that (A.17) does not require additional computation (only additional storing of a vector of size N_{cel}) compared to (A.15). This dynamic relaxation makes the present algorithm converge on Kershaw type meshes.

If $\boldsymbol{x}_c''$ and $\boldsymbol{x}_{\overline{c}}''$ were chosen as orthogonal projections on the lines passing through $\boldsymbol{x}_f$ with respective direction vectors $\boldsymbol{K}_c \cdot \boldsymbol{n}_{c>\overline{c}}$ and $\boldsymbol{K}_{\overline{c}} \cdot \boldsymbol{n}_{c>\overline{c}}$, it would not ensure the positivity of the face viscosity $K_{f_{c|\overline{c}}}$ defined by (A.8) and the resulting matrix **EM** defined in (A.12) would loose its invertibility property. Hence, we choose to maintain $\boldsymbol{x}_c''$ and $\boldsymbol{x}_{\overline{c}}''$ on the same side of the face $f_{c|\overline{c}}$ as $\boldsymbol{x}_c$ and $\boldsymbol{x}_{\overline{c}}$, by setting: $\frac{\overline{\boldsymbol{x}_c''\boldsymbol{x}_f}}{\|\boldsymbol{K}_c\cdot\boldsymbol{n}_{c>\overline{c}}\|} = \max\left[\frac{(\boldsymbol{x}_f-\boldsymbol{x}_c)\cdot\boldsymbol{K}_c\cdot\boldsymbol{n}_{c>\overline{c}}}{\|\boldsymbol{K}_c\cdot\boldsymbol{n}_{c>\overline{c}}\|^2},\ \epsilon\frac{\overline{\boldsymbol{x}_c'\boldsymbol{x}_f}}{\|\boldsymbol{K}_c\cdot\boldsymbol{n}_{c>\overline{c}}\|}\right]$, with ϵ set to 0.1. That tends to drive away the built matrix from the one defined in (A.10), and thus requires more reconstruction sweeps, but keeps the matrix as a $M-$matrix. Finally, the reconstruction gradient used in Equation (A.10), is the standard cell-gradient displayed in the theory documentation of code_saturne Team (2021).

A.3 Verification test cases

In this Section, numerical results obtained on the FVCA6 benchmark test cases Eymard et al. (2011) are presented and compared to the barycentric version of SUSHI scheme Eymard et al.

(2010). The number of unknowns of the linear system is denoted by `nu`. We deliberately have not chosen exactly the mandatory meshes of the benchmark, but all the presented cases are run on hexahedral meshes (HEX), tetrahedral meshes (TET), prism meshes with triangle bases (BLS) and prism meshes with general bases (DBLS) of the benchmark mesh database (see Figure A.2).



(a) HEX mesh. (b) TET mesh. (c) BLS mesh. (d) DBLS mesh.

Figure A.2: An overview of mesh sequences.

Then the following orders of convergence are defined:

$$\texttt{ratiol2(i)} = -3\frac{\log\left(\frac{\texttt{erl2(i)}}{\texttt{erl2(i-1)}}\right)}{\log\left(\frac{\texttt{nu(i)}}{\texttt{nu(i-1)}}\right)}, \qquad \texttt{ratiograd(i)} = -3\frac{\log\left(\frac{\texttt{ergrad(i)}}{\texttt{ergrad(i-1)}}\right)}{\log\left(\frac{\texttt{nu(i)}}{\texttt{nu(i-1)}}\right)},$$

where `i` is the number of the mesh (from the coarser to the finer), `erl2(i)` is the normalized discrete L^2-error on the solution of the mesh number `i`, `ergrad(i)` the normalized discrete L^2-error on the gradient. In all cases below, the value of the analytical solution is imposed as a Dirichlet.

The first analytical test-case is defined by:

$$\begin{aligned} \boldsymbol{K} &= \begin{pmatrix} 1 & 0.5 & 0 \\ 0.5 & 1 & 0.5 \\ 0 & 0.5 & 1 \end{pmatrix}, \\ \psi_{ana} &= 1 + \sin(\pi x)\sin\left(\pi\left(y+\tfrac{1}{2}\right)\right)\sin\left(\pi\left(z+\tfrac{1}{3}\right)\right). \end{aligned} \tag{A.18}$$

The second analytical test case is defined as follows:

$$\begin{aligned} \boldsymbol{K} &= \begin{pmatrix} 1+y^2+z^2 & -xy & -xz \\ -xy & 1+x^2+z^2 & -yz \\ -xz & -yz & 1+x^2+y^2 \end{pmatrix}, \\ \psi_{ana} &= x^3y^2z + x\sin(2\pi xy)\sin(2\pi xz)\sin(2\pi z). \end{aligned} \tag{A.19}$$

This case is representative of the applications covered by the algorithm (GGDH on the diffusive term of a scalar transport equation) with an heterogeneous and anisotropic tensor $\boldsymbol{K}$, but not discontinuous.

The last analytical test-case presented here is:

$$\boldsymbol{K} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1000 \end{pmatrix}, \ \psi_{ana} = \sin(2\pi x)\sin(2\pi y)\sin(2\pi z)\,. \tag{A.20}$$

(a) Case 1.

Meshes	ratiol2	ratiograd
HEX	1.985	1.265
TET	2.136	1.029
BLS	1.888	1.016
DBLS	1.727	1.009

(b) Case 2.

Meshes	ratiol2	ratiograd
HEX	1.968	1.447
TET	2.090	0.927
BLS	1.962	0.922
DBLS	1.728	1.467

(c) Case 3.

Meshes	ratiol2	ratiograd
HEX	2.002	2.001
TET	2.321	1.691
BLS	1.702	1.012
DBLS	2.042	1.593

Table A.1: Orders of convergence for the two finest meshes of hexahedral (HEX), tetrahedral (TET), prism with triangle bases (BLS) and prism with general bases (DBLS) mesh sequences for cases (A.18), (A.19) and (A.20).

The Picard fixed-point algorithm is considered converged when the normed residual becomes smaller than 10^{-10}. Note that, by construction, no iteration is needed for cases with hexahedral meshes and orthotropic diffusion coefficients (e.g. case (A.20) and HEX meshes). On presented verification cases about 30 iterations are needed to obtain the required precision (up to 177 sweeps for (A.20) case with the finest TET mesh). A summary of convergence ratios is displayed in Table A.1 for the two finest meshes of each mesh sequence. We can notice that the order of convergence is around 2, even greater for the two finest tetrahedral meshes. This is due to the fact that the penultimate mesh is of poorer quality than the finest one as displayed on Figure A.3. One can remark that the numerical results have a precision of the order of the barycentric version of the SUSHI scheme, but are more sensitive to the mesh quality. The worst accuracy is obtained with the tetrahedral meshes. We also must admit that the results on Kershaw-type meshes are not shown, since the present scheme is sensitive to the mesh quality criterion, which are worst as the meshes are finer in the Kershaw-type sequence.

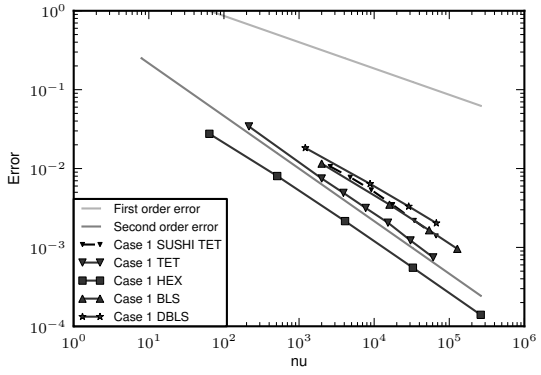
Conclusion

A new algorithm which solves anisotropic heterogeneous diffusion problem is presented. It gives satisfactory results on various types of meshes with an approximate second order of convergence in space in L^2-norm and with a precision close to the one obtained with the SUSHI scheme. Its main advantage is the small stencil needed, which allows an highly parallelisation of the algorithm within the industrial CFD code code_saturne.

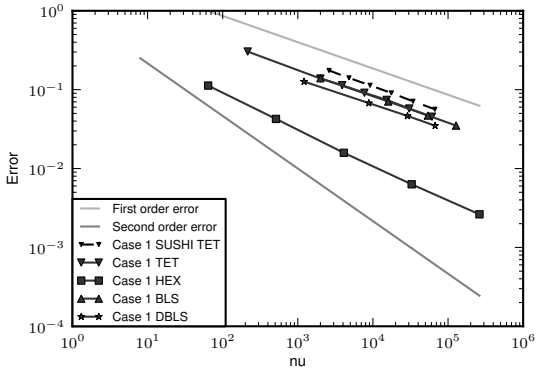
Bibliography

Dehoux, F., Benhamadouche, S., Manceau, R., 2017. An elliptic blending differential flux model for natural, mixed and forced convection. International Journal of Heat and Fluid Flow 63, 190–204.

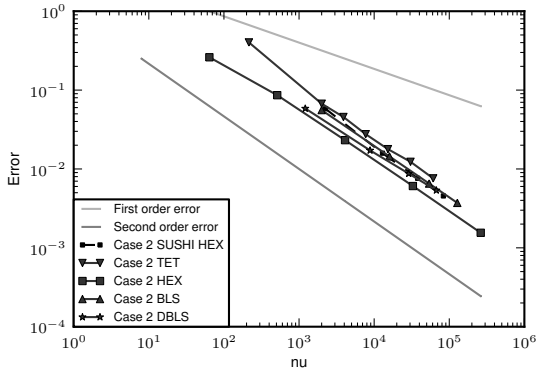
Eymard, R., Gallouët, T., Guichard, C., Herbin, R., Masson, R., 2014. TP or not TP, that is the question. Computational Geosciences 18, 285–296. doi:10.1007/s10596-013-9392-9.



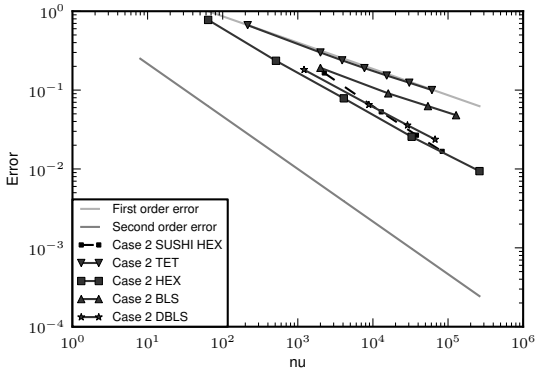
(a) L^2 error norm for Y for case 1.



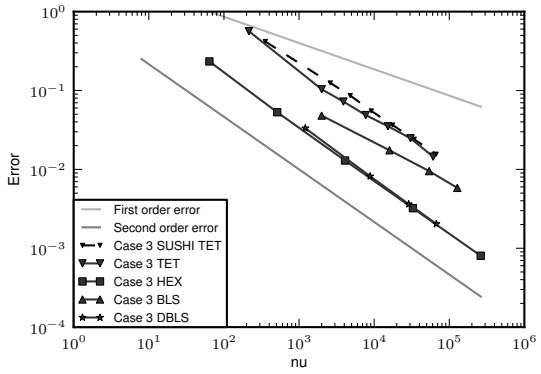
(b) L^2 error norm for **grad**Y for case 1.



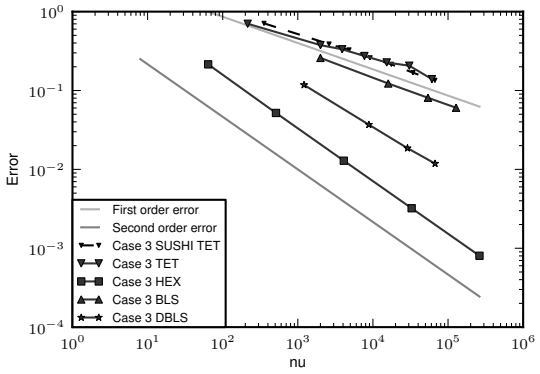
(c) L^2 error norm for Y for case 2.



(d) L^2 error norm for **grad**Y for case 2.



(e) L^2 error norm for Y for case 3.



(f) L^2 error norm for **grad**Y for case 3.

Figure A.3: L^2 error norm for Y and its gradient.

Eymard, R., Gallouët, T., Herbin, R., 2000. Finite volume methods. Handbook of Numerical Analysis 7, 713–1018.

Eymard, R., Gallouët, T., Herbin, R., 2010. Discretization of heterogeneous and anisotropic diffusion problems on general nonconforming meshes SUSHI: a scheme using stabilization and hybrid interfaces. IMA journal of numerical analysis 30, 1009–1043.

Eymard, R., Henry, G., Herbin, R., Hubert, F., Klöfkorn, R., Manzini, G., 2011. 3D Benchmark on Discretization Schemes for Anisotropic Diffusion Problems on General Grids, Springer. pp. 95–130.

Ferrand, M., Fontaine, J., Angelini, O., 2014. An anisotropic diffusion finite volume algorithm using a small stencil, in: Finite Volumes for Complex Applications VII-Elliptic, Parabolic and Hyperbolic Problems, pp. 577–585.

Svyatskiy, D., 2008. Benchmark on Anisotropic Problems.Nonlinear monotone finite volume method, in: Finite Volumes for Complex Applications V , pp. 935–947.

Team, D., 2021. code_saturne 7.0 Theory Guide. EDF R&D. URL: `https://code-saturne.org/cms/sites/default/files/docs/7.0/theory.pdf`.

B. A non-linear convective scheme to ensure minimum-maximum principle for transported variables by an incompressible flow in code_saturne

Abstract B.1 This appendix recalls the finite volume time and space discretisation used in code_saturne and introduce some classical non-linear convective schemes for scalar transport equation in an incrompressible flow. This is of particular interest for many physical modelling, such as fire modelling. This appendix presents a new continuous blending, originally proposed by Jérôme Laviéville, which ensure minimum-maximum principle for the transported scalar at the discrete level.

This work has been performed during the internship of David Iampietro in 2015 I supervised, in collaboration with Bertrand Sapa, EDF R&D, MFEE, and has been integrated in code_saturne by myself.

Résumé B.1 Cette annexe rappelle la discrétisation volumes finis en espace et en temps utilisée dans code_saturne et introduit quelques schémas classiques non-linéaire pour la discrétisation spatiale du terme convectif des équations de transport de scalaire pour un écoulement incompressible. Cela est particulièrement intéressant pour beaucoup de modélisation physiques, comme les incendies. Cette annexe présente aussi un nouveau schéma mélangeant continuement le schéma choisi à un schéma décentré amont afin d'assurer le principe du minimum-maximum au niveau discret pour le scalaire transporté. L'idée de ce schéma revient à Jérôme Laviéville.

Ce travail a été réalisé dans le cadre du stage de fin d'étude de David Iampietro que j'ai encadré, en collaboration avec Bertrand Sapa, d'EDF R&D, département MFEE, et ce travail a été intégré à code_saturne par mes soins.

B.1 Introduction

Convection is present in a wide class of fluid mechanics applications. Particularly in the case of fire modelling where the flow is highly non stationary, the medium is expansible and where many chemical species are transported. Convection is directly linked to hyperbolic conservation laws which have been studied theoretically and numerically for almost forty years. In some gas dynamics cases, physical solution has a sharp profile or is discontinuous across a propagation front. Even if conservative monotone schemes have good properties in terms of entropy, and so, on physical solution selection, they are at most first order accurate. Numerical diffusion corrupts

dreadfully the nature of the solution especially during long time simulation. Although a large panel of second-order finite difference schemes exists (e.g., P.D. Lax (1960), R. Beam (1976), or the method proposed in Fromm (1968)), they trigger spurious oscillations in areas where the computed solution gradients are sharp. As a consequence, new solution extrema appear in these areas. This can considerably alter the CFD simulation quality, for example in the case where convected chemical mass fractions are supposed to remain between zero and one overall the simulation.

Many authors have found ways to pass through this difficulty by introducing non linearities inside the numerical convective flux. Several methods have already been developed and tested numerically. J.P. Borisi (1973) and Zalesak (1979) have designed the Flux Corrected Transport (FCT) method which consist in adding successively diffusion in steep gradients areas and anti-diffusion in smooth areas. Harten and Osher (1987) and Harten et al. (1986) have laid the foundations of the Essentially Non Oscillatory (ENO) schemes. This method is based on polynomial interpolation coupled with linear combinations of stencils maintaining the spurious oscillations amplitude at the same level as the order of accuracy of the scheme. Eventually, Harten (1982) has introduced the notion of Decreasing Total Variation (DTV) schemes which prevent the non physical oscillations from appearing by imposing the discrete total variation of the computed solution to decrease throughout simulation. Sweby (1984) provides the shape of a limiter area in which a limiter function has to lie for the second order limited scheme to become DTV. Yee (1989) goes through the different limiter-inspired methods to make sure that a numerical scheme is DTV.

More recently in X. Zhang (2011), Zhang and Shu derive a high order finite volume scheme respecting the maximum principle without using the DTV criterion. Their method involves monotone fluxes[1] and feed them with face values obtained thanks to a modified Gauss-Lobatto quadrature ensuring accuracy and the maximum principle preservation.

Although these methods produce satisfying numerical results in one dimensional or multi-dimensional structured meshes, it is hard to implement them on three dimensional unstructured meshes. Indeed, even the notion of total variation does not hold any more in that case. Besides, methods based on interpolation to build the discrete high order face values become expensive computationally as the number of faces of the unstructured cell grow. In our work, as in X. Zhang (2011), spurious oscillations issues have been put aside in order to focus on the minimum and maximum principle. A new kind of limiter function has been designed to make sure that the limited numerical scheme respect the minimum-maximum principle along the simulation. This new limiter can directly be written in a 3-D unstructured formalism, is easy to compute and requires limited computational time.

In a first part the system of equations looked on is presented and the three dimensional unstructured finite volume scheme is described. Then, the new limiter formula is derived. Finally different simulation results are presented using the finite volume unstructured open-source solver code_saturne Team (2021). Firstly, the transport of a scalar at constant velocity and density is performed. Secondly a variation of this case allows density to vary in space and time.

B.2 Conservation laws

B.2.1 System of continuous equations

The system of continuous equations at stake describes mass, momentum and intensive scalar conservation:

[1] Which are known to provide numerical maximum principle if their arguments are cell values.

$$\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho\, \boldsymbol{u}) = \dot{m}, \tag{B.1a}$$

$$\frac{\partial}{\partial t}(\rho \boldsymbol{u}) + \mathbf{div}(\boldsymbol{u} \otimes \rho \boldsymbol{u}) = \mathbf{div}(\boldsymbol{\sigma}) + \boldsymbol{f_u}, \tag{B.1b}$$

$$\frac{\partial}{\partial t}(\rho \psi) + \operatorname{div}\ (\psi(\rho \boldsymbol{u})) = \operatorname{div}(\boldsymbol{J}_\psi) + S_\psi, \tag{B.1c}$$

$$\rho = \rho(\psi). \tag{B.1d}$$

In Equations (B.1), ρ is the fluid density, $\boldsymbol{u}$ the flow velocity, ψ a given conserved scalar. In our case it is supposed that E is constant. $\dot{m}$, $\boldsymbol{f_u}$ and S_ψ represent source terms. $\boldsymbol{\sigma}$ stands for the stress tensor, containing the effect of pressure p, and viscous stress $\boldsymbol{\tau}$. Assuming $\boldsymbol{\sigma}$ is a symmetric tensor, we have, for Newtonian fluids:

$$\begin{aligned} \boldsymbol{\sigma} &= \boldsymbol{\tau} - p\mathbf{1}, \\ \text{with } \boldsymbol{\tau} &= 2\,\mu\, \boldsymbol{S} + \left(\kappa - \frac{2}{3}\mu\right) \operatorname{tr}(\boldsymbol{S})\, \mathbf{1}, \\ \text{and } \boldsymbol{S} &= \frac{1}{2}(\boldsymbol{\nabla u} + \boldsymbol{\nabla u}^T). \end{aligned} \tag{B.2}$$

In Equation (B.2), μ represents the dynamic molecular viscosity and κ is the secondary viscosity, set to zero hereafter. $\mathbf{1}$ is the identity matrix.

Besides, $\boldsymbol{J}_\psi$ is the diffusive vector flux of scalar, which, using the Fick-Fourier law can usually be expressed as:

$$\boldsymbol{J}_\psi = K_\psi \mathbf{grad}\psi, \tag{B.3}$$

with K_ψ standing for the molecular diffusivity pertaining to ψ. For example, in the typical cases, ψ can be either pressure or temperature. In fire dynamics it can also model the mass fraction of a mix of two burning gas whose density are ρ_1 and ρ_2:

$$\rho(\psi) = 1/\left(\frac{\psi}{\rho_1} + \frac{(1-\psi)}{\rho_2}\right). \tag{B.4}$$

Most of the cases met in code_saturne are said to be incompressible. This means that the Mach number defined as $M := \frac{U_0}{c}$ where U_0 is a reference speed linked to the flow and c the speed of sound is widely smaller than 1. The effect of every pressure wave propagating at velocity close to c can be neglected because it will not be seen by the flow.

Subsequently, the mass production $\dot{m}$, the diffusive flux and the source terms related to the scalar ψ are no longer taken into account. Thus, Equations (B.1a) and (B.1c) become:

$$\begin{aligned} &\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho\, \boldsymbol{u}) = 0, \\ &\frac{\partial}{\partial t}(\rho \psi) + \operatorname{div}\ (\psi \rho \boldsymbol{u}) = 0. \end{aligned} \tag{B.5}$$

B.2.2 code_saturne: a multi-step finite volume code for incompressible flows

In this section, the whole algorithm to compute densities, velocity, mass fluxes and pressure in described assuming the flow is incompressible.

Time discretisation

The velocity pressure coupling in code_saturne is solved with a fractional step algorithm that can be related to Semi-Implicit Method for Pressure Linked Equations-Consistent (SIMPLEC) algorithm. Additional details can be found in Archambeau et al. (2004).

Let us consider time step n, extending from t^n to t^{n+1} and define ϕ^n the value of a variable $\phi(t)$ at time t^n. Moreover, define $\Delta t := t^{n+1} - t^n$ and momentum integrated over a time step is written $\boldsymbol{q}|_n^{n+1} := \rho \boldsymbol{u}|_n^{n+1}$. At time step n, p^n, $\boldsymbol{u}^n$, $\boldsymbol{q}|_n^{n+1}$ and ψ^n are already known from the preceding time step. In addition ρ^{n+1} has been found thanks to equation of state: $\rho^{n+1} = \rho(\psi^n)$. The algorithm can be split into three major steps:

Step 1: prediction step

Velocity is predicted using the explicit pressure gradient $\mathbf{grad} p^n$. Hence, the system finally reads:

$$\frac{\rho^n \boldsymbol{u}^* - \rho^{n-1}\boldsymbol{u}^n}{\Delta t} + \mathbf{div}\left[\boldsymbol{u}^* \otimes \boldsymbol{q}|_{n-1}^{n} - \mu\left(\boldsymbol{\nabla u}^* + (\boldsymbol{\nabla u}^*)^T - \frac{2}{3}\mathrm{div}\,(\boldsymbol{u}^*)\mathbf{1}\right)\right] = -\mathbf{grad} p^n + \boldsymbol{f}_{\boldsymbol{u}}^n. \tag{B.6a}$$

p and ψ remain unchanged in this step.

Step 2: correction step

In this step the predicted velocity is corrected by introducing a pressure variation gradient $\mathbf{grad}(p^{n+1} - p^n)$. The new momentum $\boldsymbol{q}|_n^{n+1}$ is designed to enforce mass conservation equation (B.1a). The system is therefore:

$$\frac{\rho^{n+1}\boldsymbol{u}^{n+1} - \rho^n \boldsymbol{u}^*}{\Delta t} = -\mathbf{grad}(p^{n+1} - p^n), \tag{B.7a}$$

$$\frac{\rho^{n+1} - \rho^n}{\Delta t} + \mathrm{div}\left(\boldsymbol{q}|_n^{n+1}\right) = 0, \tag{B.7b}$$

$$\boldsymbol{q}|_n^{n+1} = \rho^{n+1}\boldsymbol{u}^{n+1}. \tag{B.7c}$$

In practice, to derive a system for pressure variation, one takes the divergence of Equation (B.7a) and uses Equation (B.7b) to eliminate $\boldsymbol{q}|_n^{n+1}$, which yields the following Poisson equation:

$$\mathrm{div}\left[\Delta t\,\mathbf{grad}(p^{n+1} - p^n)\right] = \frac{\rho^{n+1} - \rho^n}{\Delta t} + \mathrm{div}\,(\rho^n \boldsymbol{u}^*). \tag{B.8}$$

Step 3: scalar update step

In this step scalar advection Equation (B.1c) is solved. Contrary to the velocity derivation in this appendix, ψ is discretised using a θ-scheme to reach second order in time. Indeed, ψ related to the convective flux (inside the divergence operator) is approached by $\psi^{n+\theta} := \Theta\left(\psi^n, \psi^{n+1}\right) = (1-\theta)\,\psi^n + \theta\,\psi^{n+1}$. Hence one gets:

$$\frac{\rho^{n+1}\psi^{n+1} - \rho^n\psi^n}{\Delta t} + \mathrm{div}\,\left[\psi^{n+\theta}\,\boldsymbol{q}|_n^{n+1}\right] = 0. \tag{B.9a}$$

Space discretisation

Space discretisation is derived only for the scalar ψ which is the variable of first interest in our simulations. By applying a spatial average $\frac{1}{V_c}\int_{V_c} .\,\mathrm{d}\Omega$ over an unstructured polygonal cell c and defining the scalar average as:

$$\widetilde{\psi}_c^n := \frac{\overline{\rho\psi}_c^n}{\overline{\rho}_c^n} := \frac{\dfrac{1}{V_c}\displaystyle\int_{V_c} \rho\psi\,\mathrm{d}\Omega}{\dfrac{1}{V_c}\displaystyle\int_{V_c} \rho\,\mathrm{d}\Omega}. \tag{B.10}$$

Notations in this section have been chosen to draw a link to the Reynolds and Favre averages in turbulence modelling.

Defining $\mathcal{F}_c$ the set of faces delimiting the border of the cell c, one obtains:

$$\overline{\rho}_c{}^{n+1} = \overline{\rho}_c{}^{n} - \frac{\Delta t}{V_c} \sum_{f \in \mathcal{F}_c} \dot{M}_{c>f}\big|_n^{n+1}, \tag{B.11a}$$

$$\overline{\rho}_c{}^{n} \frac{\widetilde{\psi}_c^{n+1} - \widetilde{\psi}_c^{n}}{\Delta t} - \frac{\widetilde{\psi}_c^{n+1}}{V_c} \sum_{f \in \mathcal{F}_c} \dot{M}_{c>f}\big|_n^{n+1} + \frac{1}{V_c} \sum_{f \in \mathcal{F}_c} \psi_f^{n+\theta}\ \dot{M}_{c>f}\big|_n^{n+1} = 0, \tag{B.11b}$$

with:

$$\dot{M}_{c>f}\big|_n^{n+1} := \left(\int_f \boldsymbol{q}|_n^{n+1}\, \mathrm{d}S \right) \cdot \boldsymbol{n}_{c>f}, \tag{B.11c}$$

$$\psi_f^{n+\theta} := \frac{\left(\int_f \psi^{n+\theta}\ \boldsymbol{q}|_n^{n+1}\, \mathrm{d}S \right) \cdot \boldsymbol{n}_{c>f}}{\dot{M}_{c>f}\big|_n^{n+1}} \tag{B.11d}$$

$$\simeq \Theta\left(\underbrace{\frac{\left(\int_f \psi^{n}\ \boldsymbol{q}|_n^{n+1}\, \mathrm{d}S \right) \cdot \boldsymbol{n}_{c>f}}{\dot{M}_{c>f}\big|_n^{n+1}}}_{\psi_f^n}, \underbrace{\frac{\left(\int_f \psi^{n+1}\ \boldsymbol{q}|_n^{n+1}\, \mathrm{d}S \right) \cdot \boldsymbol{n}_{c>f}}{\dot{M}_{c>f}\big|_n^{n+1}}}_{\psi_f^{n+1}} \right). \tag{B.11e}$$

Replacing $V_c \frac{\overline{\rho}_c{}^{n+1} - \overline{\rho}_c{}^{n}}{\Delta t} \widetilde{\psi}_c^{n+1}$ by $-\widetilde{\psi}_c^{n+1} \sum_{f \in \mathcal{F}_c} \dot{M}_{c>f}\big|_n^{n+1}$ using the exact mass balance, a numerical scheme for scalar ψ is obtained:

$$\begin{aligned} \frac{V_c \overline{\rho}_c{}^{n}}{\Delta t} \left(\psi_c^{n+1} - \psi_c^{n} \right) - \sum_{f \in \mathcal{F}_c} \psi_c^{n+1}\ \dot{M}_{c>f}\big|_n^{n+1} + \theta \sum_{f \in \mathcal{F}_c} \psi_f^{n+1}\ \dot{M}_{c>f}\big|_n^{n+1} \\ + (1-\theta) \sum_{f \in \mathcal{F}_c} \psi_f^{n}\ \dot{M}_{c>f}\big|_n^{n+1} = 0, \end{aligned} \tag{B.12}$$

where $\widetilde{.}$ are dropped for sake of simplicity from now on.

B.2.3 Face approximations

Consider Figure B.1 representing two unstructured cells sharing one common face f. ψ_c^n and $\psi_{\overline{c}}^n$ are located at the barycentres of the cells. Point $\boldsymbol{x}_f$ is the middle of the shared face whereas $\boldsymbol{x}_o$ is the middle of the segment made of the barycentres. $\boldsymbol{S}_{c>\overline{c}}$ is the *outward* face normal vector multiplied by the face surface. The face normal vector is actually oriented from cell c toward cell $\overline{c}$. Point $\boldsymbol{x}'_c$ and $\boldsymbol{x}'_{\overline{c}}$ are the orthogonal projection of the barycentres on the line normal to the face, passing through $\boldsymbol{x}_f$. In addition let us define $a_{c>\overline{c}} := \dfrac{\left| \boldsymbol{x}'_{\overline{c}} - \boldsymbol{x}_f \right|}{\left| \boldsymbol{x}'_{\overline{c}} - \boldsymbol{x}'_c \right|}$.

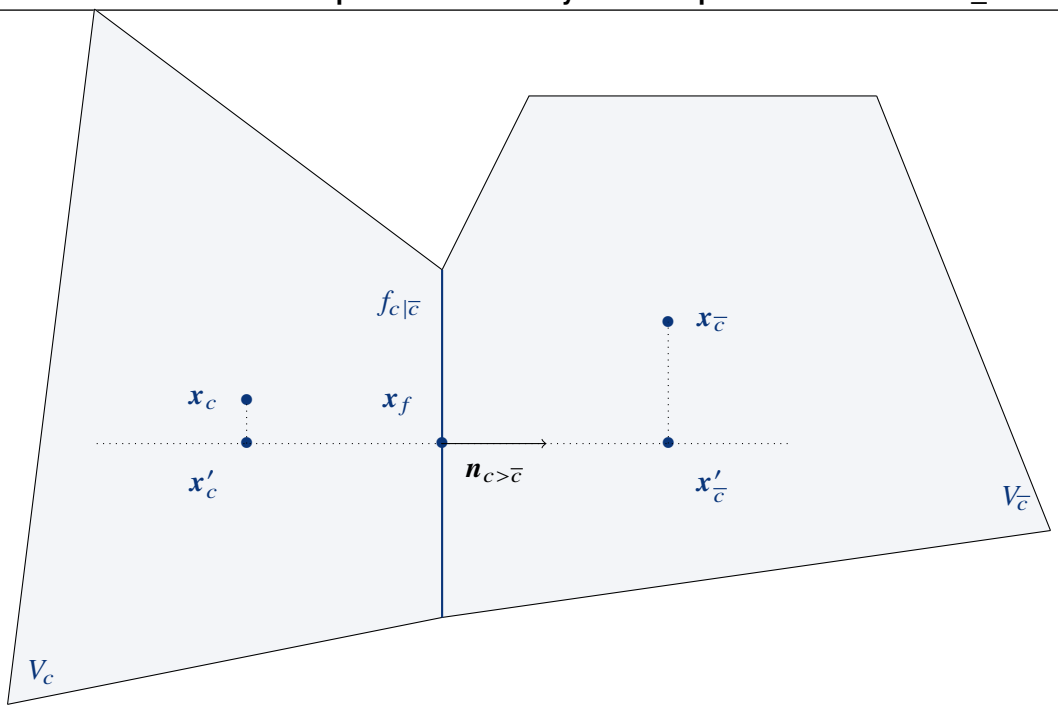


Figure B.1: Two neighbouring unstructured cells.

In code_saturne, there are many options to approximate the face values ψ_f^n using the neighboring cell values. Table B.1 summarizes all the formulae:

Name	Formula	Notation
First-order upwind	$\psi_f^{U,n} := \begin{cases} \psi_c^n \text{ if } \dot{M}_{c>f}\big\|_n^{n+1} \geq 0, \\ \psi_{\overline{c}}^n \text{ otherwise.} \end{cases}$	1U
Second-order linear upwind (SOLU)	$\psi_f^{2U\,upw,n} := \begin{cases} \psi_c^n + \mathbf{Grad}_c^U \{\psi^n\} \cdot (\boldsymbol{x}_f - \boldsymbol{x}_c) \text{ if } \dot{M}_{c>f}\big\|_n \geq 0, \\ \psi_{\overline{c}}^n + \mathbf{Grad}_{\overline{c}}^U \{\psi^n\} \cdot (\boldsymbol{x}_{\overline{c}} - \boldsymbol{x}_f) \text{ otherwise.} \end{cases}$	2U upw
SOLU variant of code_saturne	$\psi_f^{2U,n} := \begin{cases} \psi_c^n + \mathbf{Grad}_c \{\psi^n\} \cdot (\boldsymbol{x}_f - \boldsymbol{x}_c) \text{ if } \dot{M}_{c>f}\big\|_n \geq 0, \\ \psi_{\overline{c}}^n + \mathbf{Grad}_{\overline{c}} \{\psi^n\} \cdot (\boldsymbol{x}_{\overline{c}} - \boldsymbol{x}_f) \text{ otherwise.} \end{cases}$	2U classic
Center	$\psi_f^{C,n} := a_{c>\overline{c}} \psi_c^n + (1 - a_{c>\overline{c}}) \psi_{\overline{c}}^n$	C

Table B.1: Different Space Numerical Schemes: ψ_f^n.

Here, $\mathbf{Grad}_c^U\{.\}$ stands for a 3D unstructured upwind gradient:

$$\begin{aligned}\mathbf{Grad}_c^U\{\psi^n\} &:= \frac{1}{V_c}\left[\psi_c^n \sum_{f\in\mathcal{F}_c/\ \dot{M}_{c>f}|_n^{n+1}>0} \boldsymbol{S}_{c>f} + \sum_{f\in\mathcal{F}_c/\ \dot{M}_{c>f}|_n^{n+1}\leq 0} \psi_{\overline{c}}^n \boldsymbol{S}_{c>f}\right] \\ &= \frac{1}{V_c}\left[\sum_{f\in\mathcal{F}_c/\ \dot{M}_{c>f}|_n^{n+1}\leq 0} \left(\psi_{\overline{c}}^n - \psi_c^n\right)\boldsymbol{S}_{c>f}\right].\end{aligned} \tag{B.13}$$

In case of a 1-D structured mesh, expression (B.13) reduces to the classic expression which can be found in Wang and Hutter (2001).

B.2.4 Roe–Sweby limiter

A fourth way to approximate the face value can be done using Roe–Sweby limiter theory:

Centered limited by Roe–Sweby	$\psi_f^{RS/C,n} = \psi_f^{U,n} + \phi\left(r_{c>f}^n\right)\left(\psi_f^{C,n} - \psi_f^{U,n}\right)$	RS/C

Table B.2: Non Linear Limited Face Value Approximation: ψ_f^n.

In this expression, ϕ is a limiter function designed to remain in Sweby's area derived in Sweby (1984). Besides, $r_{c>f}^n := \frac{\psi_C^n - \psi_U^n}{\psi_D^n - \psi_C^n}$ is the rate between the upwind and the face slopes of the computed scalar.

Let us first define ψ_C^n respectively ψ_D^n, the "current" respectively the "downstream" values of the scalar ψ as:

$$\begin{cases} \text{If:} & \dot{M}_{c>f}|_n^{n+1} \geq 0, \\ & \psi_C^n = \psi_c^n,\ \psi_D^n = \psi_{\overline{c}}^n. \\ \text{If:} & \dot{M}_{c>f}|_n^{n+1} < 0, \\ & \psi_C^n = \psi_{\overline{c}}^n,\ \psi_D^n = \psi_c^n. \end{cases} \tag{B.14}$$

Then, the upstream value of the scalar ψ at face f is:

$$\psi_U^n := \psi_C^n - \frac{V_C}{|\boldsymbol{S}_{c>f}|}\mathbf{Grad}_C^U\{\psi^n\}\cdot\boldsymbol{n}_{c>f}. \tag{B.15}$$

The Roe–Sweby limiter applied to the centered scheme has been designed such that, in a 1-D structured case with constant density and constant mass flux the scheme (B.12) be DTV.

B.2.5 Iterative algorithm

Because the numerical scheme (B.12) can become non linear in $\psi = \left\{\psi_1^n, .., \psi_c^n, .., \psi_{N_{cells}}^n\right\}$ if, for example, the Roe–Sweby limiter function is used, an iterative process based on a fixed point algorithm is triggered to compute the scalar values at time t_{n+1}.

The principle of such iterative process is the following: first, identify the non linear discrete operator $\mathbf{E}_n(.)$ associated with the numerical scheme in equation (B.12). Then apply an iterative

fixed point algorithm: $\mathbf{E}_n(\psi^k + \delta\psi^{k+1}) \approx \mathbf{E}_n(\psi^k) + \boldsymbol{\nabla}\,\mathbf{E}_n(\psi^k)\,\delta\psi^{k+1} = 0$. By looking at the Roe–Sweby limited face value in Table B.2, a linear part $\mathbf{E}_n^L(.)$ related to the first order upwind scheme ψ_f^U and a non linear part $\mathbf{E}_n^{NL}(.)$ related to the limiter function can be identified. Then neglecting the non linear operator, one can suppose that: $\boldsymbol{\nabla}\,\mathbf{E}_n(\psi^k) = \boldsymbol{\nabla}\,\left(\mathbf{E}_n^L + \mathbf{E}_n^{NL}\right)(\psi^k) = \mathbf{E}_n^L + \cancel{\boldsymbol{\nabla}\,\mathbf{E}_n^{NL}(\psi^k)} \approx \mathbf{E}_n^L$. The iterative system solved is then:

$$\begin{aligned} &\forall c, \forall k \geq 1 : \\ &\psi_c^0 = \psi_c^n, \\ &\psi_c^{k+1} = \psi_c^k + \delta\psi_c^{k+1}, \\ &\left(\mathbf{E}_n^L\,\delta\psi^{k+1}\right)_c = -\left(\mathbf{E}_n(\psi^k)\right)_c, \end{aligned} \tag{B.16}$$

with:

$$\left(\mathbf{E}_n(\psi)\right)_c := \frac{V_c {\rho_c}^n}{\Delta t}\left(\psi_c - \psi_c^n\right) + \sum_{f\in\mathcal{F}_c}\left(\theta\psi_f - \psi_c\right)\dot{M}_{c>f}\big|_n^{n+1} + (1-\theta)\sum_{f\in\mathcal{F}_c}\psi_f^n\,\dot{M}_{c>f}\big|_n^{n+1}, \tag{B.17a}$$

$$\left(\mathbf{E}_n^L(\psi)\right)_c := \frac{V_c {\rho_c}^n}{\Delta t}\left(\psi_c - \psi_c^n\right) + \sum_{f\in\mathcal{F}_c}\left(\theta\psi_f^U - \psi_c\right)\dot{M}_{c>f}\big|_n^{n+1} + (1-\theta)\sum_{f\in\mathcal{F}_c}\psi_f^n\,\dot{M}_{c>f}\big|_n^{n+1}. \tag{B.17b}$$

One can notice that, in the implicit case where $\theta = 1$, the matrix $\mathbf{E}_n^L$ is a diagonally dominant matrix. In such a case the algorithm converges if a Jacobi linear solver is used.

B.2.6 New blending guaranteeing the discrete minimum-maximum principle of ψ

The idea behind the notion of β-limiter is to keep the same formalism as the one introduced in Table B.2: $\psi_f^{\beta,n} := \psi_f^{U,n} + \beta_f(\psi_f^{\star,n} - \psi_f^{U,n})$. Here, $\psi_f^{\star,n}$ is a second order face approximation[2]. $\beta_f \in [0,1]$ is a weight depending on the neighbouring cell values and designed such that the numerical flux (B.12) preserves the minimum-maximum principle. This result has been derived under the hypothesis that the values imposed by the inlet boundary conditions remained between the bounds of the theoretical solution. Since it is known that first order scheme $\psi_f^{U,n}$ naturally preserves the minimum-maximum principle, one will require that, in non monotone or discontinuous areas[3], β_f be close to zero. On the contrary, in smooth regions β_f will be close to one allowing the second order scheme to fully play its role and bring sufficient accuracy.

Theorem B.2.1 Consider ψ_{inf} the inferior bound of a theoretical solution of equation (B.1c). For all cells c define:

$$\beta_c^{\text{inf}} := \frac{\dfrac{V_c {\rho_c}^n}{\Delta t}\left(\psi_c^n - \psi_{\text{inf}}\right) - (1-\theta)\displaystyle\sum_{f\in\mathcal{F}_c}\left(\psi_f^{U,n} - \psi_{\text{inf}}\right)\dot{M}_{c>f}\big|_n^{n+1}}{\displaystyle\sum_{f\in\mathcal{F}_c^{int}}\left[\theta\left(\psi_f^{k,\star} - \psi_f^{k,U}\right) + (1-\theta)\left(\psi_f^{n,\star} - \psi_f^{n,U}\right)\dot{M}_{c>f}\big|_n^{n+1}\right]^+}, \tag{B.18a}$$

with $[a]^+ := \frac{a+|a|}{2}$.

[2] 2U or C for example.

[3] Where $\psi_f^{\star,n}$ could produce non-physical oscillations.

Assume that the time step respects the CFL condition:

$$\Delta t \leq \min_{c} \frac{\rho_c V_c \left(\psi_c^n - \psi_{\text{inf}}\right)}{(1-\theta)\left[\sum_{f\in\mathcal{F}_c}\left(\psi_f^{U,n} - \psi_{\text{inf}}\right)\dot{M}_{c>f}\big|_n^{n+1}\right]^+}. \tag{B.19}$$

If $\beta_{f_{c|\overline{c}}}$ is defined such that $\beta_{f_{c|\overline{c}}} := \max\left(\min\left(\beta_c^{\text{inf}}, \beta_{\overline{c}}^{\text{inf}}, 1\right), 0\right)$, then the numerical scheme (B.12) preserves the minimum principle meaning that:

$$\forall\, c \in [1, N_{cells}],\ \forall\, n \geq 0,\ \psi_c^n \geq \psi_{\text{inf}}. \tag{B.20}$$

R

1. Note that the CFL condition (B.19) if automatically fulfilled if $\theta = 1$, i.e. for implicit Euler scheme.
2. If the linear system is solved using a Jacobi solver, we can prove that the minimum-maximum principle is preserve at each iteration of the linear solver, as the matrix obtained is an M-matrix.

Proof. The iterative system (B.16) for a given cell c reads:

$$\begin{aligned}\left(\frac{V_c{\rho_c}^{n+1}}{\Delta t} + \theta \sum_{f\in\mathcal{F}_c/\ \dot{M}_{c>f}\big|_n^{n+1}>0} \dot{M}_{c>f}\big|_n^{n+1}\right)\psi_c^{k+1} + \theta \sum_{f\in\mathcal{F}_c/\ \dot{M}_{c>f}\big|_n^{n+1}<0} \psi_{\overline{c}}^{k+1}\,\dot{M}_{c>f}\big|_n^{n+1} = \\ \frac{V_c{\rho_c}^{n}}{\Delta t}\psi_c^n + \theta\sum_{f\in\mathcal{F}_c}\left(\psi_f^{k,U} - \psi_f^k\right)\dot{M}_{c>f}\big|_n^{n+1} \\ -(1-\theta)\sum_{f\in\mathcal{F}_c}\psi_f^n\,\dot{M}_{c>f}\big|_n^{n+1}, \\ \text{with: } \frac{V_c{\rho_c}^{n+1}}{\Delta t} = \frac{V_c{\rho_c}^{n}}{\Delta t} - \sum_{f\in\mathcal{F}_c}\dot{M}_{c>f}\big|_n^{n+1} \text{ thanks to (B.11a).}\end{aligned} \tag{B.21}$$

Equation (B.21) is invariant by a shifting of $-\psi_{\text{inf}}$, thanks to the mass equation (B.11a), so it is sufficient to find β_f so that $\psi_c - \psi_{\text{inf}}$ remains positive, where $\psi_c - \psi_{\text{inf}}$ also verifies (B.21).

The left hand side in Equation (B.21) provides the expression of the diagonal and extra-diagonal coefficients of the matrix $\mathbf{E}_n^L$, which is a M-matrix. Then, it just needs to ensure that the right-hand-side is positive:

$$\frac{V_c{\rho_c}^{n}}{\Delta t}\psi_c^n + \theta\sum_{f\in\mathcal{F}_c}\left(\psi_f^{k,U} - \psi_f^k\right)\dot{M}_{c>f}\big|_n^{n+1} - (1-\theta)\sum_{f\in\mathcal{F}_c}\psi_f^n\,\dot{M}_{c>f}\big|_n^{n+1} \geq 0, \tag{B.22}$$

to enforce $\psi_c^{k+1} \geq 0$. Subsequently one can rewrite (B.22) as:

$$\sum_{f\in\mathcal{F}_c^{int}}\beta_f\left[\theta\left(\psi_f^{k,\star} - \psi_f^{k,U}\right) + (1-\theta)\left(\psi_f^{n,\star} - \psi_f^{n,U}\right)\right]\dot{M}_{c>f}\big|_n^{n+1} \leq \frac{V_c{\rho_c}^{n}}{\Delta t}\psi_c^n - (1-\theta)\sum_{f\in\mathcal{F}_c}\psi_f^{U,n}\,\dot{M}_{c>f}\big|_n^{n+1}.$$

(B.23)

It has to be noticed that the sum over the pure upwind face value is done on all sets of face $\mathcal{F}_c$ whereas the face values linked to β_f are only internal face values ($f_{c|\overline{c}} \in \mathcal{F}_c^{int}$). This is simply a consequence of the fact that in code_saturne, border face values follow an upwind scheme ($\beta_f = 0$ for these faces).

Now let β_c^{inf} be a real numbers such that $\forall f_{c|\overline{c}} \in \mathcal{F}_c^{int}$:

$$0 \leq \beta_{f_{c|\overline{c}}} \leq \beta_c^{\text{inf}} \leq 1, \tag{B.24}$$

and then the following inequality stands:

$$\begin{aligned} &\sum_{f \in \mathcal{F}_c^{int}} \beta_{f_{c|\overline{c}}} \left[\theta \left(\psi_f^{k,\star} - \psi_f^{k,U} \right) + (1-\theta) \left(\psi_f^{n,\star} - \psi_f^{n,U} \right) \right] \dot{M}_{c>f} \Big|_n^{n+1} \\ \leq \quad & \beta_c^{\text{inf}} \sum_{f \in \mathcal{F}_c^{int}} \left[\theta \left(\psi_f^{k,\star} - \psi_f^{k,U} \right) + (1-\theta) \left(\psi_f^{n,\star} - \psi_f^{n,U} \right) \dot{M}_{c>f} \Big|_n^{n+1} \right]^+ ; \end{aligned} \tag{B.25}$$

It is therefore sufficient to impose:

$$\beta_c^{\text{inf}} \leq \frac{\dfrac{V_c {\rho_c}^n}{\Delta t} \left(\psi_c^n - \psi_{\text{inf}} \right) - (1-\theta) \displaystyle\sum_{f \in \mathcal{F}_c} \left(\psi_f^{U,n} - \psi_{\text{inf}} \right) \dot{M}_{c>f} \Big|_n^{n+1}}{\displaystyle\sum_{f \in \mathcal{F}_c^{int}} \left[\theta \left(\psi_f^{k,\star} - \psi_f^{k,U} \right) + (1-\theta) \left(\psi_f^{n,\star} - \psi_f^{n,U} \right) \dot{M}_{c>f} \Big|_n^{n+1} \right]^+}, \tag{B.26}$$

to ensure inequality (B.22), which implies the minimum principle.

Now, for a given face $f_{c|\overline{c}}$ between the neighbouring cells c and $\overline{c}$ define:

$$\beta_{f_{c|\overline{c}}} := \max \left(0,\ \min \left(\beta_c^{\text{inf}}, \beta_{\overline{c}}^{\text{inf}}, 1 \right) \right) \in [0,\ 1]. \tag{B.27}$$

Then the hypothesis (B.24) holds whatever cell is studied. The minimum principle is fulfilled. ■

Corollary B.2.2 Consider ψ_{sup} the superior bound of a theoretical solution of equation (B.1c). For all cell c define:

$$\beta_c^{\text{sup}} := \frac{\dfrac{V_c {\rho_c}^n}{\Delta t} \left(\psi_{\text{sup}} - \psi_c^n \right) - (1-\theta) \displaystyle\sum_{f \in \mathcal{F}_c} \left(\psi_{\text{sup}} - \psi_f^{U,n} \right) \dot{M}_{c>f} \Big|_n^{n+1}}{- \displaystyle\sum_{f \in \mathcal{F}_c^{int}} \left[\theta \left(\psi_f^{k,\star} - \psi_f^{k,U} \right) + (1-\theta) \left(\psi_f^{n,\star} - \psi_f^{n,U} \right) \dot{M}_{c>f} \Big|_n^{n+1} \right]^-}, \tag{B.28a}$$

with $[a]^- := \frac{a-|a|}{2}$.

Assume that the time step respects the CFL condition:

$$\Delta t \leq \min_c \frac{\rho_c V_c \left(\psi_{\text{sup}} - \psi_c^n \right)}{(1-\theta) \left[\displaystyle\sum_{f \in \mathcal{F}_c} \left(\psi_{\text{sup}} - \psi_f^{U,n} \right) \dot{M}_{c>f} \Big|_n^{n+1} \right]^+}. \tag{B.29}$$

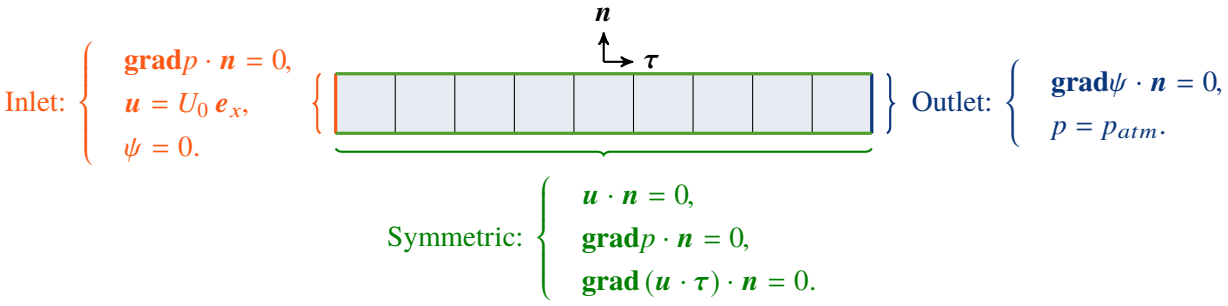


Figure B.2: 1-D Case: passive transport of a scalar.

If β_f is defined such that $\beta_f := \max\left(\min\left(\beta_c^{\text{sup}}, \beta_{\bar{c}}^{\text{sup}}, 1\right), 0\right)$, then the numerical scheme (B.12) preserves the maximum principle:

$$\forall\, c \in [1, N_{cells}],\ \forall\, n \geq 0,\ \psi_c^n \leq \psi_{\text{sup}}. \tag{B.30}$$

Proof. The proof is identical to the above one except that a new β_c^{sup} is defined such that $\forall f \in \mathcal{F}_c^{int}$:

$$0 \leq \beta_f \leq \beta_c^{\text{sup}} \leq 1, \tag{B.31}$$

and inequality (B.25) transforms itself into:

$$\begin{aligned} &\sum_{f \in \mathcal{F}_c^{int}} -\beta_f \left[\theta\left(\psi_f^{k,\star} - \psi_f^{k,U}\right) + (1-\theta)\left(\psi_f^{n,\star} - \psi_f^{n,U}\right)\right] \dot{M}_{c>f}\big|_n^{n+1} \\ \leq\ & -\beta_c^{\text{sup}} \sum_{f \in \mathcal{F}_c^{int}} \left[\theta\left(\psi_f^{k,\star} - \psi_f^{k,U}\right) + (1-\theta)\left(\psi_f^{n,\star} - \psi_f^{n,U}\right) \dot{M}_{c>f}\big|_n^{n+1}\right]^-. \end{aligned} \tag{B.32}$$

The remaining part of the proof is similar. ■

B.3 Numerical results

One wants to compare the classical first and second order numerical schemes described on Table B.1, the Roe–Sweby limiter and the new β-limiter. Different verification cases with analytical solutions have been implemented. The first one is one dimensional on a structured mesh.

B.3.1 Passive scalar transported at constant velocity

The first case has been directly inspired from Leonard (1990). Here, density is constant in space and time: $\rho = \rho_0 = 1.178\,\text{kg}\,\text{m}^{-3}$. The domain is one dimensional as presented on Figure B.2.

Here, the theoretical velocity solution of Navier–Stokes equations is $\boldsymbol{u}(x,\, t) = U_0 \boldsymbol{e}_x$. Thus the theoretical scalar conservation law becomes: $\forall\ (x,\, t),\ \psi(x,t) = \psi_0(x - U_0\, t)$, where ψ_0 is the solution at $t = 0$.

Two different kinds of initial inputs ψ_0 have been tested: a smooth sinus-squared, and a discontinuous step.

One can notice that the initial step is discontinuous at $x = 4\,\text{m}$ and $x = 6\,\text{m}$. Besides a non monotone area is present at the center of the squared-sinus and the ellipse. The length of the domain

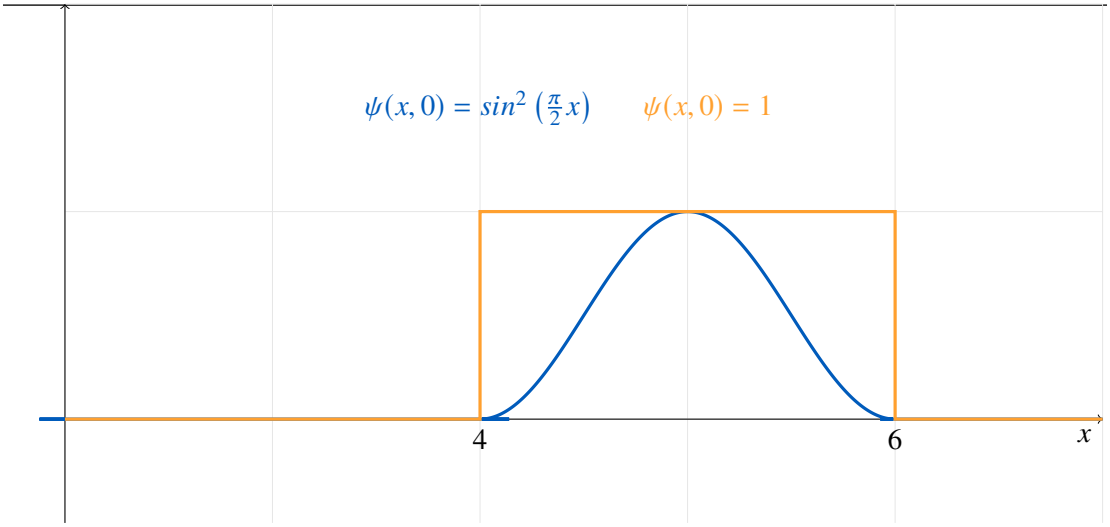
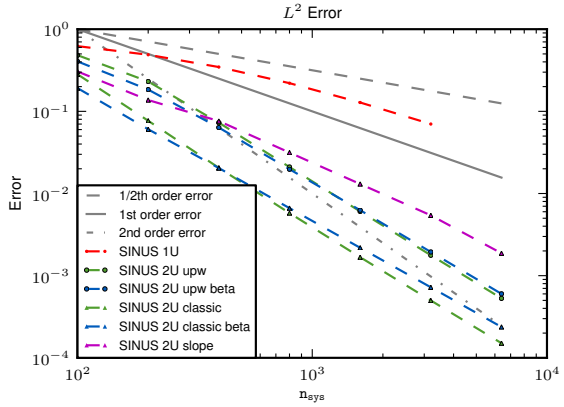


Figure B.3: Two initial conditions for the 1-D convection case.

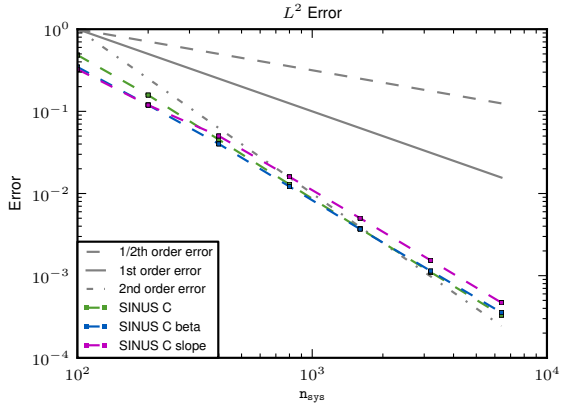
is $L = 20\,\text{m}$. In order to study the convergence of the numerical scheme order of accuracy different cells numbers have been involved : $N_{cells} \in [100, 200, 400, 800, 1600, 3200, 6400, 12800]$, $\Delta x = L/N_{Cells}$. The physical simulation duration is $T_{simu} = 45\,\text{s}$. The inlet velocity norm is $U_0 = 0.1\,\text{m}\,\text{s}^{-1}$ and the time step is defined such that the Courant number $\nu := \frac{U_0 \Delta t}{\Delta x} = 0.05$, $\forall\, \Delta x$. Besides, for both initial inputs the theoretical total variation $TV(\psi)(t) := \lim_{\epsilon \to 0} \frac{1}{\epsilon} \int_{\mathbb{R}} |\psi(x+\epsilon, t) - \psi(x,t)| dx = 2$, $\forall\, t$.

Case: Crank–Nicolson squared-sinus

Another simulation involving the squared-sinus is conducted with $\theta = 1/2$. In that case, the scheme is also not supposed to become second order accurate in time[4] because the initial condition is C^1 and not C^2. Hence, all the second order space accurate schemes should not be perturbed by the time step. A line whose slope coefficient is equal to two should be observed.



(a) Convergence study for the 2U schemes for the sinus intial profile with $\theta = 1/2$.



(b) Convergence study for the C scheme for the sinus intial profile with $\theta = 1/2$.

Figure B.4(a) shows results a of the proposed β-limiter applied to variants of the second-order linear upwind (SOLU) schemes and compared to the classical upwind scheme (denoted by "1U"),

[4] $r_t = 2$ in $e^{T_{simu}}_{\Delta t,\, \Delta x} \sim C(T_{simu})((\Delta x)^{r_x} + (\Delta t)^{r_t})$.

Scheme ($k = 3200$)	$CO(k)$
2U upw	1.77086782941
2U upw beta	1.68866658476
2U classic	1.75094816569
2U classic beta	1.59764641163
2U classic slope	1.26647185574
C	1.7618233958
C beta	1.69476449676
C slope	1.70750706693

Figure B.4: Convergence of space order for the sinus initial conditions and $\theta = 1/2$.

the original SOLU (denoted by "2U upw", see Fromm (1968)) without limiter, the code_saturne variant of the SOLU scheme (denoted by "2U classic", see Team (2021) for the details) without limiter and with the original slope test of code_saturne (see again Team (2021) for the details). Figure B.4(b) shows results a of the proposed β-limiter applied to the centered and compared to the centered scheme (denoted by "C") without limiter and with the original slope test of code_saturne (see again Team (2021) for the details). Graphically, with the exception of 2U classic slope, all space second order schemes seem to have a slope coefficient close to 1.7 which is rather satisfying although additional simulations should be done. Table B.4 writes $CO(3200)$ for the different schemes.

Besides, even in that case the Roe–Sweby and the β-limited schemes work perfectly in terms of min-max preservation. Note also that, by construction, the variable ψ is conserved locally at the discrete level.

B.4 Conclusion

This appendix has presented the space and time discretisation of a scalar with a particular attention on convective scheme. A new blending implemented on any 3-D cell shape and tested with 1-D analytical solutions, called the β-limiter is introduced to ensure minimum-maximum principle at the discrete level for the transported scalars.

Bibliography

Archambeau, F., Méchitoua, N., Sakiz, M., 2004. Code_saturne: a finite volume method for the computation of turbulent incompressible flows - industrial applications. International Journal on Finite Volumes 1, 1–62.

Fromm, J., 1968. A method for reducing dispersion in convective difference schemes. Applied Mathematics Modelling 19, 640–653.

Harten, A., 1982. High resolution schemes for hyperbolic conservation laws. Journal of Computational Physics .

Harten, A., Osher, S., 1987. Uniformly high order accurate nonoscillatory schemes i. SIAM Journal of Numerical Analysis 24, 279–309.

Harten, A., Osher, S., Engquist, B., Chakravarthy, S., 1986. Some results on uniformly high order accurate essentially non-oscillatory schemes. Applied Numerical Mathematics 2, 347–377.

J.P. Borisi, D.B., 1973. Flux corrected transport i: Shasta, a fluid transport algorithm that works. Journal of computational physics 11, 38–69.

Leonard, B.P., 1990. The ultimate conservative difference scheme appliedto unsteady one-dimensional advection. Computer Methods in Applied Mechanics and Engineering .

P.D. Lax, B.W., 1960. Systems of conservation laws. Communications in Pure and Applied Mathematics 13, 217–237.

R. Beam, R.W., 1976. An implicit finite-difference algorithm for hyperbolic systems in conservation-law form. Journal of Computational Physics 22.

Sweby, P.K., 1984. High resolution schemes using flux limiters for hyperbolic conservation laws. SIAM Journal of Numerical Analysis 21, 995–1011.

Team, D., 2021. code_saturne 7.0 Theory Guide. EDF R&D. URL: `https://code-saturne.org/cms/sites/default/files/docs/7.0/theory.pdf`.

Wang, Y., Hutter, K., 2001. Comparisons of numerical methods with respect to convectively dominated problems. International Journal for Numerical Methods in Fluids .

X. Zhang, C.S., 2011. Maximum-principle-satisfying and positivity-preserving high order schemes for conservation laws: Survey and new developments. Proceedings of the Royal Society A 471.

Yee, H.C., 1989. A Class of High-Resolution Explicit and Implicit Shock-Capturing Methods. NASA.

Zalesak, S., 1979. Fully multidimensional flux corrected transport algorithms for fluids. Journal of computational physics 31, 335–362.

C. Extension of the ALE finite volumes scheme to second-order in time

Abstract C.1 This appendix is an attempt to extend to second-order in time the ALE algorithm presented in Chapter 4.

Résumé C.1 Cette annexe propose une écriture à l'ordre deux en temps pour l'algorithme *ALE* présenté au Chapitre 4.

As in Chapter 4, we start with volume, mass an momentum balance equations written on mobile volumes:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V_c(t)} \mathrm{d}\Omega = \int_{\partial V_c(t)} \boldsymbol{v}\cdot\mathbf{d}\boldsymbol{S}, \tag{4.15}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V_c(t)} \rho\mathrm{d}\Omega + \int_{\partial V_c(t)} \rho\left(\boldsymbol{u}-\boldsymbol{v}\right)\cdot\mathbf{d}\boldsymbol{S} = 0, \tag{4.16}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V_c(t)} \rho\boldsymbol{u}\mathrm{d}\Omega + \int_{\partial V_c(t)} \boldsymbol{u}\otimes\rho\left(\boldsymbol{u}-\boldsymbol{v}\right)\cdot\mathbf{d}\boldsymbol{S} = \int_{\partial V_c(t)} \boldsymbol{\sigma}\cdot\mathbf{d}\boldsymbol{S}. \tag{4.18}$$

We supplemented that by a (buoyant) scalar balance equation transported by the fluid velocity (e.g. energy or salinity):

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V_c(t)} \rho\psi\mathrm{d}V_c + \int_{\partial V_c(t)} \psi\rho\left(\boldsymbol{u}-\boldsymbol{v}\right)\cdot\mathbf{d}\boldsymbol{S} = \int_{\partial V_c(t)} \boldsymbol{\Phi}_\psi\cdot\mathbf{d}\boldsymbol{S}, \tag{C.1}$$

with $\boldsymbol{\Phi}_\psi$ being the molecular flux of ψ.

We propose in the next Section an extension to the time-segregated scheme presented in Chapter 1, §1.2 and similar to the one proposed by Wall et al. (2002).

We recall that the time stepping is defined by the parameter θ, bounded in $[0, 1]$ and we take $\theta = 1/2$ to get the Crank-Nicolson scheme, expected to be 2^{nd} order in time.

For any field ψ, we recall that the time interpolation between times n and $n+1$ writes:

$$\Theta\left(\psi^n, \psi^{n+1}\right) = (1-\theta)\,\psi^n + \theta\psi^{n+1}. \tag{1.32}$$

For sake of simplicity, we take here constant time steps Δt.

We recall that extensive quantities integrated over cells are denoted by:

$$\begin{aligned} V_c &= \int_{V_c} \mathrm{d}\Omega, \\ M_c &= \int_{V_c} \rho \mathrm{d}\Omega, \\ \boldsymbol{Q}_c &= \int_{V_c} \rho \boldsymbol{u} \mathrm{d}\Omega. \end{aligned} \tag{1.17}$$

We also introduced two extensive face quantities, face volume flux over a time interval $[n,\ n+1]$:

$$\dot{V}_{c>f}\big|_n^{n+1} \coloneqq \int_f \boldsymbol{v} \cdot \mathrm{d}\boldsymbol{S}\Big|_n^{n+1}, \tag{C.2}$$

and face (total[1]) mass flux over a time interval $[n,\ n+1]$:

$$\dot{M}_{c>f}\big|_n^{n+1} \coloneqq \int_f \rho\,(\boldsymbol{u}-\boldsymbol{v}) \cdot \mathrm{d}\boldsymbol{S}\Big|_n^{n+1}. \tag{C.3}$$

With these notations, cell volume and mass balances rewrite:

$$\begin{aligned} \frac{[V_c]_n^{n+1}}{\Delta t} &= \sum_{f\in\mathcal{F}_c} \dot{V}_{c>f}\big|_n^{n+1}, \\ \frac{[M_c]_n^{n+1}}{\Delta t} &= \sum_{f\in\mathcal{F}_c} \dot{M}_{c>f}\big|_n^{n+1}, \end{aligned} \tag{C.4}$$

where $[X]_n^{n+1} \coloneqq X^{n+1} - X^n$.

Initialisation: The first sub-iteration is still denoted by $k = 1$ and therefore some values have to be initialised for $k = 0$. For the density at time n, $\rho_c^{n,0} = \rho_c^{n-1}$ and for all other k, $\rho_c^{n,k} = \rho_c^n$. For the density at time $n+1$, $\rho_c^{n+1,0} = \rho_c^n$. Moreover, $V_c^{n+1,0} = V_c^n$. For the mass flux, $\dot{M}_{c>f}\big|_n^{n+1,0} = \dot{M}_{c>f}\big|_{n-1}^{n}$. For the pressure force, $\boldsymbol{F}^p_{c>f}\Big|_{n-1+\theta}^{n+\theta,0} \coloneqq p\boldsymbol{S}_{c>f}\big|_{n-1+\theta}^{n+\theta,0} = p\boldsymbol{S}_{c>f}\big|_{n-2+\theta}^{n-1+\theta}$.

Buoyancy, solve $\psi^{n+1,k}$ and update $\rho^{n+1,k}$: During this stage, the values of the buoyant scalars ψ and of the density ρ are updated.

Firstly, the transport equation of buoyant scalars is solved:

$$\frac{M_c^{n+1,k-1}\psi_c^{n+1,k} - M_c^{n,k-1}\psi_c^n}{\Delta t\big|_n^{n+1,k-1}} + \sum_{f\in\mathcal{F}_c} \left\langle \Theta\left(\psi^n,\ \psi^{n+1,k}\right)\right\rangle_f \dot{M}_{c>f}\big|_n^{n+1} = RHS_c^{n+\theta,k}, \tag{C.5}$$

where the right-hand side term contains the source terms and the diffusion and is integrated over the (moving) cell. Note that for face interpolation $\left\langle \Theta\left(\psi^n,\ \psi^{n+1,k}\right)\right\rangle_f$ to be second order, the face centre should be the one at $n+\theta$: $\boldsymbol{x}_f^{n+\theta} \coloneqq \Theta\left(\boldsymbol{x}_f^n,\ \boldsymbol{x}_f^{n+1}\right)$.

For the first sub-iteration in k, namely $k = 1$, the reader is reminded that $M_c^{n,k-1}$ is M_c^{n-1}.

Secondly, the density is updated with the equation of state: $\rho^{n+1,k} = f\left(\psi^{n+1,k}\right)$. Note that this new density is not balanced by any mass flux yet. This mass balance will be performed in the upcoming correction step, which ensures mass conservation.

[1]Total means here that the mass flux is due to fluid velocity going through the surface and that the surface is moving.

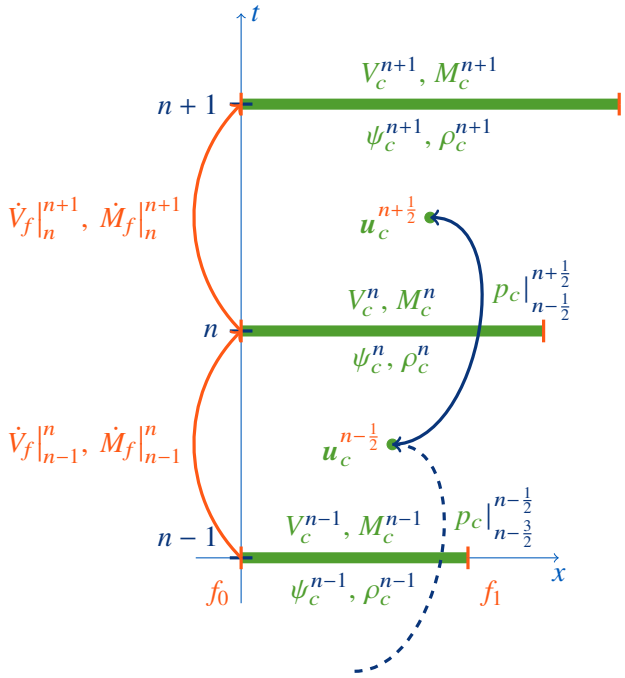


Figure C.1: Second-order in time time stepping for the finite volumes approach in ALE framework, variables localisation.

Momentum, prediction of the velocity, solve $\widetilde{\boldsymbol{u}}^k$: During this stage, the values of the predicted velocity $\widetilde{\boldsymbol{u}}$ are estimated. With the velocity-pressure coupling in mind, the prediction step is the preliminary one. The outcome of the velocity-pressure coupling is the **total** mass flux $\dot{M}_{c>f}|_n^{n+1,k}$, which is naturally centred on the time interval $[n, n+1]$. Thus, the momentum equation is solved on the time interval $[n-1+\theta, n+\theta]$:

$$\frac{\Theta\left(M_c^n,\, M_c^{n+1,k-1}\right)\widetilde{\boldsymbol{u}}_c^k - \Theta\left(M_c^{n-1},\, M_c^{n,k-1}\right)\boldsymbol{u}_c^{n-1+\theta}}{\Delta t|_{n-1+\theta}^{n+\theta}} + \sum_{f\in\mathcal{F}_c}\left\langle\Theta\left(\boldsymbol{u}^{n-1+\theta},\,\widetilde{\boldsymbol{u}}^k\right)\right\rangle_f \dot{M}_{c>f}|_{n-1+\theta}^{n+\theta,k-1} + \sum_{f\in\mathcal{F}_c}\left\langle \boldsymbol{F}^p|_{n-1+\theta}^{n+\theta,k-1}\right\rangle_{c>f} = RHS^{n+\theta,k}, \tag{C.6}$$

where the right-hand side term is a cell integral and contains some source terms and the diffusion. Note that here, for pressure force and velocities face interpolations to be second order, the face centre should be the one centred between $n-1+\theta$ and $n+\theta$, for $\theta = \frac{1}{2}$, $\boldsymbol{x}_f^n$.

The mass flux $\dot{M}|_{n-1+\theta}^{n+\theta,k-1}$ is defined by $\dot{M}|_{n-1+\theta}^{n+\theta,k-1} = \Theta\left(\dot{M}|_{n-1}^n,\, \dot{M}|_n^{n+1,k-1}\right)$ and therefore verifies the mass balance with $\frac{M_c^{n+\theta,k-1}-M_c^{n-1+\theta}}{\Delta t|_{n-1+\theta}^{n+\theta}} = \theta\frac{M_c^{n+1,k-1}-M_c^n}{\Delta t|_{n-1}^n} + (1-\theta)\frac{M_c^n-M_c^{n-1}}{\Delta t|_n^{n+1}}$.

Once again, unsteady term and convection of (C.6) can be rewritten as:

$$\Theta\left(M_c^{n-1},\, M_c^{n,k-1}\right)\frac{\widetilde{\boldsymbol{u}}_c^k - \boldsymbol{u}_c^{n-1+\theta}}{\Delta t|_{n-1+\theta}^{n+\theta}} + \sum_{f\in\mathcal{F}_c}\left\langle\Theta\left(\boldsymbol{u}^{n-1+\theta},\,\widetilde{\boldsymbol{u}}^k\right)\right\rangle_f \dot{M}_{c>f}|_{n-1+\theta}^{n+\theta,k-1} - \left(\sum_{f\in\mathcal{F}_c}\dot{M}_{c>f}|_{n-1+\theta}^{n+\theta,k-1}\right)\widetilde{\boldsymbol{u}}_c^k. \tag{C.7}$$

Mass, correction of the velocity, solve Φ^k: During this stage, the values of the pressure increment Φ, of the resulting pressure p, of the corrected velocity $\boldsymbol{u}$ and of the mass flux are updated.

With the velocity-pressure coupling in mind, the correction step is the concluding one. The resulting mass flux will balance the density obtained in the buoyancy step.

$$\begin{cases} \Theta\left(\rho^{n,k-1},\, \rho^{n+1,k-1}\right) \dfrac{\boldsymbol{u}^{n+\theta,k} - \widetilde{\boldsymbol{u}}^k}{\Delta t|_{n-1+\theta}^{n+\theta}} + \mathbf{grad}\Phi^k = \mathbf{0}, \\ \dfrac{\rho_c^{n+1,k} V_c^{n+1,k-1} - \rho_c^n V_c^n}{\Delta t|_n^{n+1}} + \displaystyle\sum_{f\in\mathcal{F}_c} \dot{\tilde{M}}_{c>f}\Big|_n^{n+1,k} = 0, \end{cases} \tag{C.8}$$

with, a first estimation (mesh and thus volumes will be updated from $k-1$ to k afterwards) of the total mass flux given by:

$$\begin{aligned} \dot{\tilde{M}}_{c>f}\Big|_n^{n+1,k} &= \left\langle \Theta\left(\rho^{n,k-1},\, \rho^{n+1,k-1}\right) \left(\boldsymbol{u}^{n+\theta} - \boldsymbol{v}|_n^{n+1,k-1}\right) \cdot \boldsymbol{S} \right\rangle_{c>f} \\ &= \left\langle \Theta\left(\rho^{n,k-1},\, \rho^{n+1,k-1}\right) \left(\widetilde{\boldsymbol{u}}^k - \boldsymbol{v}|_n^{n+1,k-1}\right) \cdot \boldsymbol{S} \right\rangle_{c>f} + \Delta t|_{n-1+\theta}^{n+\theta}\, \mathbf{grad}_f \Phi^k \cdot \boldsymbol{S}_{c>f}, \end{aligned}$$

where $\boldsymbol{v}|_n^{n+1,0}$ is initialised by $\mathbf{0}$, and in a consistent way $V_c^{n+1,k-1}$. Note that space interpolation should be done at $\boldsymbol{x}_f^{n+\theta}$. We recall that no space indices are written for the first equation of (C.8), because this equation is used both for cells (to update cell velocity) and faces (to update mass fluxes).

At the end of the correction step, the pressure over the time interval $[n-1+\theta, n+\theta]$ is updated with:

$$p|_{n-1+\theta}^{n+\theta,k} = p|_{n-1+\theta}^{n+\theta,k-1} + \Phi^k. \tag{C.9}$$

At the end of this step, one gets the new velocity which is living in time $n+\theta$.

Volume, update mesh and mass flux: We now consider that mesh nodes move, and therefore cell volumes evolve with velocity $\boldsymbol{v}|_n^{n+1,k} = \delta\boldsymbol{v}^k + \boldsymbol{v}|_n^{n+1,k-1}$, this mesh velocity increment should verify:

$$\frac{V_c^{n+1,k} - V_c^{n+1,k-1}}{\Delta t} = \sum_{f\in\mathcal{F}_c} \dot{V}_{c>f}\Big|_n^{n+1,k} - \dot{V}_{c>f}\Big|_n^{n+1,k-1}, \tag{C.10}$$

with $\dot{V}_{c>f}\Big|_n^{n+1,k} - \dot{V}_{c>f}\Big|_n^{n+1,k-1} = \delta\boldsymbol{v}^k \cdot \boldsymbol{S}_{c>f} = \frac{\boldsymbol{x}_f^{n+1,k} - \boldsymbol{x}_f^{n+1,k-1}}{\Delta t} \cdot \boldsymbol{S}_{c>f}$. Finally mass flux can be updated:

$$\dot{M}_{c>f}\Big|_n^{n+1,k} = \dot{\tilde{M}}_{c>f}\Big|_n^{n+1,k} + \left\langle \Theta\left(\rho^{n,k-1},\, \rho^{n+1,k-1}\right) \cdot \boldsymbol{S} \right\rangle_{c>f} \delta\boldsymbol{v}^k.$$

Bibliography

Wall, C., Pierce, C.D., Moin, P., 2002. A semi-implicit method for resolution of acoustic waves in low Mach number flows. Journal of Computational Physics 181, 545–563.

D. SPH Appendices

Abstract D.1 First of all, this appendix presents a summary of the discretised equations solved in the SPH solvers such as SPARTACUS-2D or Sphynx. Then the derivation of repulsive forces from Lagrangian, as performed by Kulasegaram et al. (2004), is presented. This derivation makes possible to derive skew-adjoint gradient and divergence operators, which is a key property for conservation matter. Unfortunately, all this derivation is performed for an insulated system of particles, which means that part of the boundary terms are missing.

An algorithm for the analytical computation in 3-D of the gradient of the renormalisation factor (which is somehow an inward boundary normal vector) is proposed, as published in Mayrhofer et al. (2015). This algorithm decompose boundary elements (triangular faces) intersection with the kernel support sphere into basic integrals that can be analytically computed. Its performances are demonstrated on a set of test cases.

Finally, an adaptation of boundary terms and SPH operators is given for axisymmetric formulation (this work has been performed partly during Zixing Dong internship). The proposed renormalisation factor and the deduced operators have no spurious behaviour next to the axis and show good results for the Poiseuille flow.

Résumé D.1 Tout d'abord, cette annexe présente un résumé des équations discrètes résolues par les solveurs SPH implantés dans les codes SPARTACUS-2D et Sphynx. Ensuite, la dérivation des termes de bord en partant d'un lagrangien est reproduite en suivant Kulasegaram et al. (2004). Cette dérivation rend possible la définition d'opérateurs gradient et divergence anti-adjoints, ce qui constitue une propriété fondamentale pour conserver l'énergie cinétique. Malheureusement, cette dérivation est faite en supposant que les particules constituent un système isolé, ce qui signifie qu'une partie des termes de bord manquent.

Un algorithme pour intégrer de manière analytique en 3-D le gradient du facteur de renormalisation est proposé (voir aussi Mayrhofer et al. (2015)). Cet algorithme décompose l'intersection des éléments de frontières (des triangles ici) avec la sphère du noyau SPH en une somme d'intégrales élémentaires pour lesquelles une formule analytique est donnée. Les performances de l'algorithme sont démontrées sur quelques cas tests classiques.

Enfin, les formulations de bord ainsi que les opérateurs associés sont adaptés à la

> formulation axisymétrique (ce travail a été en partie réalisé durant le stage de Zixing Dong). Le facteur de renormalisation obtenu et les opérateurs proposés n'induisent pas d'*artefact* près de l'axe, contrairement aux peu de travaux publiés à ce sujet. Le cas d'un écoulement de Poiseuille (dans un tuyau cylindrique) est traité avec succès.

D.1 Reminder of equations solved in *SPARTACUS-2D* and Sphynx

The following are the equations solved by *SPARTACUS-2D* and Sphynx (see `https://github.com/remingtonCarmi/Sphynx`). They concern only the turbulent case, as the laminar case is formally identical, with average variables that coincide with the real variables, while the eddy viscosity is constant and equal to the molecular kinematic viscosity. Furthermore the Lagrangian derivative $\frac{\mathrm{D}}{\mathrm{D}t} := \frac{\partial}{\partial t} + \overline{\boldsymbol{u}} \cdot \boldsymbol{\nabla}$ would be replaced by $\frac{\mathrm{d}}{\mathrm{d}t} := \frac{\partial}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla}$.

Continuity equation In Chapter 2

$$\rho_a \left[\beta\gamma_a + (1-\beta)\,\alpha_a\right] = \tilde{\rho}_a = \sum_{b \in \mathcal{P}} M_b w_{ab}, \tag{2.48}$$

where

$$\beta = \exp\left[-K\left(\min\left\{\frac{\alpha_a}{\gamma_a};\, 1\right\} - 1\right)^2\right], \tag{2.49}$$

and $K = 30000$. β could be 1 everywhere for confined flows.

The free-surface correction is removed in Chapter 3 and replaced by a density diffusion formula:

$$\mathrm{d}\left(\gamma_a \rho_a\right) = \gamma_a \mathrm{d}\rho_a + \rho_a \mathrm{d}\gamma_a = \mathrm{d}\left(\sum_{b \in \mathcal{P}} M_b w_{ab}\right) - \gamma_a \delta\rho_a^{\mathrm{i/o}} + \rho_a \delta\gamma_a^{\mathrm{i/o}}, \tag{3.18}$$

where $\delta\rho_a^{\mathrm{i/o}}$ and $\delta\gamma_a^{\mathrm{i/o}}$ are defined by:

$$\delta\rho_a^{\mathrm{i/o}} := \frac{1}{\gamma_a} \sum_{v \in \mathcal{V}^{\mathrm{i/o}}} M_v \boldsymbol{\nabla} w_{av} \cdot \left(\boldsymbol{u}_v - \boldsymbol{v}_v\right) \mathrm{d}t,$$

$$\delta\gamma_a^{\mathrm{i/o}} := \sum_{s \in \mathcal{S}^{\mathrm{i/o}}} \boldsymbol{\nabla}\gamma_{as} \cdot \left(\boldsymbol{u}_s - \boldsymbol{v}_s\right) \mathrm{d}t.$$

Shepard filter α

$$\alpha_a := \sum_{b \in \mathcal{P}} \frac{M_b}{\rho_b} w_{ab}, \tag{2.46}$$

$$\begin{aligned} \alpha_v &:= \sum_{b \in \mathcal{F}} \frac{M_b}{\rho_b} w_{vb}, \\ \alpha_s &:= \sum_{b \in \mathcal{F}} \frac{M_b}{\rho_b} w_{sb}. \end{aligned} \tag{2.47}$$

Governing equation for γ

$$\begin{aligned} \frac{\mathrm{D}\gamma_a}{\mathrm{D}t} &= \sum_{s \in \mathcal{S}} \boldsymbol{\nabla}\gamma_{as} \cdot \boldsymbol{u}_a^{\mathcal{R}_s}, \\ \gamma_a &= 1 \qquad\qquad \text{if } \partial\Omega \cap \Omega_a = \emptyset. \end{aligned} \tag{2.57}$$

Analytical value of $\boldsymbol{\nabla}\gamma_{as}$ for the quintic kernel

$$\boldsymbol{\nabla}\gamma_{as} \coloneqq \left(\int_{\boldsymbol{x}_{v1}}^{\boldsymbol{x}_{v2}} w\,(r)\,\mathrm{d}l\right)\boldsymbol{n}_s, \tag{2.18}$$

with

$$\begin{aligned}\int_{\boldsymbol{x}_{v1}}^{\boldsymbol{x}_{v2}} w\,(r)\,\mathrm{d}l \;=\;& \frac{(q_2\cos\alpha_2)}{\pi}P_{q_0}\,(q_2) - \frac{(q_1\cos\alpha_1)}{\pi}P_{q_0}\,(q_1)\\ &+\frac{q_0^4}{\pi}\left(\frac{105}{64}+\frac{35}{512}q0^2\right)\left[\begin{array}{l}\mathtt{sign}\,(q_2\cos\alpha_2)\ln\left(\dfrac{q_2+|q_2\cos\alpha_2|}{|q0|}\right)\\ -\mathtt{sign}\,(q_1\cos\alpha_1)\ln\left(\dfrac{q_1+|q_1\cos\alpha_1|}{|q0|}\right)\end{array}\right],\end{aligned} \tag{2.51}$$

where:

$$\begin{aligned}P_{q_0}\,(q) \;=\;& \frac{7}{192}q^5-\frac{21}{64}q^4+\frac{35}{32}q^3-\frac{35}{24}q^2+\frac{7}{4}\\ &+q_0^2\left(\frac{35}{768}q^3-\frac{7}{16}q^2+\frac{105}{64}q-\frac{35}{12}\right)\\ &+q_0^4\left(\frac{35}{512}q-\frac{7}{8}\right).\end{aligned} \tag{2.52}$$

Momentum equation

$$\begin{aligned}\frac{\mathrm{D}\overline{\boldsymbol{u}}_a}{\mathrm{D}t} \;=\;& -\frac{1}{\rho_a}\mathbf{Grad}_a^{\gamma^+}\left\{p_b+\tfrac{2}{3}k_b\right\}+\boldsymbol{g}\\ &+\frac{1}{\rho_a}\boldsymbol{\nabla}\cdot(\mu\boldsymbol{\nabla}\,\overline{\boldsymbol{u}})_a\,.\end{aligned} \tag{D.1}$$

Pressure *gradient*

$$\mathbf{Grad}_a^{\gamma^+}\{p_b\} \;\coloneqq\; \frac{\rho_a}{\gamma_a}\sum_{b\,\in\,\mathcal{P}}M_b\left(\frac{p_a}{\rho_a^2}+\frac{p_b}{\rho_b^2}\right)\boldsymbol{\nabla}w_{ab}-\frac{\rho_a}{\gamma_a}\sum_{s\,\in\,\mathcal{S}}\left(\frac{p_a}{\rho_a^2}+\frac{p_s}{\rho_s^2}\right)\rho_s\boldsymbol{\nabla}\gamma_{as}. \tag{2.23}$$

Boundary pressure term

$$\begin{aligned}\rho_v \;&=\; \frac{1}{\alpha_v}\sum_{b\,\in\,\mathcal{F}}V_b\rho_b w_{bv},\\ \frac{p_v}{\rho_v} \;&=\; \frac{1}{\alpha_v}\sum_{b\,\in\,\mathcal{F}}V_b\left(\frac{p_b}{\rho_b}-\boldsymbol{g}\cdot(\boldsymbol{x}_b-\boldsymbol{x}_v)+\frac{u_{bv}^2}{2}\right),\end{aligned} \tag{2.30}$$

$$\begin{aligned}\rho_s \;&=\; \frac{\rho_{v1}+\rho_{v2}}{2},\\ \frac{p_s}{\rho_s} \;&=\; \frac{p_{v1}/\rho_{v1}+p_{v2}/\rho_{v2}}{2},\end{aligned} \tag{2.32}$$

where the vertex particles $v1$ and $v2$ are defined in §2.3.1.

Morris' viscous term

$$\frac{1}{\rho_a}\boldsymbol{\nabla}\cdot(\mu\boldsymbol{\nabla}\,\overline{\boldsymbol{u}})_a=\frac{1}{\gamma_a}\sum_{b\,\in\,\mathcal{P}}M_b\frac{\mu_a+\mu_b}{\rho_a\rho_b}\frac{\overline{\boldsymbol{u}}_{ab}}{r_{ab}^2}\boldsymbol{x}_{ab}.\boldsymbol{\nabla}w_{ab}-\frac{2u_{\tau a}\boldsymbol{u}_{\tau a}}{\gamma_a}\sum_{s\,\in\,\mathcal{S}}|\boldsymbol{\nabla}\gamma_{as}|\,. \tag{D.2}$$

Velocity of vertex particles for turbulent flows

$$\frac{\mathrm{D}\overline{\boldsymbol{u}}_v}{\mathrm{D}t} = \frac{1}{\gamma_v}\sum_{b\,\in\,\mathcal{P}} M_b \frac{\mu_{Te}+\mu_{Tb}}{\rho_v\rho_b}\frac{\overline{\boldsymbol{u}}_{vb}}{r_{vb}^2}\boldsymbol{x}_{vb}\cdot\boldsymbol{\nabla}w_{vb} - \frac{2u_{\tau v}\boldsymbol{u}_{\tau v}}{\gamma_v}\sum_{s\,\in\,\mathcal{S}}|\boldsymbol{\nabla}\gamma_{es}|\,, \tag{2.65}$$

$$\overline{\boldsymbol{u}}_s = \frac{\overline{\boldsymbol{u}}_{v1}+\overline{\boldsymbol{u}}_{v2}}{2}.$$

Laminar friction velocity

$$u_{\tau a}\boldsymbol{u}_{\tau a} = \frac{\nu\boldsymbol{u}_a}{z_a}. \tag{2.37}$$

Smooth turbulent friction velocity

$$\frac{\overline{\boldsymbol{u}}_a}{u_{\tau a}} = \frac{1}{\kappa}\ln\left(\frac{z_a u_{\tau a}}{\nu}\right) + 5.2. \tag{2.64}$$

Wall friction velocity

$$\rho_v u_{\tau v}\boldsymbol{u}_{\tau v} = \frac{1}{\alpha_v}\sum_{b\,\in\,\mathcal{F}} V_b\rho_b u_{\tau b}\boldsymbol{u}_{\tau b}w_{bv}, \tag{2.38}$$

$$\rho_s u_{\tau s}\boldsymbol{u}_{\tau s} = \frac{\rho_{v1}u_{\tau v1}\boldsymbol{u}_{\tau v1}+\rho_{v2}u_{\tau v2}\boldsymbol{u}_{\tau v2}}{2}.$$

Equation of motion

$$\frac{\mathrm{D}\boldsymbol{x}_a}{\mathrm{D}t} = \overline{\boldsymbol{u}}_a, \tag{D.3}$$

$$\begin{aligned}\frac{\mathrm{D}\boldsymbol{x}_v}{\mathrm{D}t} &= \text{velocity of the wall,}\\ \frac{\mathrm{D}\boldsymbol{x}_s}{\mathrm{D}t} &= \text{velocity of the wall.}\end{aligned} \tag{D.4}$$

Equation of state

$$p_a = \frac{\rho_0 c_0^2}{\gamma}\left[\left(\frac{\rho_a}{\rho_0}\right)^{\gamma} - 1\right] \tag{2.4}.$$

Eddy viscosity

$$\nu_T = C_\mu\frac{k^2}{\epsilon}. \tag{2.10}$$

$k-\epsilon$ model

$$\begin{aligned}\frac{\mathrm{D}k_a}{\mathrm{D}t} &= \frac{1}{\rho_a}\mathrm{Lap}_a^{\gamma}\left(\left\{\mu_b+\frac{\mu_{Tb}}{\sigma_k}\right\},\{k_b\}\right) + P_a - \epsilon_a,\\ \frac{\mathrm{D}\epsilon_a}{\mathrm{D}t} &= \frac{1}{\rho_a}\mathrm{Lap}_a^{\gamma}\left(\left\{\mu_b+\frac{\mu_{Tb}}{\sigma_\epsilon}\right\},\{\epsilon_b\}\right) + \frac{\epsilon_a}{k_a}\left(C_{\epsilon1}P_a - C_{\epsilon2}\epsilon_a\right).\end{aligned}$$

Diffusion terms

$$\frac{1}{\rho_a}\mathrm{Lap}_a^{\gamma}\left(\left\{\mu_b+\frac{\mu_{Tb}}{\sigma_k}\right\},\{k_b\}\right)=\frac{1}{\gamma_a}\sum_{b\in\mathcal{P}}M_b\frac{2\mu+\mu_{Ta}/\sigma_k+\mu_{Tb}/\sigma_k}{\rho_a\rho_b}\frac{k_{ab}}{r_{ab}^2}\boldsymbol{x}_{ab}.\boldsymbol{\nabla}w_{ab}, \tag{2.66}$$

$$\begin{aligned}\frac{1}{\rho_a}\mathrm{Lap}_a^{\gamma}\left(\left\{\mu_b+\frac{\mu_{Tb}}{\sigma_\epsilon}\right\},\{\epsilon_b\}\right) &= \frac{1}{\gamma_a}\sum_{b\in\mathcal{P}}M_b\frac{2\mu+\mu_{Ta}/\sigma_\epsilon+\mu_{Tb}/\sigma_\epsilon}{\rho_a\rho_b}\frac{\epsilon_{ab}}{r_{ab}^2}\boldsymbol{x}_{ab}\cdot\boldsymbol{\nabla}w_{ab}\\ &+ \frac{4}{\gamma_a\rho_a}\sum_{s\in\mathcal{S}}\left|\boldsymbol{\nabla}\gamma_{as}\right|\rho_s\frac{u_s^{\star 4}}{\sigma_\epsilon\kappa\delta_{as}}.\end{aligned} \tag{2.72}$$

Boundary terms

$$\begin{aligned}k_v &= \frac{u_{\tau v}^2}{\sqrt{C_\mu}},\\ k_s &= \frac{u_{\tau s}^2}{\sqrt{C_\mu}},\\ u_s^{\star} &= u_{\tau s}.\end{aligned}$$

Production

$$P_a=\nu_{Ta}S_a^2. \tag{2.11}$$

Strain rate

$$S^2{:=}2\boldsymbol{S}:\boldsymbol{S},$$

with

$$\mathbf{Grad}_a^{\gamma^-}\{\overline{\boldsymbol{u}}\}=-\frac{1}{\gamma_a\rho_a}\sum_{b\in\mathcal{P}}M_b\overline{\boldsymbol{u}}_{ab}\otimes\boldsymbol{\nabla}w_{ab}+\frac{1}{\gamma_a}\sum_{s\in\mathcal{S}}\overline{\boldsymbol{u}}_{as}\otimes\boldsymbol{\nabla}\gamma_{as}, \tag{2.24}$$

or

$$\nu_{Ta}S_a^2=-\frac{1}{2\gamma_a}\sum_{b\in\mathcal{P}}M_b\frac{\mu_{Ta}+\mu_{Tb}}{\rho_a\rho_b}\frac{\overline{u}_{ab}^2}{r_{ab}^2}\boldsymbol{x}_{ab}.\boldsymbol{\nabla}w_{ab}-\frac{2}{\gamma_a}\sum_{s\in\mathcal{S}}u_{\tau s}^2\overline{u}_{as}\left|\boldsymbol{\nabla}\gamma_{as}\right|.$$

D.2 Derivation of the repulsive force from a Lagrangian

This derivation is done following Kulasegaram et al. (2004). In Lagrangian mechanisms, the dynamics of particles can be derived from the Lagrange equations for an **insulated system of particles**:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial\boldsymbol{u}_a}\right)-\left(\frac{\partial L}{\partial\boldsymbol{x}_a}\right)=0, \tag{D.5}$$

where the Lagrangian L is defined by:

$$L\left(\boldsymbol{x}_a,\boldsymbol{u}_a\right)=E_{ke}\left(\boldsymbol{u}_a\right)-E_{int}\left(\boldsymbol{x}_a\right)-E_{ext}\left(\boldsymbol{x}_a\right). \tag{D.6}$$

The energies E_{ke}, E_{int} and E_{ext} are respectively the kinetic energy, the internal energy and the external energy. The kinetic energy is given by (D.7a). If the external forces are due to the

gravitational field $\boldsymbol{g}$ the external energy is given by (D.7b) and for an adiabatic reversible process, the internal energy can be written as (D.7c)

$$E_{ke} = \frac{1}{2}\sum_{b\,\in\,\mathcal{P}} M_b \boldsymbol{u}_b^2, \tag{D.7a}$$

$$E_{ext} = -\sum_{b\,\in\,\mathcal{P}} M_b \boldsymbol{x}_b \cdot \boldsymbol{g}, \tag{D.7b}$$

$$E_{int} = \sum_{b\,\in\,\mathcal{P}} M_b\, e_{int}\,(\rho_b), \tag{D.7c}$$

and the energy accumulated per unit of mass e_{int} varies with the density ρ according to(D.8).

$$p = \rho^2 \frac{\mathrm{d}e_{int}}{\mathrm{d}\rho}. \tag{D.8}$$

Finally we define the density variation to be:

$$\begin{aligned} \rho(\boldsymbol{x}) &= \frac{\displaystyle\int_\Omega \sum_{b\,\in\,\mathcal{P}} M_b \delta\,(\boldsymbol{x}' - \boldsymbol{x}_b)\; w\,(\boldsymbol{x} - \boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}'}{\displaystyle\int_\Omega w\,(\boldsymbol{x} - \boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}'} \\ &= \frac{\displaystyle\sum_{b\,\in\,\mathcal{P}} M_b\, w\,(\boldsymbol{x} - \boldsymbol{x_b})}{\displaystyle\int_{\Omega\cap\Omega_r} w\,(\boldsymbol{x} - \boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}'}. \end{aligned} \tag{D.9}$$

Hence it follows:

$$\gamma\,(\boldsymbol{x})\;\rho\,(\boldsymbol{x}) = \sum_b M_b\, w\,(\boldsymbol{x} - \boldsymbol{x_b}), \tag{D.10}$$

which is a continuous function of the position $\boldsymbol{x}$ and gives us a way to compute the value of $\rho_a := \rho\,(\boldsymbol{x}_a)$ if we know the positions of all particles.

Now we are able to derive the momentum equation from the equation of Lagrange (D.5) and obtain for each particle a:

$$M_a \frac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t} = \boldsymbol{F}_a^p + \boldsymbol{F}_a^B + M_a \boldsymbol{g}, \tag{D.11}$$

where the internal forces due to the pressure $\boldsymbol{F}_a^p$ and due to the boundary effect $\boldsymbol{F}_a^B$ are defined by:

$$\begin{aligned} \boldsymbol{F}_a^p + \boldsymbol{F}_a^B &= -\frac{\partial E_{int}}{\partial \boldsymbol{x}_a} \\ &= -\frac{\partial}{\partial \boldsymbol{x}_a}\left[\sum_{b\,\in\,\mathcal{P}} M_b\, e_{int}\,(\rho_b)\right] \\ &= -\sum_{b\,\in\,\mathcal{P}} M_b \frac{\mathrm{d}e_{int}}{\mathrm{d}\rho_b}\frac{\mathrm{d}\rho_b}{\mathrm{d}\boldsymbol{x}_a} \\ &= -\sum_{b\,\in\,\mathcal{P}} M_b \frac{p_b}{\rho_b^2}\frac{\mathrm{d}\rho_b}{\mathrm{d}\boldsymbol{x}_a}, \end{aligned} \tag{D.12}$$

but we can derive the density using its definition (D.10):

$$\begin{aligned}\frac{\mathrm{d}\left(\gamma_b\rho_b\right)}{\mathrm{d}\boldsymbol{x}_a} &= \gamma_b\frac{\mathrm{d}\rho_b}{\mathrm{d}\boldsymbol{x}_a}+\rho_b\frac{\mathrm{d}\gamma_b}{\mathrm{d}\boldsymbol{x}_a} &= \gamma_b\frac{\mathrm{d}\rho_b}{\mathrm{d}\boldsymbol{x}_a}+\delta_{ab}\rho_a\boldsymbol{\nabla}\gamma_a\\ &= \sum_{c\in\mathcal{P}}M_c\,\frac{\mathrm{d}w_{bc}}{\mathrm{d}\boldsymbol{x}_a} &= M_a\boldsymbol{\nabla}_a w_{ab}+\delta_{ab}\sum_{c\in\mathcal{P}}M_c\,\boldsymbol{\nabla}_a w_{ac},\end{aligned}$$

and (D.12) becomes

$$\begin{aligned}\boldsymbol{F}_a^p+\boldsymbol{F}_a^B &= -\sum_{b\in\mathcal{P}}M_b\,\frac{p_b}{\rho_b^2}\frac{\mathrm{d}\rho_b}{\mathrm{d}\boldsymbol{x}_a}\\ &= -\sum_{b\in\mathcal{P}}M_b\,\frac{p_b}{\gamma_b\rho_b^2}\left[M_a\boldsymbol{\nabla}_a w_{ab}+\delta_{ab}\left(\sum_{c\in\mathcal{P}}M_c\,\boldsymbol{\nabla}_a w_{ac}\right)-\delta_{ab}\rho_a\boldsymbol{\nabla}\gamma_a\right]\\ &= -\sum_{b\in\mathcal{P}}M_aM_b\left(\frac{p_b}{\gamma_b\rho_b^2}+\frac{p_a}{\gamma_a\rho_a^2}\right)\boldsymbol{\nabla}_a w_{ab}+\frac{M_ap_a}{\gamma_a\rho_a}\boldsymbol{\nabla}\gamma_a.\end{aligned}$$

We can see in these calculi that the density ρ and the function γ must depend only on positions if we expect the correction to be effective. So we must use a time discretisation where the variation of the quantity $\gamma_a\rho_a$ depends only on the variation of positions such as the temporal scheme (2.42). Such a scheme gives in fact at any time:

$$\gamma_a\rho_a=\left(\gamma_a^0\rho_a^0-\sum_{b\in\mathcal{P}}M_b\,w_{ab}^0\right)+\sum_{b\in\mathcal{P}}M_b\,w_{ab}. \tag{D.13}$$

If the field ρ_a^0 is initialised at such that $\gamma_a^0\rho_a^0=\sum_{b\in\mathcal{P}}M_b\,w_{ab}^0$, the derivation from the Lagrangian holds, but the pressure field will be chaotic. If the field ρ_a^0 is initialised at a density reference, say, ρ_0 the pressure field will be smoother. But, doing so, we will generate distinctions between particles from a free surface and from the inner fluid for instance. Indeed, if a particle initially next to a free surface with the same density ρ_0 as an other particle in the fluid, is forced to go in the middle of the fluid (due to a wave break for example), its density will be multiplied by almost $\frac{3}{2}$. That is to say that we are setting a non-homogeneous fluid by initialising the density at ρ_0 to correct the error done by the interpolation (D.10).

D.3 Skew-adjointness of the operators *gradient* and *divergence*

The goal of this appendix is to prove that the two operators *gradient* and *divergence* defined by Kulasegaram et al. (2004) method are skew-adjoint. We recall that they are defined by:

$$\begin{aligned}\mathbf{Grad}_a^K\left\{A_b\right\} &:= \rho_a\sum_{b\in\mathcal{P}}M_b\left(\frac{A_a}{\gamma_a\rho_a^2}+\frac{A_b}{\gamma_b\rho_b^2}\right)\boldsymbol{\nabla}w_{ab}-\frac{A_a}{\gamma_a}\boldsymbol{\nabla}\gamma_a,\\ \mathrm{Div}_a^K\left\{\boldsymbol{A}_b\right\} &:= -\frac{1}{\gamma_a\rho_a}\sum_{b\in\mathcal{P}}M_b\boldsymbol{A}_{ab}\cdot\boldsymbol{\nabla}w_{ab}+\frac{1}{\gamma_a}\boldsymbol{A}_a\cdot\boldsymbol{\nabla}\gamma_a,\end{aligned} \tag{2.17}$$

for two arbitrary fields of scalar $\{A_b\}$ and of vector $\{\boldsymbol{B}_b\}$. We also recall that the skew-adjointness property means that the following property holds **when boundary terms are zero**[1]:

$$\left\langle\mathbf{Grad}_b^K\left\{A_a\right\},\boldsymbol{B}_b\right\rangle=-\left(A_b,\,\mathrm{Div}_b^K\left\{\boldsymbol{B}_a\right\}\right),$$

[1]For instance zero pressure at the free-surface and zero velocity at the wall boundaries, no inlet/outlet.

where the dot products are defined by:

$$\begin{aligned}\langle \boldsymbol{A}_b,\, \boldsymbol{B}_b\rangle &:= \sum_{b\,\in\,\mathcal{P}} V_b \boldsymbol{A}_b \cdot \boldsymbol{B}_b,\\ (A_b,\, B_b) &:= \sum_{b\,\in\,\mathcal{P}} V_b A_b B_b.\end{aligned}$$

Thus, we have:

$$\begin{aligned}\left\langle \mathbf{Grad}_b^K\left\{A_a\right\},\, \boldsymbol{B}_b\right\rangle &= \sum_{b\,\in\,\mathcal{P}} V_b \mathbf{Grad}_b^K\left\{A_a\right\}\cdot \boldsymbol{B}_b\\ &= \sum_{b\,\in\,\mathcal{P}} V_b\left[\rho_b \sum_{c\,\in\,\mathcal{P}} M_c\left(\frac{A_b}{\gamma_b\rho_b^2}+\frac{A_c}{\gamma_c\rho_c^2}\right)\nabla w_{bc} - \frac{A_b}{\gamma_b}\nabla\gamma_b\right]\cdot \boldsymbol{B}_b\\ &= \sum_{b,\,c\,\in\,\mathcal{P}} M_b M_c \frac{A_b}{\gamma_b\rho_b^2}\nabla w_{bc}\cdot \boldsymbol{B}_b + \underbrace{\sum_{b,\,c\,\in\,\mathcal{P}} M_b M_c \frac{A_c}{\gamma_c\rho_c^2}\nabla w_{bc}\cdot \boldsymbol{B}_b}_{=\,-\sum_{b,\,c\,\in\,\mathcal{P}} M_b M_c \frac{A_b}{\gamma_b\rho_b^2}\nabla w_{bc}\cdot \boldsymbol{B}_c}\\ &\quad - \sum_{b\,\in\,\mathcal{P}} V_b A_b \frac{1}{\gamma_b}\boldsymbol{B}_b\cdot\nabla\gamma_b\\ &= -\sum_{b\,\in\,\mathcal{P}} V_b A_b\left[-\frac{1}{\gamma_b\rho_b}\sum_{c\,\in\,\mathcal{P}} M_c \boldsymbol{A}_{bc}\cdot\nabla w_{bc} + \frac{1}{\gamma_b}\boldsymbol{A}_b\cdot\nabla\gamma_b\right]\\ &= -\sum_{b\,\in\,\mathcal{P}} V_b A_b \mathrm{Div}_b^K\left\{\boldsymbol{B}_c\right\}\\ &= -\left(A_b,\, \mathrm{Div}_b^K\left\{\boldsymbol{B}_a\right\}\right),\end{aligned} \tag{D.14}$$

where we have used the antisymmetry of the kernel gradient and have exchanged the mute indices b and c to complete the proof. The proof of the skew-adjointness of the operator (2.7) presented in Chapter 1 is just a special case of the present proof where γ_a is fixed at 1 everywhere and $\nabla\gamma = 0$.

D.4 Algorithm for the analytical computation of $\nabla\gamma_a$ in 3-D

This Section is to describe an exact integration of $\nabla\gamma_{as}$ in 3-D, and was published in Mayrhofer et al. (2015).

The general principle will be discussed at first before describing the details of the domain decomposition algorithm and the formulae for the special cases. This Section will be concluded by showing the possibility to optimise the calculation on a plane wall.

Clearly the boundary of a three dimensional geometry is two dimensional. In order to discretise the boundary it is required to generalise the 1-D line segments to a 2-D equivalent. The most basic bounded 2-D element is a triangle. It can be described by 3 vertices which are always coplanar. Thus in the following we will assume that a boundary segment is given by a triangle and its virtual position, required for the imposition of boundary conditions, is set to be the triangle's barycentre.

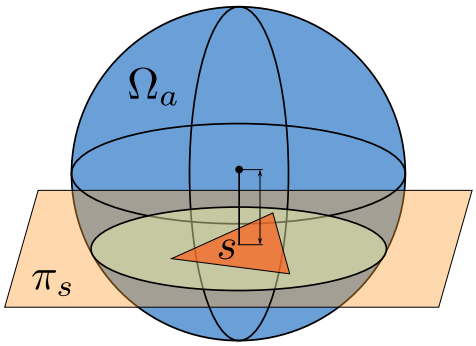


Figure D.1: Illustration of the general problem.

Thus, $\nabla\gamma_{as}$ is the integral of the Wendland kernel over a triangle. More precisely it is the integration of a radial polynomial over the intersection of a triangle with a sphere as illustrated in Figure D.1. Clearly this intersection is not necessarily a triangle, further complicating the integration. Due to the inherent singularity of the Wendland kernel at the origin (only twice continuously differentiable, i.e. $w \in C^2(\mathbb{R}^n)$) it is not advisable to use a numerical algorithm. The authors tried several numerical approaches and the only one providing promising results was an adaptive integration which was not viable due to its computational cost. Due to this it was decided to develop an analytical formula, as it was done for 2-D in Ferrand et al. (2013b).

The integration of an arbitrary triangle / sphere intersection is too complicated and thus it was decided to decompose the surface integration domain, denoted by s into certain building blocks for which it would be easier to find an analytical solution. This domain decomposition will be discussed in detail in the next Section. The formulae for the building blocks will be shown in Section D.4.2.

In contrast to the approach by Amicarelli et al. (2012) which uses square segments the triangular decomposition allows more flexibility. Additionally, traditional meshing software can be used in order to create the surface mesh for arbitrary complex geometries.

D.4.1 The domain decomposition algorithm

Before the algorithm for the decomposition is given, consider the 8 possible configurations that can result from the triangle / sphere intersection as shown in Figure D.2. As mentioned above, the integration domain $s_{as} = \partial s \cap \Omega_a$ consists of an intersection between a triangle and a circle. In the following, one segment s will be investigated with its three associated vertices $\{v_1, v_2, v_3\}$. The number n_v will determine how many vertices are inside the integration domain and n_e is the number of edges intersecting with s_{as}, the boundary of the integration domain.

A systematic description of the eight possible cases via a decision tree is given below.

- All vertices inside the domain ($n_v = 3 \Rightarrow n_e = 0$) $\Rightarrow$ Case 1
- One vertex outside the domain ($n_v = 2 \Rightarrow n_e = 2$) $\Rightarrow$ Case 2
- Two vertices outside the domain ($n_v = 1$)
  - Two edges intersecting ($n_e = 2$) $\Rightarrow$ Case 3
  - All edges intersecting ($n_e = 3$) $\Rightarrow$ Case 4
- All vertices outside the domain ($n_v = 0$)
  - No edges intersecting ($n_e = 0$) $\Rightarrow$ Case 5

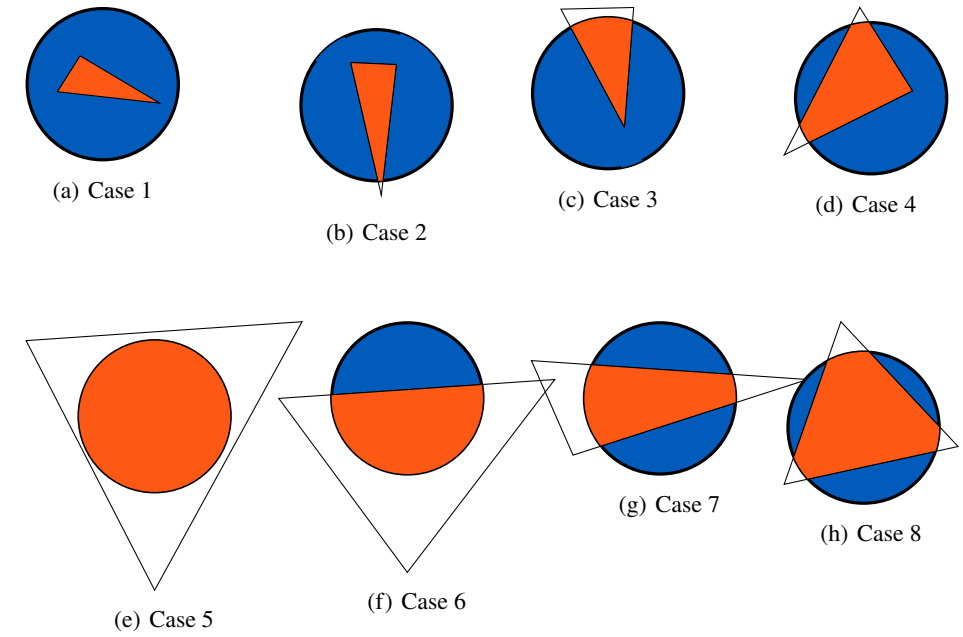


Figure D.2: The 8 different cases.

- One edge intersecting ($n_e = 1$) $\Rightarrow$ Case 6
- Two edges intersecting ($n_e = 2$) $\Rightarrow$ Case 7
- All edges intersecting ($n_e = 3$) $\Rightarrow$ Case 8

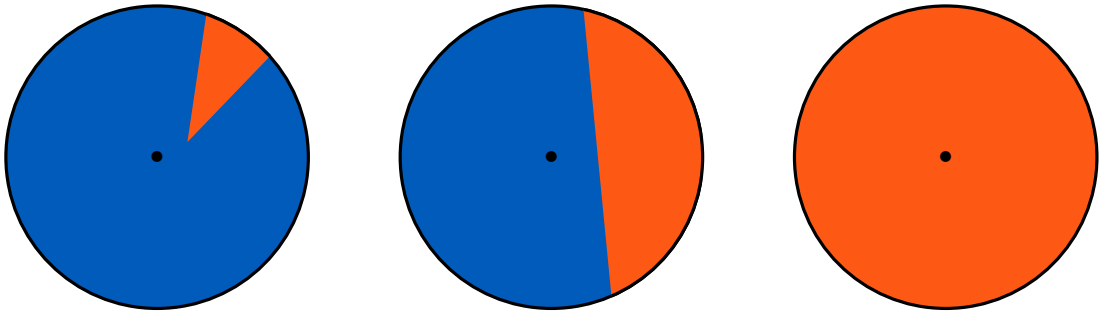


Figure D.3: The 3 building blocks.

After identifying the different cases the next step is to decompose the specific domain into the three basic building blocks shown in Figure D.3. From the left to right there is the circular sector, the circular segment and the full circle. In Figure D.4 the different decomposition algorithms are shown.

In the following Case 1 will be explained as an example. The detailed procedure for this can be seen in Figure D.5. The integration domain is a triangle, completely inside the support. In order to compute the integral, three circular sectors need to be computed. Circular Segment

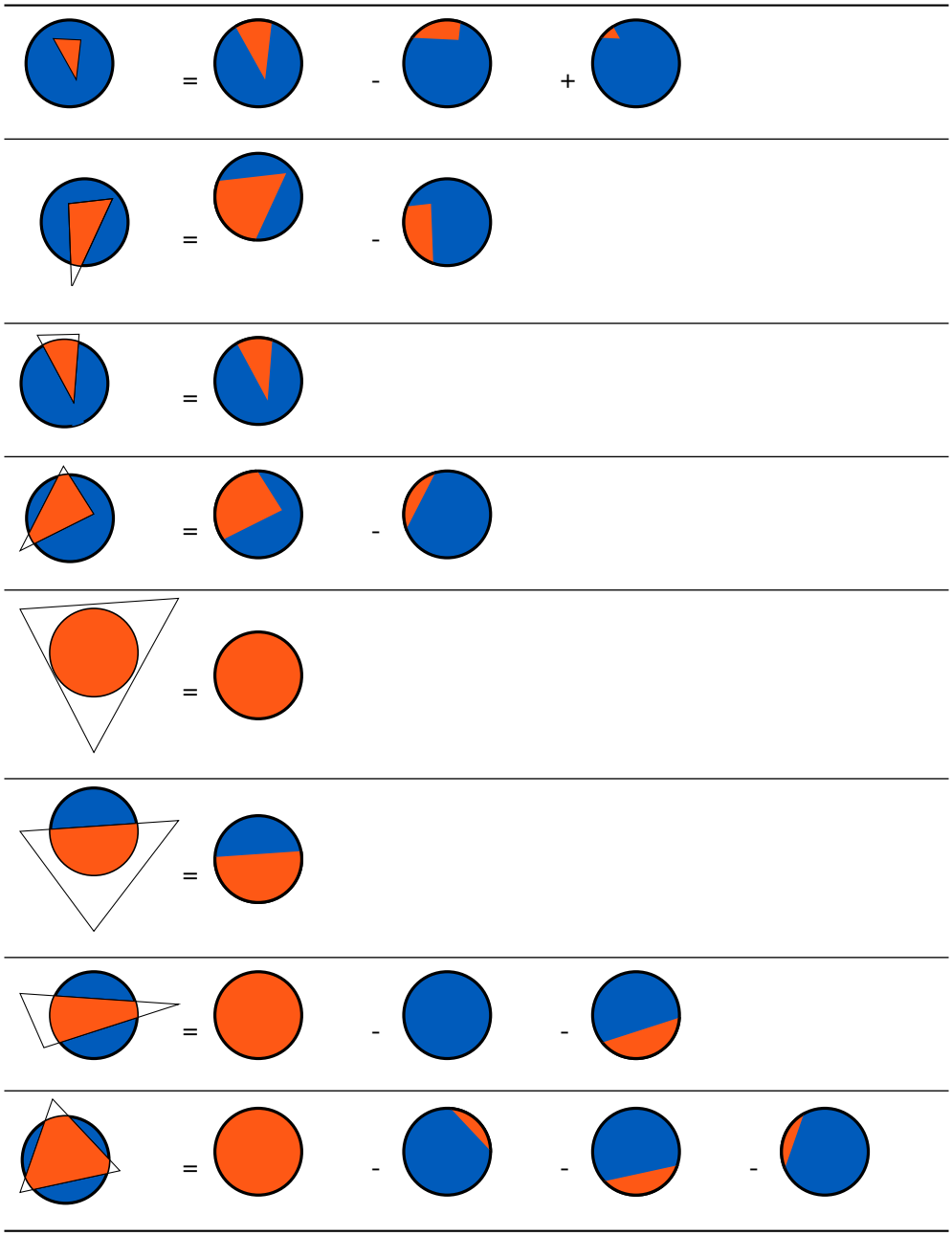


Figure D.4: Illustration of the domain decompositioning algorithm.

1 contains the triangle and its centre is vertex v_1. In the next step Circular Segment 2 will be subtracted in order to remove the part of Circular Segment 1 that does not contain the triangle. However, there is Circular Segment 3 that does not overlap with Circular Segment 1 and thus it needs to be added in the final step so that the integral really only covers the triangle domain.

D.4.2 Analytical formulae for the basic domains

After decomposing the domain in the previous Section, the analytical formulae for the integral (2.18) over the three basic domains will be given for the Wendland kernel (1.86). Let us first introduce the non-dimensionalised integral I:

$$I = \frac{1}{h^{m-1}\alpha_m} \int_s w_{ab} \mathrm{d}r_b. \tag{D.15}$$

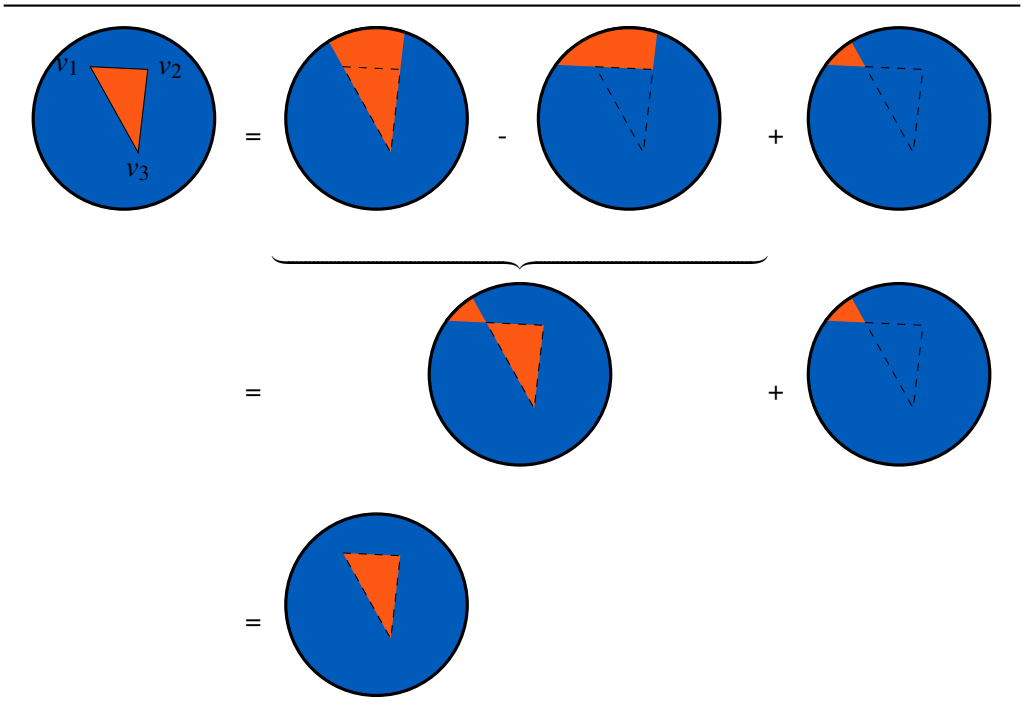


Figure D.5: Algorithm for case 1 in detail.

Equation (2.18) then reads:

$$\nabla\gamma_{as} = \alpha_m h^{m-1} I \boldsymbol{n}_s. \tag{D.16}$$

Circular sector

The case of a circular sector needs to be simplified to two special circular sectors as illustrated in Figure D.6(a). One special circular segment can be identified by the distance d of the particle to the plane, the distance to the projected centre r and one angle θ as shown in Figure D.6(b). The formula for one such special circular segment is then formally given by

$$I_{sector}\,(d,r,\theta) = \int_{q=\sqrt{d^2+r^2}}^{2} w\,(q)\left(\theta - \arcsin\left(\frac{r\sin\theta}{\sqrt{q^2-d^2}}\right)\right) q\mathrm{d}q. \tag{D.17}$$

With the help of a standard symbolic mathematics software it becomes

$$\begin{aligned} I_{sector}\,(d,r,\theta) &= \frac{2}{7}\left(\theta - \arcsin\left(\tfrac{r\sin\theta}{\sqrt{4-d^2}}\right)\right) \\ &+ \Big[\Psi_{d,r,\theta}\,(q) + \Psi_{-d,r,\theta}\,(q) - r\sin\theta\, P_{d,r,\theta}\,(q)\sqrt{q^2-d^2-r^2\sin^2(\theta)} \\ &\quad -r\sin\theta\, Q_{d,r,\theta}\ln\left(q+\sqrt{q^2-d^2-r^2\sin^2(\theta)}\right)\Big]_{q=\sqrt{r^2+d^2}}^{q=2}, \end{aligned} \tag{D.18}$$

where

$$\Psi_{d,r,\theta}\,(q) = \frac{1}{14}\arctan\left(\frac{r^2\sin^2\theta + (d+q)d}{r\sin\theta\sqrt{q^2-d^2-r^2\sin^2(\theta)}}\right)\left(2-(2+5d+4d^2)\left(1-\frac{d}{2}\right)^5\right), \tag{D.19}$$

and

$$\begin{aligned} P_{d,r,\theta}(X) &= \frac{1}{2688}[-924d^4 - 1120r^2\sin^2\theta - 2800d^2 - 728r^2\sin^2\theta d^2 - 224r^4\sin^4\theta + 1344 \\ &\quad +(48d^2r^2\sin^2\theta + 504r^2\sin^2\theta + 15r^4\sin^4\theta + 57d^4 + 1176d^2)X \\ &\quad +(-252d^2 - 112r^2\sin^2\theta - 560)X^2 + (22d^2 + 10r^2\sin^2\theta + 336)X^3 - 84X^4 + 8X^5], \end{aligned} \tag{D.20}$$

$$\Psi(d,r) = \sqrt{4 - d^2 - r^2}, \tag{D.21}$$

and

$$Q_{d,r,\theta} = \frac{1}{2688}[2520d^4 + 105d^6 + (105d^4 + 1680d^2)r^2\sin^2\theta + (504 + 63d^2)r^4\sin^4\theta + 15r^6\sin^6\theta]. \tag{D.22}$$

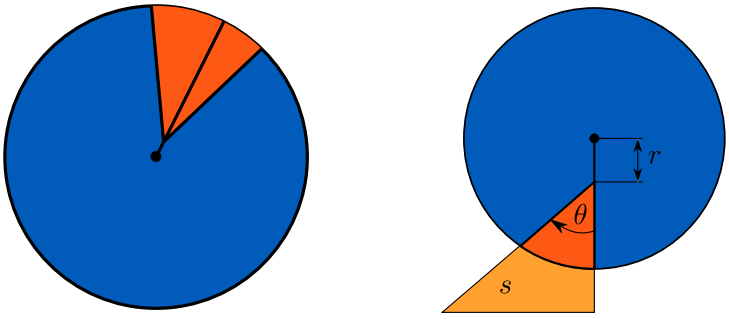
The above formula is valid for $0 < \theta < \pi/2$, $0 \le r \le 2$ and $0 \le d \le 2$.

Circular segment

Let r denote the minimum distance from the circular segment to $\boldsymbol{r}_a$ projected to the plane π_s. Assume furthermore that the circular segment does not cover this projected point, then the formula for the integral is given by

$$I_{segment}(d,r) = 2I_{sector}\left(d, r, \frac{\pi}{2}\right). \tag{D.23}$$

Note that many algebraic simplifications and optimisations can be done in $I_{sector}(d,r,\theta)$ when $\theta = \frac{\pi}{2}$.



(a) Decomposition of the circular sector.

(b) Variables of a special circular sector.

Figure D.6: The circular sector in detail.

Full circle

The integral for the case of a full circle only depends on the distance to the plane d and can be calculated as

$$I_{full}(d) = 2I_{segment}(d,0) = \frac{2}{7}\pi(2 + 5d + 4d^2)\left(1 - \frac{d}{2}\right)^5, \tag{D.24}$$

D.4.3 Optimisation on plane walls

On plane walls the values of γ_a and $\nabla\gamma_a$ can be easily computed in terms of the dimensionless distance d normal to the wall, normalised by the smoothing length h. They are given as

$$\gamma_a(d) = 1 - \frac{1}{512}(2 - d)^6\left(4 + 6d + 3d^2\right)^6, \tag{D.25}$$

and

$$\nabla\gamma_a(d) = \alpha_3 h^2 I_{full}(d)\,. \tag{D.26}$$

This avoids the time-integration of γ for particles only influenced by a plane wall. This reduces the possible time-stepping error and in case a higher order time-stepping scheme is used, lowers the computation time, as $\nabla\gamma_{as}$ has to be computed only once.

D.4.4 Simulations

In the following Section the algorithm provided above will be used to perform three different simulations. The first one will be used to compare the proposed algorithm to the analytical value when approaching a plane wall. The second simulation will show a box with a complex floor that is filled with water in order to investigate the spurious velocities. Finally, a dynamic flow is simulated based on the geometry of the second SPHERIC test case.

D.4.5 Plane wall

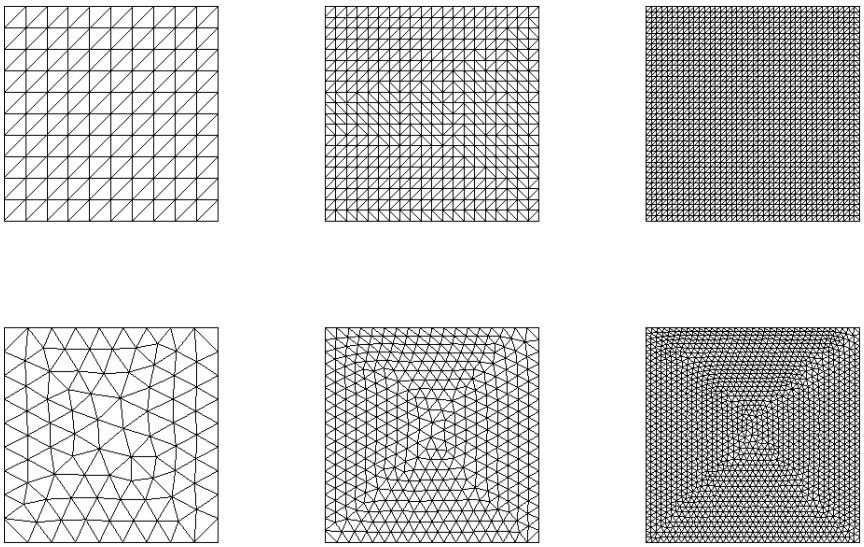


Figure D.7: Discretisations of the plane walls.

The first simulation will consist of three particles approaching six different plane walls. The discretisation of these walls is illustrated in Figure D.7 and as it can be seen there are three planes with a regular discretisation and three with a random mesh. For each of those two sets there is one plane where the triangle size is of the order of Δr (particle size), one with double and one with quadruple resolution. Furthermore for each of those six walls there are three particles, initially at $z = 4\Delta r$ ($h/\Delta r = 2$) that approach the wall with constant speed $c_0/10$, where c_0 is the numerical speed of sound in weakly compressible SPH Violeau (2012). The difference between these three particles is that one is located on top of a vertex, one is on top of an edge and one is located inside a triangle.

The analytical formula for $\nabla\gamma$ on a plane wall for the Wendland kernel (1.86) is given by Equation (D.26). Note that the optimisation presented in Section D.4.3 is not used in this simulation. In Figure D.8 the relative error for the different walls and particle positions can be seen. Clearly, the error is as large as numerical epsilon for the used double precision variables and no configuration showed any larger errors, indicating that the formulation adapts perfectly to any given discretisation.

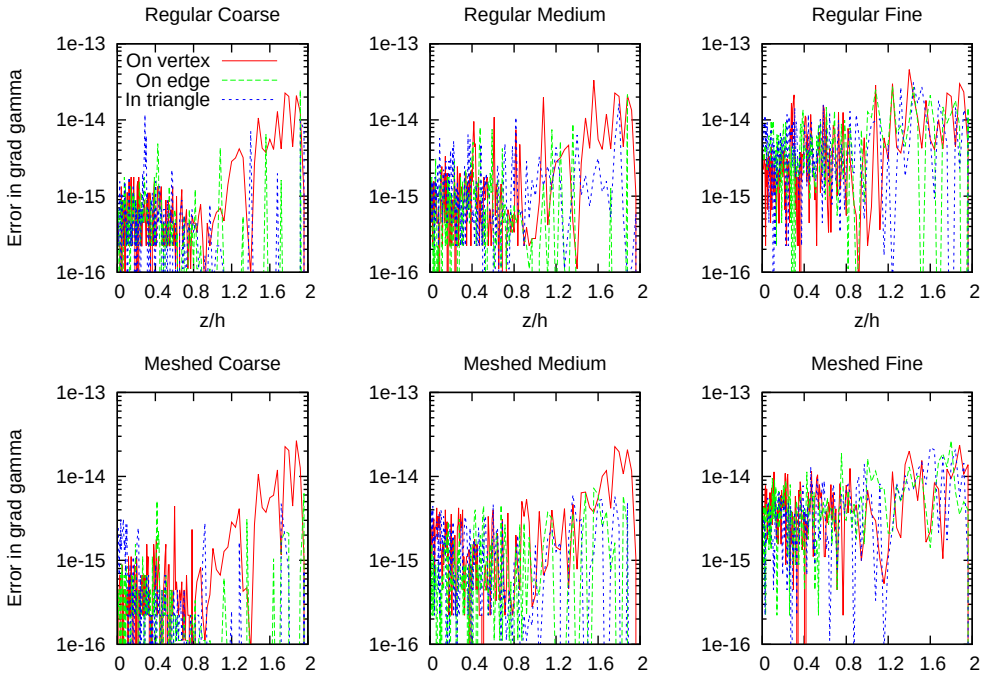


Figure D.8: Error of $\nabla\gamma$ for different walls.

D.4.6 Still water

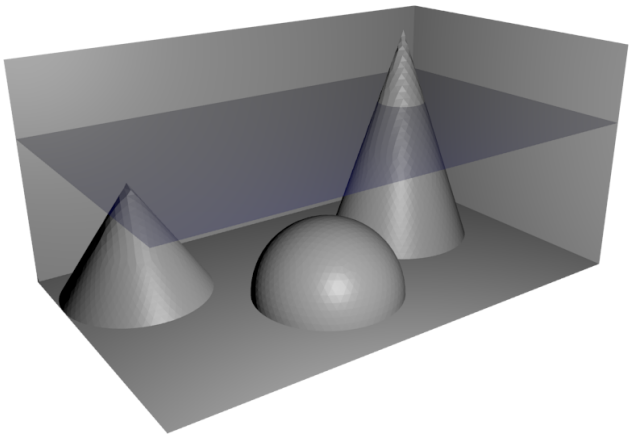
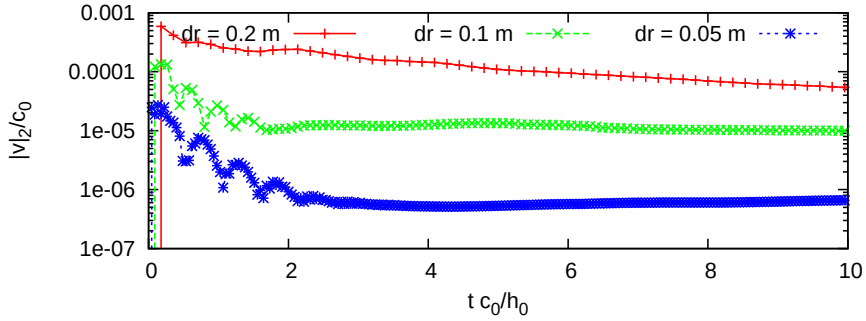


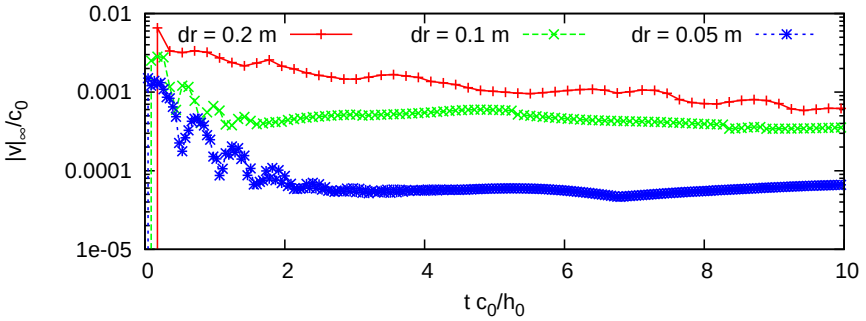
Figure D.9: Geometry of the still water test case.

After demonstrating the algorithm in a simple case the question of course is if the algorithm also performs well when using highly complex geometries. In the following the box shown in Figure D.9 will be used in order to quantify the decay of velocity in a flow at rest under the influence of gravity. Note that for illustration purposes two side walls were removed to allow viewing the inside of the box. The box is partially filled with water so that there is one submerged half-sphere, one submerged cone and one cone which pierces the surface. Initially all the particles have uniform density (thus uniform pressure) and thus an initial velocity is expected which should die down rather quickly. The optimisation presented in Section D.4.3 is used in all following computations.

The viscosity is given by $\nu = 0.1\,\mathrm{m\,s^{-2}}$ and the numerical speed of sound is $c_0 = 45\,\mathrm{m\,s^{-1}}$,



(a) L_2 norm



(b) L_∞ norm

Figure D.10: Residual velocity.

which is based on the initial water-depth of $h_0 = 2\,\text{m}$. The Neumann boundary conditions are implemented using the technique described in Mayrhofer et al. (2013). In Figure D.10 the magnitude of velocity can be seen in both L_2 and L_∞ norms. The L_2 norm is given as

$$\frac{|v|_2}{c_0} = \sqrt{\frac{1}{|\mathcal{P}|}\sum_{a\in\mathcal{P}}\left(\frac{\|\boldsymbol{v}_a\|_2}{c_0}\right)^2}. \tag{D.27}$$

In order to see whether the formulation provides convergence, three different resolutions are compared. After the expected initial damping, the velocity remains more or less constant and convergence can be observed. The final pressure distribution can be seen in Figure D.11.
In Figure D.12 the present algorithm was compared to a simulation which was conducted using the approximation

$$\nabla\gamma_{as} = S_s \boldsymbol{n}_s w_{as}, \tag{D.28}$$

where S_s is the surface area of boundary segment s. In the lower left corner of Figure D.12(a) a particle can be seen penetrating a wall with a value of γ_a significantly lower than the expected value of $\gamma_a \approx 0.25$. At later stages of the simulations further particles penetrate the wall. It shall be noted that due to the modification presented in Section D.4.3 the particles along plane walls do not penetrate due to their correct value of γ_a. It can be observed in Figure D.12(b) that no particle penetrate the walls in case the algorithm presented in Section D.4 is used. This is due to the fact that the value of $\nabla\gamma_a$ is correctly computed and so the only error in γ_a is coming from the time-integration which is negligible.

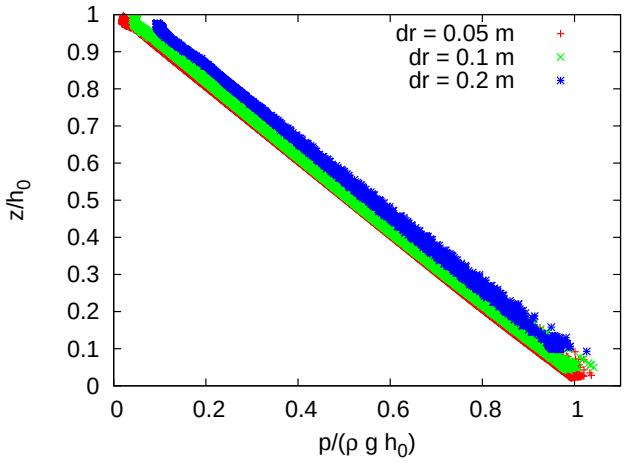
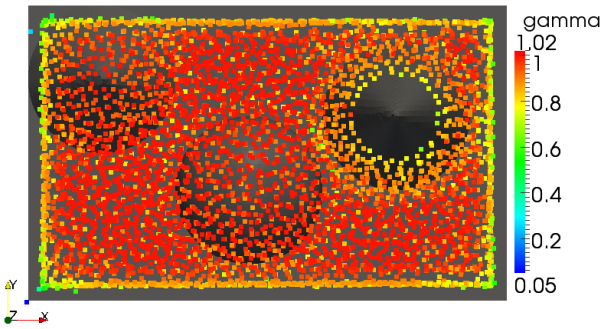
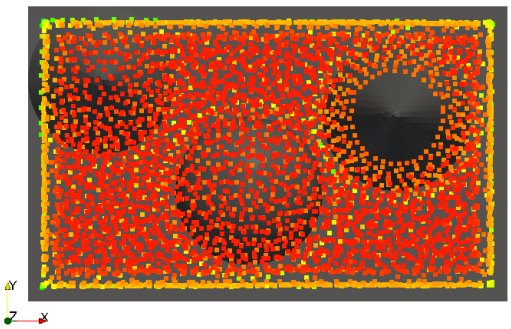


Figure D.11: Pressure of all particles at the end of the simulation.



(a) Approximate $\nabla\gamma_{as}$, Equation (D.28)



(b) Analytical $\nabla\gamma_{as}$, Section D.4

Figure D.12: Comparison between the approximate and the analytical approach.

D.4.7 Dam-break with obstacle

The final simulation will show a violent free-surface flow in form of a dam-break with an obstacle. The geometry used is the same as given in the second SPHERIC validation test (`https://wiki.manchester.ac.uk/spheric/index.php/Test2`). Compared to the proposed viscosity of $\nu = 10^{-6}\,\mathrm{m\,s^{-2}}$ a value of $\nu = 10^{-2}\,\mathrm{m\,s^{-2}}$ is used in order to ensure a laminar flow. This is due to the fact that no turbulence models have been implemented in our code and thus the results would not be accurate enough with a smaller viscosity. The pressure probes and water depth probes are

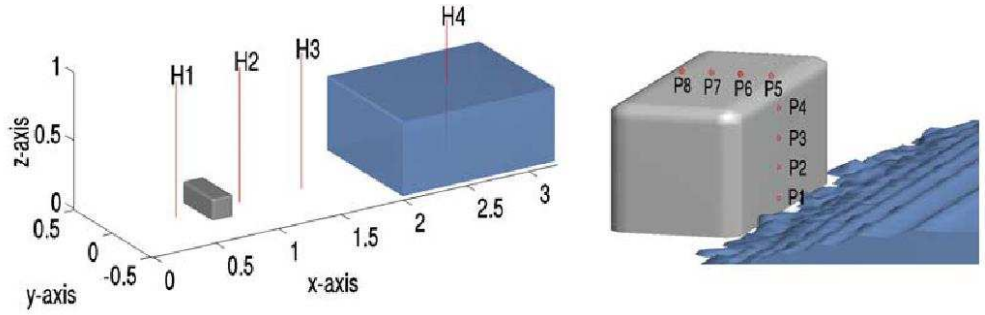


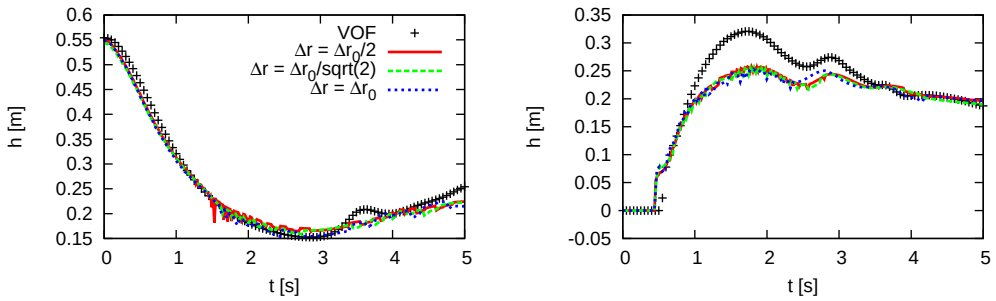
Figure D.13: Location of probes according to Kleefsman et al. (2005).

located at exactly the same position as in the SPHERIC validation test shown in Figure D.13. In order to have a comparison with a different approach the same simulation was conducted using a Volume-of-Fluid (VOF) code. It is expected that this reference solution will not match the solution computed with the SPH method presented above as the former is computed using a multiphase approach. However, before the impact of the fluid onto the obstacle we expect similar results as the air phase has negligible impact on this stage of the flow.

The simulation was run for three different resolutions, the first corresponding to the one of the validation test which is $\Delta r_0 = 0.55/30\,\mathrm{m}$ and the other two being $\Delta r = \Delta r_0/\sqrt{2}$ and $\Delta r = \Delta r_0/2$. The numerical speed of sound is set to $c_0 = 40\,\mathrm{m\,s^{-1}}$ and the modified volume diffusion term presented in Mayrhofer et al. (2013) is used.

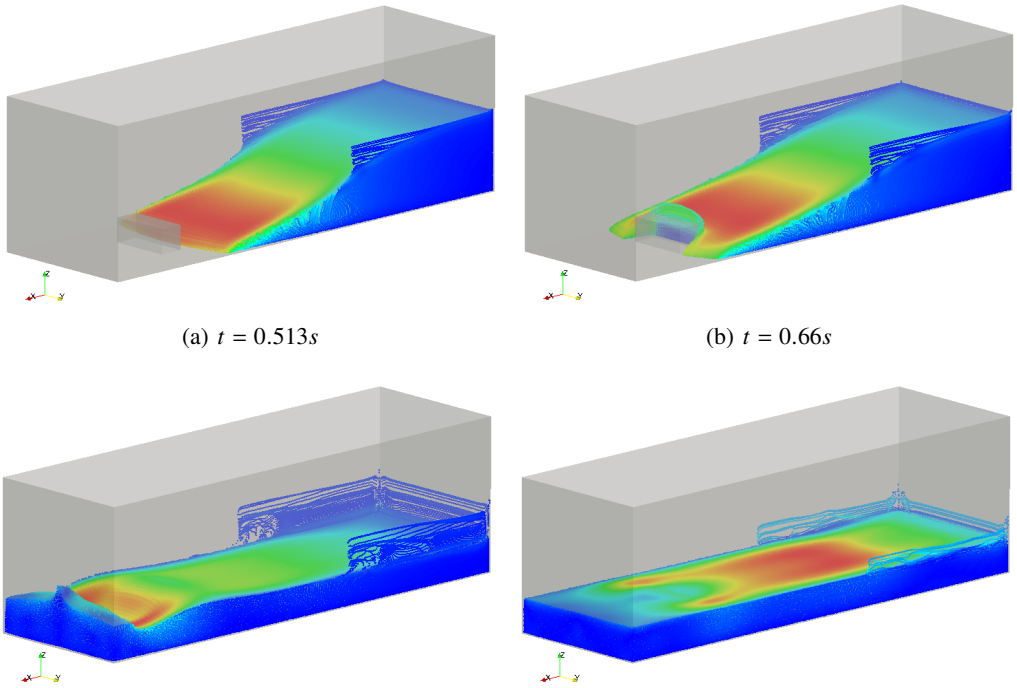
In Figure D.14 the result of two different probes measuring the water levels can be observed and Figure D.15 shows the flow at different stages during the simulation. The probe on the left-hand side is located inside the initial cube of water and the decay agrees nicely with the VOF reference solution as expected. It can be seen that there is more fluid being held back by the obstacle as indicated by the difference in levels in Probe H2. This also implies that the wave reflected by the obstacle is only visible in the VOF solution as seen in the plot of Probe H4 after $3.5s$. The cause for this difference is most likely related to be air entrapped downstream of the obstacle, which causes the water to flow slower over the obstacle thus creating a higher wave upstream.

In Figure D.16 the result of two pressure probes can be observed. The first probe is located at the front face of the obstacle, whereas the second one is located on top of it. Probe P2 indicates that the pressure peak is not appropriately resolved indicating that either space or time discretisation is not fine enough. In comparison with the VOF simulation it is expected that the



(a) Probe H4 (b) Probe H2

Figure D.14: Water levels.

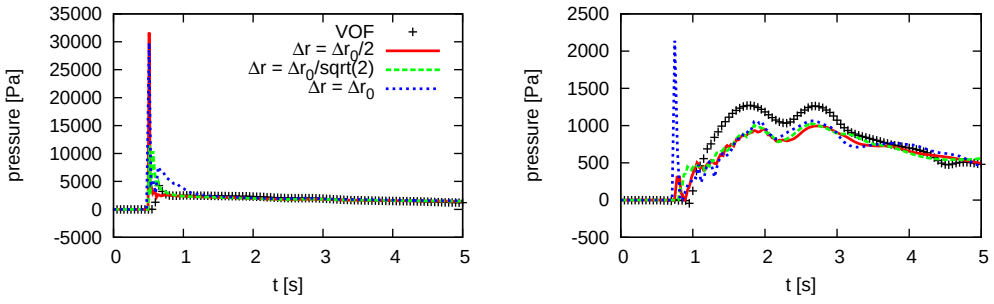


(a) $t = 0.513s$ (b) $t = 0.66s$

(c) $t = 1.76s$ (d) $t = 4.4s$

Figure D.15: Snapshots of the simulation coloured according to velocity magnitude.

peak is appearing earlier and higher due to the absence of air. After the impact event the pressure is slightly lower when compared to the reference simulation which is once again explained by the difference in height of the wave reflected by the obstacle. This can also be observed in the right-hand side of Figure D.16 where the pressure at Probe P5 is shown. Due to the lower water level the pressure is reduced but both reflection events are captured.



(a) Probe P2 (b) Probe P5

Figure D.16: Pressure gauges.

As the goal of this section is to demonstrate the extension of the semi-analytical wall boundary condition to 3-D it shall be mentioned that there were no particles penetrating the walls thanks to the accurate imposition of the boundary terms. If again the approximation shown in Equation (D.28) is used, the simulation behaves similar to the one presented in the previous chapter, there are some particles which penetrate the wall. Finally, the computation with the approximative value of $\nabla\gamma_{as}$ requires 0.8 s per iteration, whereas the simulation with the analytical value requires 0.97 s per iteration. This means that the simulation is about 15% slower, an acceptable

value considering the improved behaviour of the simulation.

D.5 Extension of wall boundary condition to axisymmetric formulation for SPH

This Section is to propose an extension of boundary terms in an axisymmetric framework. This work has been performed in 2016 during the internship of Zixing Dong, a former student of École des Ponts ParisTech in collaboration with Christophe Peniguel, EDF R&D, MFEE.

The motivation of this extension is to study the dynamic of falling cylindrical objects in confined domains.

D.5.1 Recall of SPH interpolations and nomenclature

For a fluid flow represented by a domain Ω, the value of an arbitrary physical property A located at position denoted in this Section[2] by $\boldsymbol{s}_a$ at time t can be expressed by using the Dirac distribution as:

$$A(\boldsymbol{s}_a, t) = \int_\Omega A(\boldsymbol{s}', t)\delta(\boldsymbol{s}_a - \boldsymbol{s}')d\boldsymbol{s}'. \tag{D.29}$$

If the Dirac distribution is replaced by a continuous function with a compact support Ω_a centred on $\boldsymbol{s}_a$ in space, called interpolation kernel in SPH, denoted as $w_h(\boldsymbol{s}_a - \boldsymbol{s}')$, $A(\boldsymbol{s}_a)$ is approximated by its continuous interpolation:

$$A(\boldsymbol{s}_a, t) \approx [A]^c(\boldsymbol{s}_a, t) = \int_{\Omega\cap\Omega_a} A(\boldsymbol{s}', t)w_h(\boldsymbol{s}_a - \boldsymbol{s}')d\boldsymbol{s}'. \tag{D.30}$$

In SPH, Ω is discretised with a set of fluid particles $\mathcal{P}$ and any particle b has its own mass M_b, density ρ_b and other physical properties. The continuous interpolation above is then discretised over particles, which leads to the smoothed discretised interpolation:

$$[A]^c(\boldsymbol{s}, t) \approx [A]_a^d = \sum_{b\in\mathcal{P}} \frac{M_b}{\rho_b} A_b w_{ab}, \tag{D.31}$$

where $w_{ab} = w_h(\boldsymbol{s}_a - \boldsymbol{s}_b)$.

In the Unified Semi-Analytical Wall (USAW) boundary conditions framework proposed in Ferrand et al. (2013a), boundary is decomposed in a set of surface elements which are segments in 2-D. The vertices of segments' ends are truncated fluid particles, called vertex particles. Vertex particle's volume is a function of a ratio factor, noted θ, which is the angle between the boundary segments that it belongs to, divided by 2π. For fluid particles, this ratio is set to be 1.

$$V_v = \theta_v V_0. \tag{D.32}$$

Here are a list of notations used in this Section:

Ω: domain of the fluid flow to simulate;

$\mathcal{P}$: set of fluid and vertex particles;

$\mathcal{F}$: set of fluid particles;

$\mathcal{V}$: set of vertex particles;

[2] The change of notation is made here to make a clear distinction between the position vector $\boldsymbol{s}$ and the radial component r.

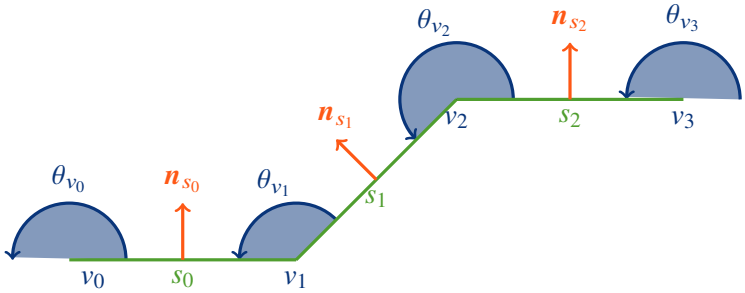


Figure D.17: Segments and their vertex particles.

$\mathcal{S}$: set of boundary surfaces which are segments in 2-D and triangular surfaces in 3-D;

s: an arbitrary element of $\mathcal{S}$;

$\boldsymbol{n}_s$: inward normal to the segment s;

Δs: mean distance between particles;

ρ_0: reference density of fluid particles, for water in standard condition of pressure and temperature, $\rho_0 = 10^3 kg/m^3$;

$\widehat{V}_0$: reference pseudo-volume of fluid particles, in 2-D Cartesian SPH and 2-D axisymmetric SPH it is a surface $\widehat{V}_0 = S_0 = \Delta s^2$;

M_a: mass associated to particle a;

$\widehat{M}_a$: pseudo-mass associated to particle a: $\widehat{M}_a = \frac{M_a}{2\pi r_a}$;

A_a: physical property A of particle a, for example, ρ_a is the density of a, p_a its pressure, $\boldsymbol{u}_a$ its velocity, V_a its the volume...

$\boldsymbol{s}_a$: position of a, in 2-D Cartesian SPH, $\boldsymbol{s}_a = (x_a, z_a)$ and in 2-D axisymmetric SPH, $\boldsymbol{s}_a = (r_a, z_a)$;

$\boldsymbol{s}_{ab} = \boldsymbol{s}_a - \boldsymbol{s}_b$

$s_{a\sigma}$: projected distance from particle a and the nearest boundary;

h: smoothing length, characterizing the dependence of w_h with respect to $\boldsymbol{s}_a - \boldsymbol{s}$;

$w_h(\boldsymbol{s}_a - \boldsymbol{s})$: (2-D) kernel interpolation centred on $\boldsymbol{s}_a$;

$w(\boldsymbol{s}_a - \boldsymbol{s})$: $w_h(\boldsymbol{s}_a - \boldsymbol{s})$ when h is a constant;

Ω_a: (compact) support of $w_h(\boldsymbol{s}_a - \boldsymbol{s})$;

w_{ab}: $w_{ab} = w_h(\boldsymbol{s}_a - \boldsymbol{s}_b)$;

$q = \frac{r}{h}$: dimensionless distance used in kernel interpolation;

Λ: dimensionless weighting coefficient of volume diffusion term.

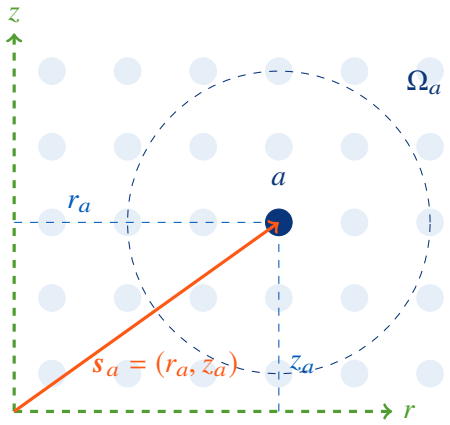


Figure D.18: Fluid domain and particles' positions in 2-D axisymmetric SPH.

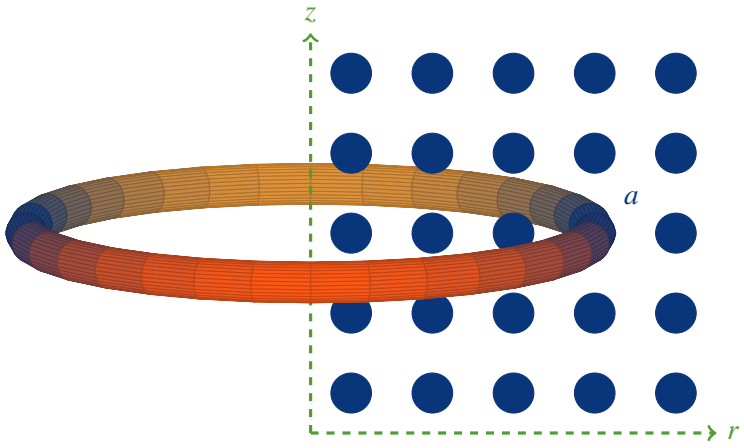


Figure D.19: A particle in the plane (r, z) represents a donut of matter in 3-D space.

D.5.2 Introduction of axisymmetric SPH and literature review

2-D axisymmetric SPH aims at solving in cylindrical coordinates problems independent of the rolling angle θ.

The 3-D space, denoted by Ω^{3D} can therefore be built by extruding by rotation of θ from 0 to 2π around the symmetric axis (denoted by z here) of a 2-D domain Ω^{axi}. Thus, each particle represents a donut of the flow in 3-D space (see Figure D.5.2). The 2-D studied domain can therefore be reduced to Ω^{axi} which is included in the half-plane $(r \geq 0, z)$. It can also be convenient to define the symmetric domain of Ω^{axi}, denoted by $(\Omega^{axi})^s$, with respect to the symmetric axis. Then, the 2-D domain is defined by $\Omega^{2D} = \Omega^{axi} \cup (\Omega^{axi})^s$ and will be denoted by Ω in the following.

Some works have already been done for 2-D axisymmetric SPH. Some pioneer work was performed by Herant and Benz (1992), and gave the definition η, density divided by the radial component, and their formulation of axisymmetric SPH is similar to the 2-D Cartesian SPH:

$$\begin{aligned} \eta_a &:= \sum_{b\in\mathcal{P}} M_b w_{ab}, \\ \rho_a &= \frac{\eta_a}{2\pi r_a}, \\ \frac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t} &= -\sum_{b\in\mathcal{P}} \frac{M_b}{2\pi r_b}\left[\frac{p_b}{\rho_b^2} + \frac{p_a}{\rho_a^2} + \Pi_{ab}\right]\nabla w_{ab}, \end{aligned} \tag{D.33}$$

where particle's mass is defined at the beginning of simulation as a function of the distance to the symmetric axis: $M = 2\pi r \rho_0 S$ and remains a constant in time, and Π is the artificial viscosity to stabilise the system.

Brookshaw (2003) improves this formulation by using Lagrangian mechanisms. For an isentropic compressible flow, the Lagrangian reads

$$L = \sum_{b\in\mathcal{P}} (E_{ke,b} - E_{int,b}), \tag{D.34}$$

where $E_{ke,b} = \frac{1}{2} M_b \boldsymbol{u}_b^2$ and $\mathrm{d}E_{int,b} = M_b \frac{p_b}{\rho_b^2}\mathrm{d}\rho_b$. As the Lagrange equations are

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \boldsymbol{u}_a}\right) - \frac{\partial L}{\partial \boldsymbol{s}_a} = \boldsymbol{0}, \tag{D.35}$$

using the definition of ρ given in (D.33), the hoop-stress term $2\pi\frac{p_a}{\eta_a}\boldsymbol{e}_r$ appears naturally and the Lagrange equations become:

$$\frac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t} = -2\pi \sum_{b\in\mathcal{P}} M_b \left(\frac{p_a r_a}{\eta_a^2} + \frac{p_b r_b}{\eta_b^2}\right)\nabla w_{ab} + 2\pi\frac{p_a}{\eta_a}\boldsymbol{e}_r. \tag{D.36}$$

As mentioned in the conclusion of Brookshaw (2003), Brookshaw's formulation is not suitable for particles near the symmetric axis when $r < 2h$. In the thesis of Relaño Castillo (2012), a renormalisation factor $f_1(r)$ is introduced to fix problems in the axis neighbourhood. Theoretically, η should be zero at symmetric axis but if $\eta_a = \sum_{b\in\mathcal{P}} M_b w_{ab}$ is used, this property can not be fulfilled. Forcing η to tend towards zero when particle moves towards the symmetric axis gives rise to a renormalisation factor $f_1(r)$.

If the Gaussian kernel is used, as it reads

$$\begin{aligned} w_{2D}^G\left(\frac{|\boldsymbol{s}-\boldsymbol{s}'|}{h}\right) &= K_{2D}\exp\left(\frac{-|r-r'|^2}{h^2}\right)\exp\left(\frac{-|z-z'|^2}{h^2}\right) \\ &= K_{1D}^2 w_{1D}^G\left(\frac{|r-r'|}{h}\right) w_{1D}^G\left(\frac{|z-z'|}{h}\right). \end{aligned} \tag{D.37}$$

$f_1(r)$ can be expressed as

$$\frac{1}{f_1(r)} = \frac{K_{1D}}{|r|}\int_{-\infty}^{+\infty} |r'| w_{1D}^G\left(\frac{|r-r'|}{h}\right) dr'. \tag{D.38}$$

The renormalised formulation reads:

$$\begin{aligned}
\widehat{\eta}_a &= \eta_a f_1(r_a),\\
A_a &= \sum_{b\in\mathcal{P}} \frac{M_b}{\widehat{\eta}_b} A_b w_{ab},\\
\frac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t} &= 2\pi\frac{p_a}{\widehat{\eta}_a}\boldsymbol{e}_r - 2\pi\sum_{b\in\mathcal{P}}\left[M_b\left(\frac{p_a r_a}{\widehat{\eta}_a^2} f_1(r_a) + \frac{p_b r_b}{\widehat{\eta}_b^2} f_1(r_b)\right)\nabla w_{ab}\right]\\
&\quad - \left(\frac{2\pi p_a r_a}{\widehat{\eta}_a f_1(r_a)}\right)\frac{\mathrm{d}f_1(r_a)}{\mathrm{d}r_a}.
\end{aligned} \tag{D.39}$$

In this document, f_1 has not been used but another renormalisation factor is introduced to treat at the same time boundary renormalisation presented in Chapter 2 and axis renormalisation.

The discrete interpolation in axisymmetric SPH used in Brookshaw (2003) and in Relaño Castillo (2012) to get smooth interpolation even if particle mass is heterogeneous reads:

$$[A]_a^d = \sum_{b\in\mathcal{P}} \frac{M_b}{\eta_b} A_b w_{ab}. \tag{D.40}$$

This proposition has the following characteristics:

- Mass of particles is defined at the beginning of the simulation and is conserved in time:

  $$M_a = 2\pi r_a(t=0)\rho_0 S_0. \tag{D.41}$$

  Therefore, when a particle moves towards the symmetric axis, its density increases and it repulses particles around;

- Smoothing length is taken variable, and depends on the surrounding particles to ensure that each particle has a constant number of neighbour particles.

More recently, Li et al. (2020) proposed an axisymmetric formulation for multiphase flows, however they get some spurious acceleration next to the axis for the hydrostatic test. They also introduced radial shifting when particles are next to the axis and an artificial damping term. Moreover they used mirror particles to deal with solid walls. Sun et al. (2021) proposed a splitting–merging techniques to deal with the singularity of the axis.

In the next sub-sections, we want to keep the smoothing length constant over time and space, which is the choice we made for 2-D Cartesian WCSPH. For this, a possible solution is to conserve the surfacial mass (mass of the donut by unit of ring length) of particles in time and in space. This implies that the mass of each fluid particle varies with its position in radial direction.

A constant mass by unit of ring length, $\widehat{M}$, is defined as $\widehat{M} := \frac{M_a}{2\pi\theta_a r_a} = \rho_a S_a$, and then $\widehat{M}_a$ is defined as a new field attached to each particle such that $\widehat{M}_a := \widehat{M}\theta_a$.

As for SPH interpolations used in multi-fluid simulations (see e.g. Ghaïtanellis et al., 2018), the proposed discrete interpolation for axisymmetric SPH, reads:

$$[A]_a^d = \frac{M_a}{2\pi\theta_a r_a}\sum_{b\in\mathcal{P}}\frac{A_b}{\rho_b}\theta_b w_{ab} = \sum_{b\in\mathcal{P}}\widehat{M}_b\frac{A_b}{\rho_b} w_{ab}. \tag{D.42}$$

In the next Section D.5.3, we use definition (D.42) to discretise mass equation. Then discrete divergence operator is derived, and, from Lagrangian mechanics, an attempt is made to get discrete gradient operator which is the divergence skew-adjoint operator. Unfortunately, this gradient operator is not accurate, especially gradient of constants are non-zero next to the axis. Thus, as in Chapter 2, we step back to continuous interpolation to derive a gradient operator which make boundary terms appear naturally. A renormalisation factor is introduced which allows one to treat at the same time axis terms and boundary terms.

D.5.3 Present axisymmetric formulation based on mirroring particles and a unified renormalisation factor

In 3-D axisymmetric space, a particle is in fact a donut generated by the rotation around the symmetric axis and a position $\boldsymbol{s}_a$ stands for a ring.

Specular particles next to the axis

When $0 \leq r_a < 2h$, the donut generated by the rotation around the axis of Ω_a is composed of a volume (generated by negative $r' \in \Omega_a$) which should be counted twice.

There are two possibilities to renormalise: either we define a renormalisation factor integrating only on $\Omega_a \cap \Omega^{axi}$, or we create mirroring particles on $(\Omega^{axi})^s$ and we integrate on Ω as proposed by (Relaño Castillo, 2012). Here we prefer to introduce specular particles on the other side of the symmetric axis, because this makes easier to satisfy the symmetry condition at the discrete level.

R The relations between the specular particles and the original particles are:

$$\begin{aligned} r^s &= -r, \\ z^s &= z, \\ M^s &= M, \\ v_r^s &= -v_r, \\ v_z^s &= v_z. \end{aligned} \tag{D.43}$$

Each particle located at (r, z) gives birth to a symmetric particle of the same mass at $(-r, z)$ (see Figure D.20). Note that this means that the radial parameter r can be negative due to the symmetry.

Unified renormalisation factor and mass equation in axisymmetric SPH

Thus the continuous interpolation of density can be written as:

$$\rho_a = \frac{\displaystyle\int_{\Omega_a \cap \Omega} \rho(\boldsymbol{s}) w(\boldsymbol{s} - \boldsymbol{s}_a) |r| \mathrm{d}r\mathrm{d}z}{\displaystyle\int_{\Omega_a \cap \Omega} w(\boldsymbol{s} - \boldsymbol{s}_a) |r| \mathrm{d}r\mathrm{d}z}. \tag{D.44}$$

And the integral renormalisation factor, depending on particles' position, reads:

$$R_a := \int_{\Omega_a \cap \Omega} w(\boldsymbol{s} - \boldsymbol{s}_a) |r| \mathrm{d}r\mathrm{d}z. \tag{D.45}$$

It is a factor that combines the influence of the boundary and that of the symmetric axis. For particle a far from the boundary and the symmetric axis, $R_a = |r_a|$. However, when particle a is

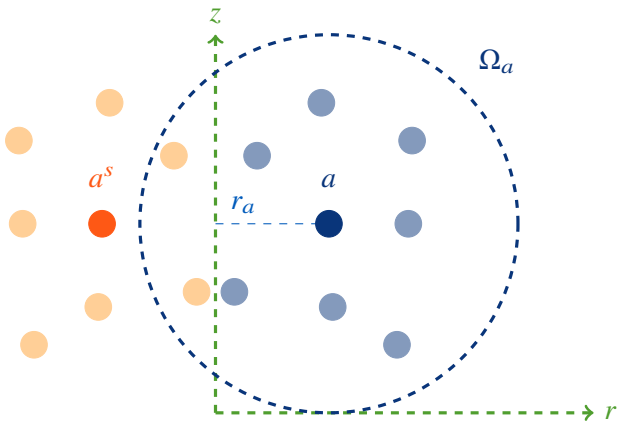


Figure D.20: Specular particles (orange) and original particles (blue).

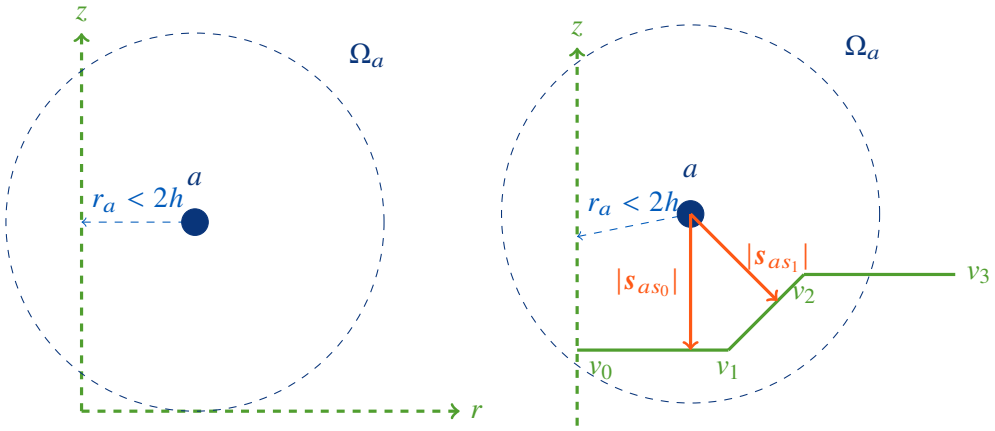


Figure D.21: Examples where $R_a \neq |r_a|$.

near the symmetric axis ($r_a < 2h$) or near the boundary ($s_{a\sigma} < 2h$), the support of its interpolation kernel is no more a donut in 3-D space, the replacement of R_a by $|r_a|$ will underestimate the value of volume density. In fact, R_a is an extension of Kulasegaram's renormalisation factor γ Kulasegaram et al. (2004).

It can be deduced that the discrete interpolation of density variation reads

$$\boxed{\mathrm{d}\left(R_a \rho_a\right) = \mathrm{d}\left(\sum_{b \in \mathcal{P}} \widehat{M}_b |r_b| w_{ab}\right).} \tag{D.46}$$

An attempt to derive skew-adjoint operators from Lagrange equations

Mass equation is already discretised in (D.46). Note that this equation already take walls into account and has no singularity at the axis because $R(r = 0) > 0$. Equation (D.46) can be used to formally define a discrete divergence operator. The choice made here to keep $\widehat{M}_a$ constant

implies that mass is not conserved for a particle moving in the radial direction:

$$\begin{aligned} 2\pi\widehat{M}_a\frac{\mathrm{d}|r_a|}{\mathrm{d}t} = \frac{\mathrm{d}M_a}{dt} &= \frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_a^{3D}}\rho\mathrm{d}V \\ &\simeq V_a^{3D}\frac{\mathrm{d}\rho_a}{\mathrm{d}t} + \rho_a\frac{\mathrm{d}V_a^{3D}}{\mathrm{d}t} \\ &= 2\pi|r_a|\frac{\widehat{M}_a}{\rho_a}\left(\frac{\mathrm{d}\rho_a}{\mathrm{d}t} + \rho_a\nabla_a\cdot\boldsymbol{v}\right). \end{aligned} \tag{D.47}$$

It should be noticed that $\boldsymbol{v}$ is not exactly equal to the fluid velocity, but is the velocity of the moving control volume V_a^{3D}, its centre is moved with the fluid velocity $\boldsymbol{u}$ under the constraint that $\widehat{M}_a$ is kept constant. Note also here that for all particles a, we consider that $\boldsymbol{v}_a = \boldsymbol{u}_a$, which means that they are Lagrangian points moving with the fluid velocity. So with a little abuse of notation, we replace $\nabla_a\cdot\boldsymbol{v}$ by $\nabla_a\cdot\boldsymbol{u}$ in the following.

The particle mass equation (D.47) rewrites:

$$\frac{\mathrm{d}|r_a|}{\mathrm{d}t} = \frac{|r_a|}{\rho_a}\frac{\mathrm{d}\rho_a}{\mathrm{d}t} + |r_a|\nabla_a\cdot\boldsymbol{u}, \tag{D.48}$$

which is used to deduce the divergence of velocity:

$$\nabla_a\cdot\boldsymbol{u} = \frac{1}{|r_a|}\frac{\mathrm{d}|r_a|}{dt} + \frac{1}{R_a}\frac{\mathrm{d}R_a}{\mathrm{d}t} - \frac{1}{R_a\rho_a}\frac{\mathrm{d}(R_a\rho_a)}{\mathrm{d}t}, \tag{D.49}$$

but, using (D.46) we get:

$$\begin{aligned} \frac{\mathrm{d}(R_a\rho_a)}{\mathrm{d}t} &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\sum_{b\in\mathcal{P}}\widehat{M}_b|r_b|w_{ab}\right) \\ &= \sum_{b\in\mathcal{P}}\widehat{M}_b\left(w_{ab}\boldsymbol{u}_b\cdot\nabla|r_b| + |r_b|\boldsymbol{u}_{ab}\cdot\nabla w_{ab}\right), \end{aligned} \tag{D.50}$$

Finally, we can define the discrete divergence operator in axisymmetric SPH:

$$\begin{aligned} \mathrm{Div}_a^{L^-}\{\boldsymbol{u}_b\} \coloneqq &-\frac{1}{R_a\rho_a}\sum_{b\in\mathcal{P}}\widehat{M}_b\left(|r_b|\boldsymbol{u}_{ab}\cdot\nabla w_{ab} + w_{ab}\boldsymbol{u}_b\cdot\nabla|r_b|\right) \\ &+\frac{1}{|r_a|}\boldsymbol{u}_a\cdot\nabla|r_a| + \frac{1}{R_a}\boldsymbol{u}_a\cdot\nabla R_a. \end{aligned} \tag{D.51}$$

Note that this operator is not used in the time marching, (D.50) is integrated directly.

We will now derive the discrete pressure gradient from Lagrangian mechanics. Lagrange equations (D.35) for variable mass rewrites (see e.g. Pesce et al., 2006) here:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial\boldsymbol{u}_a} - \frac{\partial L}{\partial\boldsymbol{s}_a} = \sum_{b\in\mathcal{P}}\dot{M}_b\boldsymbol{u}_b^{in}\cdot\frac{\partial\boldsymbol{s}_b}{\partial\boldsymbol{s}_a} + \sum_{b\in\mathcal{P}}\left[\frac{1}{2}\frac{d}{dt}\left(\frac{\partial M_b}{\partial\boldsymbol{u}_a}\boldsymbol{u}_b^2\right) - \frac{1}{2}\frac{\partial M_b}{\partial\boldsymbol{s}_a}\boldsymbol{u}_b^2\right], \tag{D.52}$$

where $\boldsymbol{u}^{in}$ represents the velocity of the expelled or gained mass. It is set to zero as the system is assumed to be closed here. Moreover, particle mass does not vary with particle velocity $\frac{\partial M_b}{\partial\boldsymbol{u}_a} = \boldsymbol{0}$. For an isentropic flow, the Lagrangian can be written as

$$L = \sum_{b\in\mathcal{P}}E_{ke,b} - \sum_{b\in\mathcal{P}}E_{int,b}, \tag{D.53}$$

where $E_{ke,b}$ and $E_{int,b}$ are respectively particle b's kinetic energy and internal energy and vary as:

$$\begin{aligned} \mathrm{d}E_{ke,b} &= \mathrm{d}\left(\frac{1}{2}M_b\boldsymbol{u}_b^2\right) = \tfrac{1}{2}\boldsymbol{u}_b^2\mathrm{d}M_b + M_b\boldsymbol{u}_b\cdot\mathrm{d}\boldsymbol{u}_b, \\ \mathrm{d}E_{int,b} &= M_b\frac{p_b}{\rho_b^2}d\rho_b - \frac{p_b}{\rho_b}\mathrm{d}M_b. \end{aligned} \tag{D.54}$$

Lagrange equations (D.52) can be split into two parts (kinematic and dynamic parts):

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \boldsymbol{u}_a} = 2\pi\widehat{M}_a\frac{\mathrm{d}}{\mathrm{d}t}\left(|r_a|\boldsymbol{u}_a\right), \tag{D.55}$$

and

$$\frac{\partial L}{\partial \boldsymbol{s}_a} + \sum_{b\in\mathcal{P}}\left[\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial M_b}{\partial \boldsymbol{u}_a}\boldsymbol{u}_b^2\right) - \frac{1}{2}\frac{\partial M_b}{\partial \boldsymbol{s}_a}\boldsymbol{u}_b^2\right] = -\sum_{b\in\mathcal{P}}\frac{\partial E_{int,b}}{\partial \boldsymbol{s}_a}. \tag{D.56}$$

(D.56) can be developed as

$$\begin{aligned} \sum_{b\in\mathcal{P}}\frac{\partial E_{int,b}}{\partial \boldsymbol{s}_a} &= \sum_{b\in\mathcal{P}} M_b\frac{p_b}{\rho_b^2}\frac{\partial\rho_b}{\partial \boldsymbol{s}_a} - \sum_{b\in\mathcal{P}}\frac{p_b}{\rho_b}\frac{\partial M_b}{\partial \boldsymbol{s}_a} \\ &= \sum_{b\in\mathcal{P}} 2\pi\widehat{M}_b\frac{p_b}{\rho_b^2}\frac{|r_b|}{R_b}\frac{\partial(R_b\rho_b)}{\partial \boldsymbol{s}_a} - 2\pi\widehat{M}_a\frac{p_a}{\rho_a}\frac{|r_a|}{R_a}\frac{\partial R_a}{\partial \boldsymbol{s}_a} \\ &\quad -2\pi\widehat{M}_a\frac{p_a}{\rho_a}\nabla|r_a|, \end{aligned} \tag{D.57}$$

where, using again (D.46), we have

$$\mathrm{d}\left(R_b\rho_b\right) = \mathrm{d}\left(\sum_{c\in\mathcal{P}}\widehat{M}_c|r_c|w_{bc}\right). \tag{D.58}$$

Combining Equations (D.55), (D.56) and (D.58), the expression of the equation of movement reads

$$\begin{aligned} \frac{\mathrm{d}(|r_a|\boldsymbol{u}_a)}{\mathrm{d}t} &= -\sum_{b\in\mathcal{P}}\widehat{M}_b\left(\frac{p_a}{\rho_a^2}\frac{|r_a|}{R_a}|r_b| + \frac{p_b}{\rho_b^2}\frac{|r_b|}{R_b}|r_a|\right)\nabla w_{ab} \\ &\quad +\frac{p_a}{\rho_a}\frac{|r_a|}{R_a}\nabla R_a + \frac{p_a}{\rho_a}\nabla|r_a| \\ &\quad -\sum_{b\in\mathcal{P}}\widehat{M}_b\frac{p_b}{\rho_b^2}\frac{|r_b|}{R_b}w_{ab}\nabla|r_a|. \end{aligned} \tag{D.59}$$

This expression can be used to define a discrete gradient operator of an arbitrary field A in 2-D axisymmetric SPH as:

$$\begin{aligned} \mathbf{Grad}_a^{L^+}\{A_b\} &:= \frac{\rho_a}{|r_a|}\sum_{b\in\mathcal{P}}\widehat{M}_b\left(\frac{A_a}{\rho_a^2}\frac{|r_a|}{R_a}|r_b| + \frac{A_b}{\rho_b^2}\frac{|r_b|}{R_b}|r_a|\right)\nabla w_{ab} \\ &\quad -\frac{A_a}{R_a}\nabla R_a - \frac{A_a}{|r_a|}\nabla|r_a| \\ &\quad +\frac{\rho_a}{|r_a|}\sum_{b\in\mathcal{P}}\widehat{M}_b\frac{A_b}{\rho_b^2}\frac{|r_b|}{R_b}w_{ab}\nabla|r_a|, \end{aligned} \tag{D.60}$$

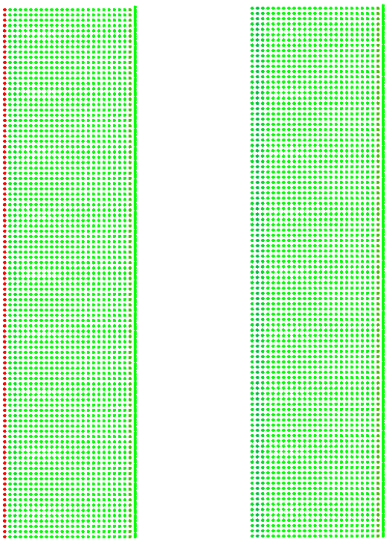


Figure D.22: Comparison of gradient operators in the radial direction $\frac{\partial p}{\partial r}$, with the last two terms in (D.60) (left) and without the last two terms in (D.60) (right) in a cylinder for a constant pressure $p = 28\,000\,\text{Pa}$. Reference density $\rho_0 = 1000\,\text{kg}\,\text{m}^{-3}$ and $c_0 = 20\,\text{m}\,\text{s}^{-1}$.

and the momentum equation now writes $\frac{\mathrm{d}(|r_a|\boldsymbol{u}_a)}{\mathrm{d}t} = -\frac{|r_a|}{\rho_a}\mathbf{Grad}_a^{L^+}\{p_b\}$. The exponent $^+$ means that, considering only terms summed over neighbouring particles b, the pressure force can be written as $\frac{M_a}{\rho_a}\mathbf{Grad}_a^{L^+}\{A_b\} = \sum_{b\in\mathcal{P}} \boldsymbol{F}^p_{a\to b}$ + other terms, with $\boldsymbol{F}^p_{a\to b} = -\boldsymbol{F}^p_{b\to a}$, and thus (part of) the pressure gradient verifies the action-reaction principle.

Using the definition made in equations (D.51) and (D.60), the skew-adjointness property $(A_a, \mathrm{Div}_a^{L^-}\{\boldsymbol{B}_b\}) = -<\mathbf{Grad}_a^{L^+}\{A_b\}, \boldsymbol{B}_a>$ is well-verified.

However, note that Equation (D.60) is not accurate for particles near the symmetry axis, because the last two terms lead up to a large radial pressure force for a constant pressure field and make particles move radially. In fact, if the last two terms are removed, this problem does not exist anymore (see Figure D.22).

Derivation of the discrete gradient operator from a continuous smoothed interpolation

As expressed in the Section above, the pressure gradient is not accurate near the axisymmetric axis. To circumvent this problem we are stepping back to the continuous interpolation.

Another issue is that, similar to the problems in 2-D Cartesian SPH, the Lagrange equations (D.52) are no more fulfilled for the particles next to the boundary: the force exerted by the boundary on the fluid flow is not taken into account. In the following part of this Section, we use the same methodology as in Chapter 2 and De Leffe et al. (2009): the discrete interpolation which combines the boundary and symmetric term is built from the continuous interpolation. Then, the derivation product rule in continuous interpolations is used to construct the discrete interpolation of $\nabla|r_a|$.

If we go back to the continuous interpolation for an arbitrary field A

$$[A]_a^{axi,c} = \frac{1}{R_a}\int_{\Omega\cap\Omega_a} A(\boldsymbol{s})w(\boldsymbol{s}-\boldsymbol{s}_a)|r|\mathrm{d}r\mathrm{d}z, \tag{D.61}$$

and corresponding discrete interpolation is now defined as:

$$[A]_a^{axi,d} = \frac{1}{R_a}\sum_{b\in\mathcal{P}} \frac{\widehat{M}_b}{\rho_b} A_b|r_b|w_{ab}. \tag{D.62}$$

Applying the continuous interpolation (D.61) to a field gradient gives:

$$\begin{aligned}[\nabla A]_a^{axi,c} = &\frac{1}{R_a}\int_{\Omega\cap\Omega_a} A(\boldsymbol{s})\nabla_a w(\boldsymbol{s}-\boldsymbol{s}_a)|r|\mathrm{d}r\mathrm{d}z - \frac{1}{R_a}\int_{\partial(\Omega\cap\Omega_a)} |r|A(\boldsymbol{s})w(\boldsymbol{s}-\boldsymbol{s}_a)\boldsymbol{n}dl \\ &-\frac{1}{R_a}\int_{\Omega\cap\Omega_a} A(\boldsymbol{s})w(\boldsymbol{s}-\boldsymbol{s}_a)\mathrm{d}r\mathrm{d}z\,\mathtt{sign}(r)\boldsymbol{e}_r.\end{aligned} \tag{D.63}$$

The corresponding discrete interpolation is

$$\begin{aligned}[\nabla A]_a^{axi,d} = &\frac{1}{R_a}\sum_{b\in\mathcal{P}}\frac{\widehat{M}_b}{\rho_b}A_b|r_b|\nabla w_{ab} - \frac{1}{R_a}\sum_{s\in\mathcal{S}}A_s\nabla R_{as} \\ &-\frac{1}{R_a}\sum_{b\in\mathcal{P}}\mathtt{sign}(r_b)\frac{\widehat{M}_b}{\rho_b}A_b w_{ab}\boldsymbol{e}_r,\end{aligned} \tag{D.64}$$

where ∇R_{as} is defined by

$$\nabla R_{as} \coloneqq \int_s |r|w(\boldsymbol{s}-\boldsymbol{s}_a)dl\boldsymbol{n}_s. \tag{D.65}$$

(D.64) applied to the pressure field does not fulfill action/reaction principle between two neighbouring particles.

For particles far from the boundary and the symmetric axis, if we go back a while to the operator derived from the Lagrangian, (D.60) becomes

$$\begin{aligned}\mathbf{Grad}_a^{L^+}\{A_b\} = &\frac{\rho_a}{|r_a|}\sum_{b\in\mathcal{P}}\widehat{M}_b\left(\frac{A_a}{\rho_a^2}|r_b| + \frac{A_b}{\rho_b^2}|r_a|\right)\nabla w_{ab} \\ &-2\frac{A_a}{|r_a|}\nabla|r_a| + \frac{\rho_a}{|r_a|}\sum_{b\in\mathcal{P}}\widehat{M}_b\frac{A_b}{\rho_b^2}w_{ab}\nabla|r_a|,\end{aligned} \tag{D.66}$$

and can therefore be written as:

$$\begin{aligned}\mathbf{Grad}_a^{L^+}\{A_b\} = &\frac{A_a}{\rho_a}[\nabla\rho]_a^{axi,d} + \rho_a|r_a|\left[\nabla\left(\frac{A}{\rho|r|}\right)\right]_a^{axi,d} \\ &-2\frac{A_a}{|r_a|}\nabla|r_a| + \rho_a\left[\frac{A}{\rho|r|}\right]_a^{axi,d}\nabla|r_a|.\end{aligned} \tag{D.67}$$

This decomposition is extended for all particles (including particles near the boundary and the symmetric axis). To remove singularities, we propose to replace $\frac{\nabla|r_a|}{|r_a|}$ by its discrete interpolation $[\frac{\nabla|r|}{|r|}]_a^{axi,d}$ and $\rho_a[\frac{A}{\rho|r|}]_a^{axi,d}\nabla|r_a|$ by $A_a[\frac{\nabla|r|}{|r|}]_a^{axi,d}$, using definitions (D.62) and (D.64), equation (D.66) is replaced by:

$$\boxed{\begin{aligned}\mathbf{Grad}_a^{R^+}\{A_b\} = &\frac{\rho_a}{R_a}\sum_{b\in\mathcal{P}}\widehat{M}_b\left(\frac{A_a}{\rho_a^2}|r_b| + \frac{A_b}{\rho_b^2}|r_a|\right)\nabla w_{ab} \\ &-\frac{A_a}{R_a}\sum_{b\in\mathcal{P}}\mathtt{sign}(r_b)\frac{\widehat{M}_b}{\rho_b}w_{ab}\boldsymbol{e}_r \\ &-\frac{\rho_a}{R_a}\sum_{s\in\mathcal{S}}\rho_s\left(\frac{A_a}{\rho_a^2}\nabla R_{as} + \frac{A_s}{\rho_s^2}|r_a|\nabla\gamma_{as}^{2D}\right).\end{aligned}} \tag{D.68}$$

Some comments can be made on the proposed gradient (D.68) compared to (D.60) and the existing literature Brookshaw (2003); Relaño Castillo (2012); Li et al. (2020). First, (D.68) can

be applied to a particle going on the symmetry axis without singularity, and, thanks to specular particles, it is obvious that the gradient vanishes imposing at the discrete level $\frac{\partial}{\partial r} = 0$. Second, boundaries are treated with a unique renormalisation factor, without using additional correction factor.

Laplacian operator in axisymmetric SPH

Morris et al. (1997) proposed that Laplacian can be written as the divergence of a gradient where the divergence is in its SPH form and the gradient is expressed by a finite difference approximation. Following this idea, (Leroy, 2014) proposed that the SPH form of divergence is antisymmetric to ensure the action-reaction principle. The antisymmetric form derived from the idea of (Monaghan, 1992) in axisymmetric framework reads:

$$\begin{aligned}\mathrm{Div}_a^{R^+}\left\{\boldsymbol{A}_b\right\} = &\frac{1}{R_a}\sum_{b\in\mathcal{P}}\frac{\widehat{M}_b}{\rho_b}\left(|r_b|\boldsymbol{A}_a + |r_a|\boldsymbol{A}_b\right)\cdot\nabla w_{ab}\\ &-\frac{1}{R_a}\sum_{s\in\mathcal{S}}\left(\boldsymbol{A}_a\cdot\nabla R_{as} + |r_a|\boldsymbol{A}_s\cdot\nabla\gamma_{as}^{2D}\right).\end{aligned} \tag{D.69}$$

As $\mathbf{Lap}_a^R\left(\{\mu_b\},\,\{\boldsymbol{u}_b\}\right) = \mathbf{Div}_a^{R^+}\left\{\mu_b\nabla\boldsymbol{u}_b\right\}$, after some algebra, Laplacian in 2-D axisymmetric is deduced as

$$\begin{aligned}\mathbf{Lap}_a^R\left(\{\mu_b\},\,\{\boldsymbol{u}_b\}\right) = &\frac{1}{R_a}\sum_{b\in\mathcal{P}}\frac{\widehat{M}_b}{\rho_b}\left(|r_b|\mu_a\nabla\boldsymbol{u}_a + |r_a|\mu_b\nabla\boldsymbol{u}_b\right)\cdot\nabla w_{ab}\\ &-\frac{1}{R_a}\sum_{s\in\mathcal{S}}\left(\mu_a\nabla\boldsymbol{u}_a\cdot\nabla R_{as} + |r_a|\mu_s\nabla\boldsymbol{u}_s\cdot\nabla\gamma_{as}^{2D}\right).\end{aligned} \tag{D.70}$$

In Equation (D.70), a two-point flux approximation and flux continuity is used to replace them by $\mu_a\nabla\boldsymbol{u}_a\cdot\boldsymbol{e}_{ab} \simeq \mu_b\nabla\boldsymbol{u}_b\cdot\boldsymbol{e}_{ab} \simeq \frac{2\mu_a\mu_b}{\mu_a+\mu_b}\frac{\boldsymbol{u}_{ab}\cdot\boldsymbol{e}_{ab}}{|\boldsymbol{s}_{ab}|}$ and next to the boundary $\mu_a\nabla\boldsymbol{u}_a\cdot\boldsymbol{n}_s \simeq \mu_s\nabla\boldsymbol{u}_s\cdot\boldsymbol{n}_s \simeq \mu_a\frac{\boldsymbol{u}_{as}\cdot\boldsymbol{n}_s}{\boldsymbol{s}_{as}\cdot\boldsymbol{n}_s}$.

Time stepping in 2-D axisymmetric SPH

The time stepping proposed in 2-D axisymmetric SPH is similar to that in 2-D Cartesian SPH used in Chapter 2:

$$\begin{aligned}\boldsymbol{F}_a^n &= -\frac{|r_a^n|}{\rho_a^n}\mathbf{Grad}_a^{R^+}\left\{p_b^n\right\} + \frac{|r_a^n|}{\rho_a^n}\mathbf{Lap}_a^R\left(\{\mu_b\},\,\left\{\boldsymbol{u}_b^n\right\}\right) + |r_a^n|\boldsymbol{g},\\ r_a^{n+1}\boldsymbol{u}_a^{n+1} &= r_a^n\boldsymbol{u}_a^n + \Delta t\boldsymbol{F}_a^n,\\ \boldsymbol{s}_a^{n+1} &= \boldsymbol{s}_a^n + \Delta t\boldsymbol{u}_a^{n+1},\\ \rho_a^{n+1} &= \frac{1}{R_a^{n+1}}\left[R_a^n\rho_a^n + \sum_{b\in\mathcal{P}}\widehat{M}_b(|r_b^{n+1}|w_{ab}^{n+1} - |r_b^n|w_{ab}^n)\right],\end{aligned} \tag{D.71}$$

where R_a, $\mathbf{Grad}_a^{R^+}\{p_b\}$ and $\mathbf{Lap}_a^R\left(\{\mu_b\},\,\{\boldsymbol{u}_b\}\right)$ are defined respectively by Equations (D.45), (D.68) and (D.70).

The main specificity is that the velocity $\boldsymbol{u}^{n+1}$ involves r^{n+1}, and integration of positions from time n to $n+1$ leads to a second degree polynomial equation in r^{n+1}:

$$\left(r_a^n + \Delta t\boldsymbol{u}_a^{n+1}\cdot\boldsymbol{e}_r\right)\boldsymbol{u}_a^{n+1} = r_a^n\boldsymbol{u}_a^n + \Delta t\boldsymbol{F}_a^n. \tag{D.72}$$

The positive root of this equation is:

$$\begin{aligned} r_a^{n+1} &= \frac{1}{2}\left(r_a^n + \sqrt{r_a^{n,2} + 4\Delta t r_a^n \boldsymbol{u}_a^n \cdot \boldsymbol{e}_r + 4\Delta t^2 \boldsymbol{F}_a^n \cdot \boldsymbol{e}_r}\right), \\ \boldsymbol{u}_a^{n+1} \cdot \boldsymbol{e}_r &= \frac{r_a^{n+1} - r_a^n}{\Delta t}, \\ \boldsymbol{u}_a^{n+1} \cdot \boldsymbol{e}_z &= \frac{r_a^n \boldsymbol{u}_a^n \cdot \boldsymbol{e}_z + \Delta t \boldsymbol{F}_a^n \cdot \boldsymbol{e}_z}{r_a^{n+1}}. \end{aligned} \tag{D.73}$$

Volume diffusion term

As in WCSPH, the equation of state is used to calculate p from ρ. Small errors on ρ can lead to a big error on p. A stabilisation method is required to smooth the field of pressure. The artificial viscosity is one of these methods widely used. Another method that is used in this Section is the addition of the volume diffusion term in the continuity equation. The original idea is proposed by Brezzi and Pitkäranta (1984) for finite elements and then adapted in Semi-Analytical SPH boundary conditions by Ghaitanellis et al. (2015).

First, we have the relationship between the divergence operator in cylindrical coordinates and in Cartesian coordinates: $r\mathrm{div}^{cyl}(\boldsymbol{v}) = \mathrm{div}^{cart}(r\boldsymbol{v})$. We use this trick to write $r_a \mathrm{Div}_a^{R^-}\{\boldsymbol{v}_b\} = \mathrm{Div}_a^-\{r_b \boldsymbol{v}_b\}$. Equation (D.48) could be time-discretised as follows:

$$\begin{aligned} |r_a^{n+1}|\frac{\rho_a^{n+1} - \rho_a^n}{\Delta t} &\simeq \rho_a^n \frac{|r_a^{n+1}| - |r_a^n|}{\Delta t} - \rho_a^n r_b^{n+1} \mathrm{Div}_a^{R^-}\left\{\boldsymbol{u}_b^{n+1}\right\} \\ &= \rho_a^n \frac{|r_a^{n+1}| - |r_a^n|}{\Delta t} - \rho_a^n \mathrm{Div}_a^-\left\{r_b^n \boldsymbol{u}_b^n - \tfrac{\Delta t |r_a^n|}{\rho_a^n} \mathbf{Grad}_a^{R^+}\left\{p_b^n\right\} + \Delta t |r_a^n| \boldsymbol{g}\right\} \\ &= \rho_a^n \frac{|r_a^{n+1}| - |r_a^n|}{\Delta t} - \rho_a^n r_a^n \mathrm{Div}_a^{R^-}\left\{\boldsymbol{u}_b^n\right\} \\ &\quad + \rho_a^n |r_a^n| \mathrm{Div}_a^{R^-}\left\{\tfrac{\Delta t}{\rho_b^n} \mathbf{Grad}_b^{R^+}\left\{p_c^n\right\} - \Delta t \mathbf{Grad}_b^{R^+}\left\{\boldsymbol{g} \cdot \boldsymbol{s}_c^n\right\}\right\} \\ &\simeq \rho_a^n \frac{|r_a^{n+1}| - |r_a^n|}{\Delta t} - \rho_a^n r_a^n \mathrm{Div}_a^{R^-}\left\{\boldsymbol{u}_b^n\right\} + \rho_a^n |r_a^n| \Delta_a^n, \end{aligned} \tag{D.74}$$

where Δ^n is defined as

$$\boxed{\Delta_a^n = \mathbf{Lap}_a^R\left(\left\{\tfrac{\Delta t}{\rho_b^n}\right\}, \left\{p_b^n\right\}\right) - \mathbf{Lap}_a^R\left(\{\Delta t\}, \{\boldsymbol{g}\}\right).} \tag{D.75}$$

Adding the dimensionless weighting coefficient Λ, the density time scheme is improved as

$$\rho_a^{n+1} = \frac{1}{R_a^{n+1}}\left[R_a^n \rho_a^n + \sum_{b \in \mathcal{P}} \widehat{M}_b (|r_b^{n+1}| w_{ab}^{n+1} - |r_b^n| w_{ab}^n) + \Delta t \Lambda \Delta^n\right]. \tag{D.76}$$

Analytical calculation of the renormalisation factor

For a fluid particle a, the renormalisation factor R_a reads:

$$\begin{aligned} R_a &\coloneqq \int_{\Omega_a \cap \Omega} w(\boldsymbol{s} - \boldsymbol{s}_a)\,|r|\,dr\,dz \\ &= |r_a| - \sum_{s \in S} \int_s \mathtt{sign}(r)\left(r_a \boldsymbol{W}(\boldsymbol{s} - \boldsymbol{s}_a) + Q(\boldsymbol{s} - \boldsymbol{s}_a)\right) \boldsymbol{e}_r\, dl \cdot \boldsymbol{n}_s \\ &\quad - 2\int_{\partial(\Omega_a \cap \Omega), r=0} \left(r_a \boldsymbol{W}(\boldsymbol{s} - \boldsymbol{s}_a) \cdot \boldsymbol{e}_r + Q(\boldsymbol{s} - \boldsymbol{s}_a)\right) dz, \end{aligned} \tag{D.77}$$

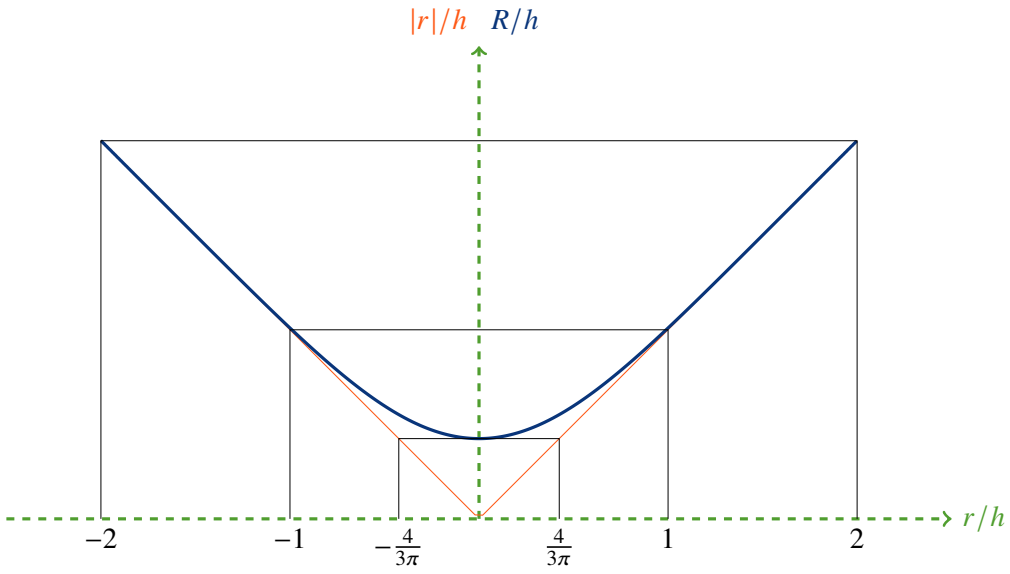


Figure D.23: Renormalisation factor in axisymmetric SPH ($R_a = \int_{\Omega_a} w|r|drdz$) against the radial direction.

where:

$$\begin{aligned} Q(\boldsymbol{s}-\boldsymbol{s}_a) &= -\tfrac{1}{2\pi}(1-\tfrac{q}{2})^5(1+\tfrac{5q}{2}+2q^2), \\ \boldsymbol{W}(\boldsymbol{s}-\boldsymbol{s}_a) &= \tfrac{Q(q)}{h^2q^2}(\boldsymbol{s}-\boldsymbol{s}_a), \end{aligned} \tag{D.78}$$

with $q := \frac{|\boldsymbol{s}-\boldsymbol{s}_a|}{h}$.

R_a and its gradient are plotted against the radial direction r for particle a far from solid boundaries ($\Omega_a \cap \Omega = \Omega_a$) on Figure D.23.

For $0 < r_a < 2$ we obtain after some algebra:

$$\begin{aligned} R_a &= |r_a| - \tfrac{r_a}{\pi}\arccos\left(\tfrac{r_a}{2}\right) \\ &+ \frac{3\left(5(\frac{r_a}{2})^2+56\right)(\frac{r_a}{2})^6\left(\ln\left(\sqrt{1-(\frac{r_a}{2})^2}+1\right)-\ln((\frac{r_a}{2}))\right)+\sqrt{1-(\frac{r_a}{2})^2}\left(-113(\frac{r_a}{2})^6-110(\frac{r_a}{2})^4+72(\frac{r_a}{2})^2+16\right)}{12\pi}. \end{aligned} \tag{D.79}$$

D.5.4 2-D axisymmetric SPH test-cases

Poiseuille flow in 2-D axisymmetric SPH

This test case is designed to check whether (or not) the viscous force is computationally consistent with the theoretical one. For a Poiseuille flow in a periodical pipe $D = 5\,\mathrm{m}$ which is presented in Figure D.24, the velocity profile reads:

$$u_z^{ana}(r) = u_{max}\left(1 - 4\frac{r^2}{D^2}\right), \tag{D.80}$$

where u_{max} is defined as:

$$u_{max} = \frac{D^2}{\nu}\boldsymbol{g}\cdot\boldsymbol{e}_z. \tag{D.81}$$

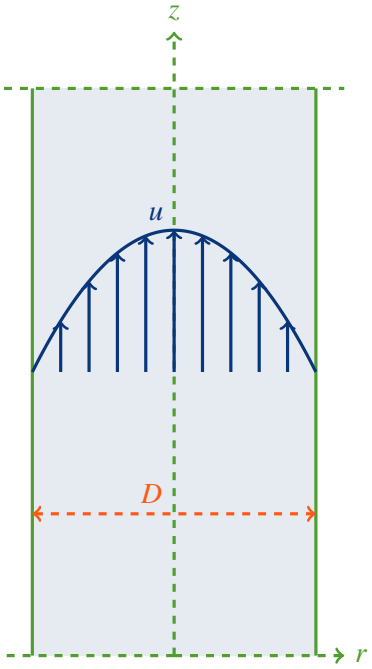


Figure D.24: Sketch of the Poiseuille flow in a circular pipe.

If the Reynolds number based on the maximum velocity is set to 10 and $\boldsymbol{g} = 1.0\,\mathrm{m\,s^{-2}}\boldsymbol{e}_z$ we get

$$\begin{cases} Re = \frac{u_{max} D}{2\nu} = 10, \\ u_{max} = \frac{D^2}{\nu}\boldsymbol{g}\cdot\boldsymbol{e}_z, \end{cases} \iff \begin{cases} u_{max} = 2.5\,\mathrm{m/s}, \\ \nu = 0.625\,\mathrm{m^2/s}. \end{cases} \tag{D.82}$$

We also set $\rho_0 = 1000\,\mathrm{kg\,m^{-3}}$, $c_0 = 20\,\mathrm{m\,s^{-1}}$, a background pressure $p_0 = 28\,000\,\mathrm{Pa}$ in the for the Equation of State $p = \frac{\rho_0 c_0^2}{\gamma}\left(\frac{\rho}{\rho_0} - 1\right)^{\gamma}$, with $\gamma = 7.0$.

In the simulations, we impose true periodic conditions on z with a computational pitch of $L = 10\,\mathrm{m}$ length (particles leaving the computational domain at $z = 10\,\mathrm{m}$ are instantaneously shifted one meter below, all particles next to $z = 0\,\mathrm{m}$ and $z = 10\,\mathrm{m}$ see neighbouring particles form the other part of the torus domain as if they were shifted from 10 m). In all this computations, the volume diffusion factor Λ is set to 0.

Velocities are initialised by the analytical values, as can be seen in Figure D.25, which gives the comparison of the velocity fields at the beginning ($t = 0\,\mathrm{s}$) and at the end ($t = 8\,\mathrm{s}$) of the simulation, which corresponds to 2 passes for particles next to the axis. The comparison of the velocity profile at $t = 5\,\mathrm{s}$ with the theoretical one which is imposed at $t = 0\,\mathrm{s}$ is presented on Figure D.28. The profile is well conserved during the simulation.

The variation of maximum velocity in time is presented in Figure D.27 for three space discretisations ($\frac{\Delta s}{D} = 0.01$, $\frac{\Delta s}{D} = 0.005$ and $\frac{\Delta s}{D} = 0.0025$). After the stabilisation at the beginning of simulation (during $\approx 1\,\mathrm{s}$), the absolute error is:

$$\epsilon = \frac{|u_{max} - u_{max,computed}|}{u_{max}} \approx 0.4\%. \tag{D.83}$$

The discrete L_2 errors for the vertical velocity is defined as:

$$\varepsilon_{u_z} = \sqrt{\frac{8}{LD^2} \sum_{b \in \mathcal{P}/r_b > 0} \widehat{V}_0 r_b \left(u_{z,b} - u_z^{ana}(r_b)\right)^2}. \tag{D.84}$$

Its evolution over time for the same discretisations is presented in Figure D.26.

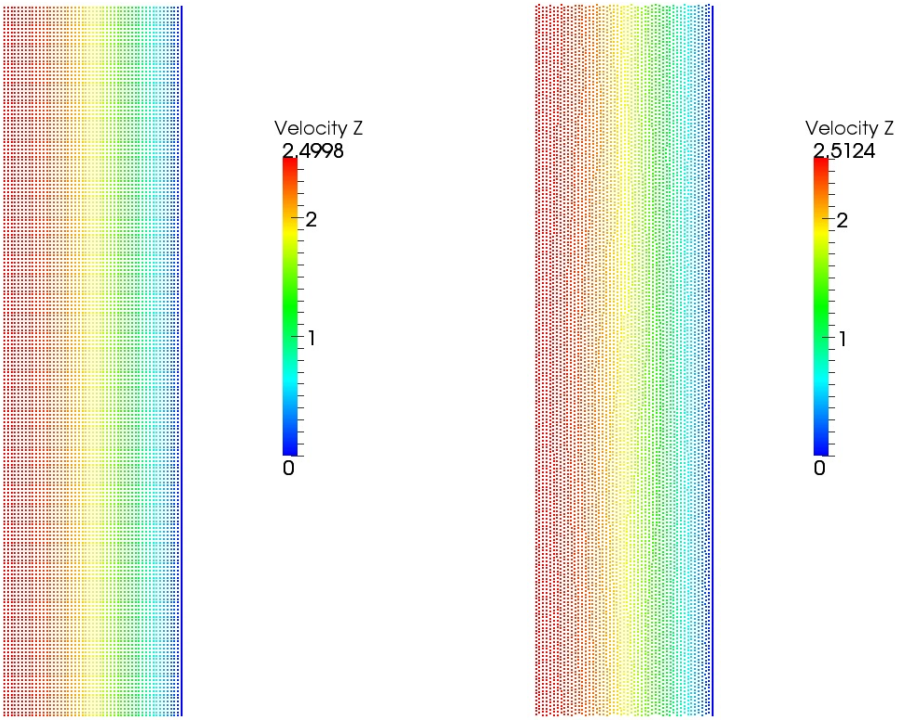


Figure D.25: Vertical component of the velocity field at $t = 0\,\mathrm{s}$ (left) and at $t = 8\,\mathrm{s}$ (right) ($\frac{\Delta s}{D} = 0.01$).

D.5.5 Conclusion

A 2-D axisymmetric formulation is proposed for SPH with a renormalisation factor unifying wall and axis correction. No singularity is present in the proposed scheme and particles are able to reach the axis. A basic test-case is proposed to check the correct behavior of the proposed operators (pressure gradient, mass equation and Laplacian used to compute shear stresses).

Future work could be made to increase the validation database and to extend this proposition to open boundaries as done in Chapter 3. The analytical expression used to compute the renormalisation factor has also to be extended to other shapes than horizontal or vertical segments treated in this Section.

Bibliography

Amicarelli, A., Agate, G., Guandalini, R., 2012. Development and validation of a SPH model using discrete surface elements at boundaries, in: Proceedings of the 7th International SPHERIC Workshop, Prato. pp. 369–374.

Brezzi, F., Pitkäranta, J., 1984. On the stabilization of finite element approximations of the Stokes equations, in: Efficient solutions of elliptic systems. Springer, pp. 11–19.

Brookshaw, L., 2003. Smooth particle hydrodynamics in cylindrical coordinates. ANZIAM Journal 44, 114–139.

De Leffe, M., Le Touzé, D., Alessandrini, B., 2009. Normal flux method at the boundary for SPH, in: 4^{th} SPHERIC workshop, Nantes, France. pp. 149 – 156.

Ferrand, M., Laurence, D., Rogers, B., Violeau, D., Kassiotis, C., 2013a. Unified semi-analytical

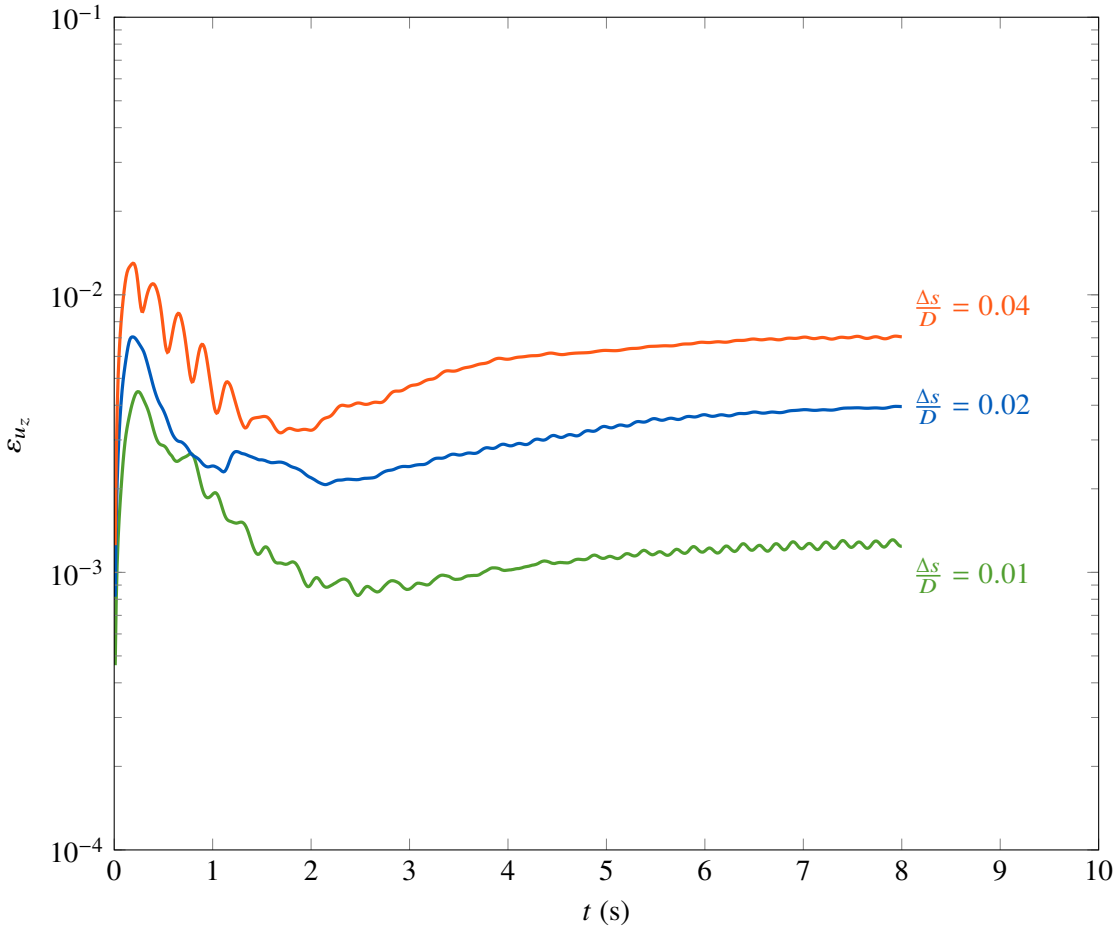


Figure D.26: 2-D axisymmetric L-2 errors for the velocity in the direction $\boldsymbol{e}_z$ defined by (D.84) over time.

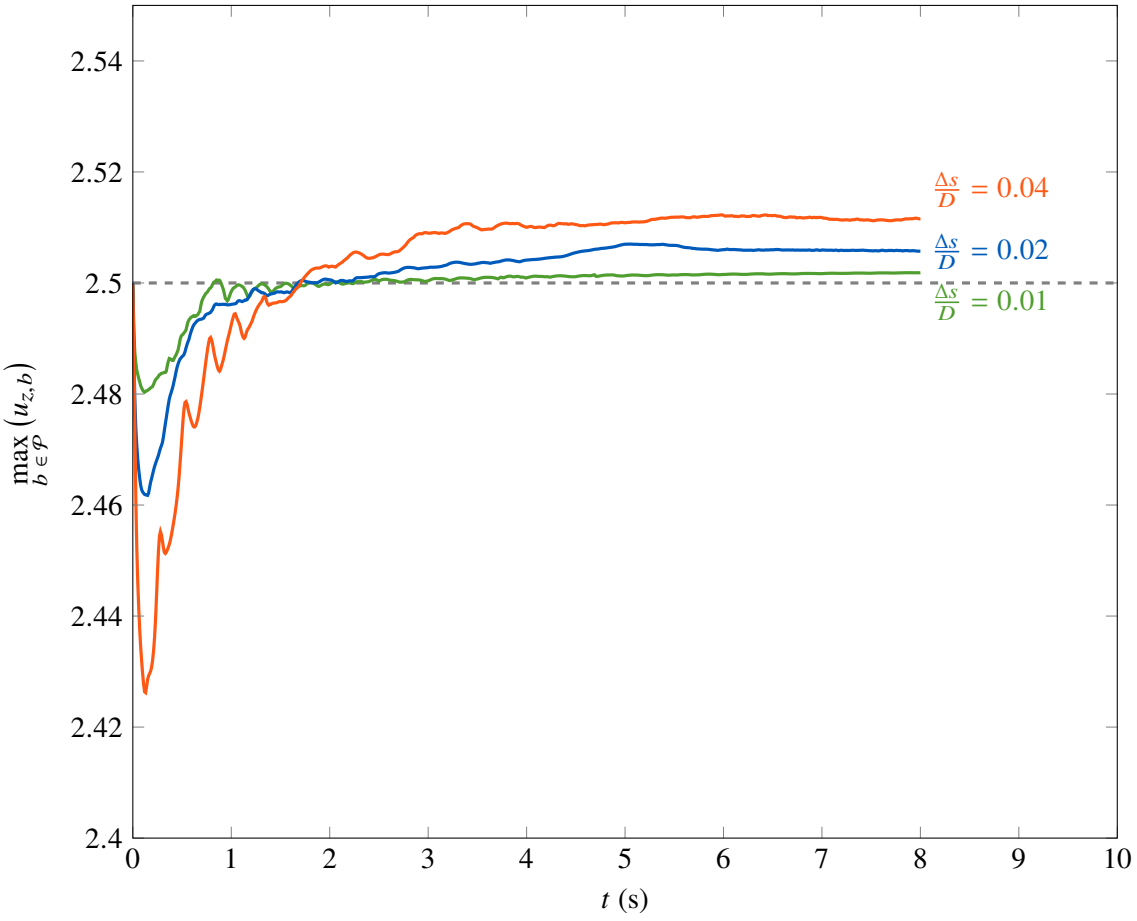


Figure D.27: Time evolution of the maximum velocity in the direction $\boldsymbol{e}_z$.

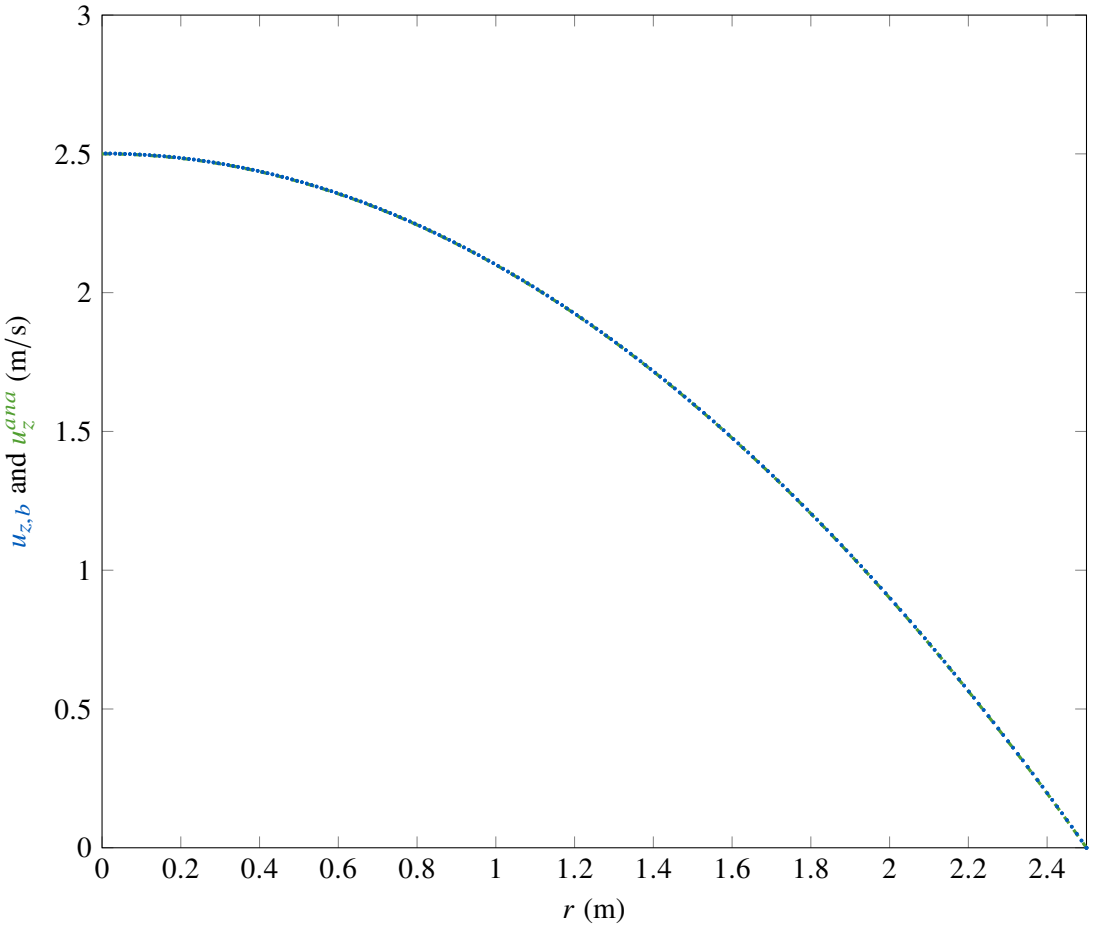


Figure D.28: Velocity profile at $t = 5\,\mathrm{s}$ at position $z = 5\,\mathrm{m}$.

wall boundary conditions for inviscid, laminar or turbulent flows in the meshless SPH method. International Journal for Numerical Methods in Fluids 71, 446–472.

Ferrand, M., Laurence, D.R., Rogers, B.D., Violeau, D., Kassiotis, C., 2013b. Unified semi-analytical wall boundary conditions for inviscid, laminar or turbulent flows in the meshless SPH method. International Journal for Numerical Methods in Fluids 71, 446–472. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.3666, doi:10.1002/fld.3666, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.3666.

Ghaïtanellis, A., Violeau, D., Ferrand, M., Abderrezzak, K.E.K., Leroy, A., Joly, A., 2018. A SPH elastic-viscoplastic model for granular flows and bed-load transport. Advances in Water Resources 111, 156–173.

Ghaitanellis, A., Violeau, D., Ferrand, M., Leroy, A., Joly, A., 2015. Application of the unified semi-analytical wall boundary conditions to multi-phase SPH, in: Proceedings of the 10th SPHERIC International Workshop, pp. 16–18.

Herant, M., Benz, W., 1992. Postexplosion hydrodynamics of SN 1987A. The Astrophysical Journal 387, 294–308.

Kleefsman, K., Fekken, G., Veldman, A., Iwanowski, B., Buchner, B., 2005. A volume-of-fluid based simulation method for wave impact problems. Journal of Computational Physics 206, 363–393. URL: http://www.sciencedirect.com/science/article/pii/S0021999104005170, doi:10.1016/j.jcp.2004.12.007.

Kulasegaram, S., Bonet, J., Lewis, R.W., Profit, M., 2004. A variational formulation based contact algorithm for rigid boundaries in two-dimensional SPH applications. Computational Mechanics 33, 316–325.

Leroy, A., 2014. A new incompressible SPH model: towards industrial applications. Ph.D. thesis. Université Paris-Est.

Li, M.K., Zhang, A.M., Ming, F.R., Sun, P.N., Peng, Y.X., 2020. An axisymmetric multiphase SPH model for the simulation of rising bubble. Computer Methods in Applied Mechanics and Engineering 366, 113039.

Mayrhofer, A., Ferrand, M., Kassiotis, C., Violeau, D., Morel, F.X., 2015. Unified semi-analytical wall boundary conditions in SPH: analytical extension to 3-D. Numerical Algorithms 68, 15–34.

Mayrhofer, A., Rogers, B.D., Violeau, D., Ferrand, M., 2013. Investigation of wall bounded flows using SPH and the unified semi-analytical wall boundary conditions. Computer Physics Communications 184, 2515–2527.

Monaghan, J.J., 1992. Smoothed Particle Hydrodynamics. Annual review of Astronomy and Astrophysics 30, 543–574.

Morris, J.P., Fox, P.J., Zhu, Y., 1997. Modeling Low Reynolds Number Incompressible Flows Using SPH. Journal of Computational Physics 136, 214–226.

Pesce, C.P., Tannuri, E.A., Casetta, L., 2006. The Lagrange equations for systems with mass varying explicitly with position: some applications to offshore engineering. Journal of the Brazilian Society of Mechanical Sciences and Engineering 28, 496–504.

Relaño Castillo, A., 2012. AxisSPH: devising and validating an axisymmetric smoothed particle hydrodynamics code .

Sun, P.N., Le Touzé, D., Oger, G., Zhang, A.M., 2021. An accurate sph volume adaptive scheme for modeling strongly-compressible multiphase flows. part 2: Extension of the scheme to cylindrical coordinates and simulations of 3d axisymmetric problems with experimental validations. Journal of Computational Physics 426, 109936.

Violeau, D., 2012. Fluid Mechanics and the SPH Method: Theory and Applications. Oxford University Press. URL: http://ukcatalogue.oup.com/product/academic/earthsciences/hydrology/9780199655526.do.

E. Tensors, differential operators and some notations

Abstract E.1 This appendix is made to define tensor notations, differential operators and to recall classical theorems and relationships.

Résumé E.1 Cette annexe est faite pour définir les notations tensorielles utilisées, les opérateurs différentiels et rappelle quelques résultats classiques d'intégration.

E.1 Tensor and index notations

In a fixed reference-frame $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$, with Einstein convention of sum over repeated indices we have:

0^{th}**-order tensor:** *e.g.* scalar T

1^{st}**-order tensor:** *e.g.* vector $\boldsymbol{u} = u_i\, \boldsymbol{e}_i$

2^{nd}**-order tensor:** *e.g.* matrix $\boldsymbol{\sigma} = \sigma_{ij}\, \boldsymbol{e}_i \otimes \boldsymbol{e}_j$

3^{rd}**-order tensor:** $\boldsymbol{a} = a_{ijk}\, \boldsymbol{e}_i \otimes \boldsymbol{e}_j \otimes \boldsymbol{e}_k$

$\vdots$

n^{th}**-order tensor:** $a^{(n)} = a_{i_1 i_2 \cdots i_n}\, \boldsymbol{e}_{i_1} \otimes \boldsymbol{e}_{i_2} \otimes \cdots \otimes \boldsymbol{e}_{i_n}$

Operation	Symbol	Formula
tensor product	$\otimes$	$\boldsymbol{u} \otimes \boldsymbol{u} = u_i u_j \qquad \boldsymbol{e}_i \otimes \boldsymbol{e}_j$
		$a^{(n)} \otimes b^{(m)} = a_{i_1 \cdots i_n} b_{j_1 \cdots j_m} \qquad \boldsymbol{e}_{i_1} \otimes \cdots \otimes \boldsymbol{e}_{i_n} \otimes \boldsymbol{e}_{j_1} \otimes \cdots \otimes \boldsymbol{e}_{j_m}$
inner product	$\cdot$	$\boldsymbol{u} \cdot \boldsymbol{u} = u_i u_i$
		$a^{(n)} \cdot b^{(m)} = a_{i_1 \cdots i_{n-1} k} b_{k j_2 \cdots j_m} \qquad \boldsymbol{e}_{i_1} \otimes \cdots \otimes \boldsymbol{e}_{i_{n-1}} \otimes \boldsymbol{e}_{j_2} \otimes \cdots \otimes \boldsymbol{e}_{j_m}$
double inner product	$:$	$\boldsymbol{\sigma} : \boldsymbol{\sigma} = \sigma_{ij} \sigma_{ji}$
		$a^{(n)} : b^{(m)} = a_{i_1 \cdots i_{n-2} k l} b_{l k j_3 \cdots j_m}\, \boldsymbol{e}_{i_1} \otimes \cdots \otimes \boldsymbol{e}_{i_{n-2}} \otimes \boldsymbol{e}_{j_3} \otimes \cdots \otimes \boldsymbol{e}_{j_m}$
cross product	$\times$ *or* $\wedge$	$\boldsymbol{u} \wedge \boldsymbol{v} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \wedge \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} u_2 v_3 - u_3 v_2 \\ u_3 v_1 - u_1 v_3 \\ u_1 v_2 - u_2 v_1 \end{pmatrix}$

Table E.1: Operations on tensors.

E.2 Differential operators and classical results

$$\mathbf{div}\,(\mathbf{curl}\,\boldsymbol{u}) = 0 \qquad \mathbf{curl}\,(\boldsymbol{\nabla} T) = \boldsymbol{0}$$

$$\mathbf{div}\,(\alpha \boldsymbol{u}) = \alpha \mathbf{div}\,\boldsymbol{u} + \boldsymbol{\nabla}\alpha \cdot \boldsymbol{u} \qquad \mathbf{curl}\,(\alpha \boldsymbol{u}) = \alpha \mathbf{curl}\,\boldsymbol{u} + \boldsymbol{\nabla}\alpha \wedge \boldsymbol{u}$$

$$\mathbf{div}\,(\boldsymbol{u} \wedge \boldsymbol{v}) = \boldsymbol{v} \cdot \mathbf{curl}\,\boldsymbol{u} - \boldsymbol{u} \cdot \mathbf{curl}\,\boldsymbol{v} \qquad \mathbf{curl}\,(\mathbf{curl}\,\boldsymbol{u}) = \mathbf{grad}\,(\mathbf{div}\,\boldsymbol{u}) - \boldsymbol{\Delta}\boldsymbol{u}$$

$$\mathbf{grad}\,(\boldsymbol{u} \cdot \boldsymbol{v}) = (\mathrm{grad}\,\boldsymbol{u}) \cdot \boldsymbol{v} + (\mathrm{grad}\,\boldsymbol{v}) \cdot \boldsymbol{u} + \boldsymbol{u} \wedge \mathbf{curl}\,\boldsymbol{v} + \boldsymbol{v} \wedge \mathbf{curl}\,\boldsymbol{u}$$

Operator	Symbol	Definition	Cartesian formula
gradient	**grad**	$\nabla^{(n+1)}\left[a^{(n)}\right] \cdot \mathbf{d}\boldsymbol{x} = \mathrm{d}a^{(n)}\big\vert_{\mathbf{d}\boldsymbol{x}}$	$\boldsymbol{\nabla} T = \dfrac{\partial T}{\partial x_i}\boldsymbol{e}_i$
	or		$\boldsymbol{\nabla}\boldsymbol{u} = \dfrac{\partial \boldsymbol{u}}{\partial x_j} \otimes \boldsymbol{e}_j = \dfrac{\partial u_i}{\partial x_j}\boldsymbol{e}_i \otimes \boldsymbol{e}_j$
	$\boldsymbol{\nabla}$		$\nabla^{(n+1)}\left[a^{(n)}\right] = \dfrac{\partial a^{(n)}}{\partial x_{i_{n+1}}} \otimes \boldsymbol{e}_{i_{n+1}}$
divergence	$\nabla\cdot$	$\nabla^{(n-1)} \cdot \left[a^{(n)}\right] = \nabla^{(n+1)}\left[a^{(n)}\right] : \mathbf{1}$	$\mathbf{div}\ \boldsymbol{u} = \dfrac{\partial u_i}{\partial x_i}$
	or		$\mathbf{div}\ \boldsymbol{\sigma} = \dfrac{\partial \sigma_{ij}}{\partial x_j}\boldsymbol{e}_i$
	div		$\mathbf{div}^{(n-1)}\left[a^{(n)}\right] = \dfrac{\partial a_{i_1\cdots i_n}}{\partial x_{i_n}}\boldsymbol{e}_{i_1} \otimes \cdots \otimes \boldsymbol{e}_{i_{n-1}}$
Laplacian	Δ	$\Delta a^{(n)} = \mathbf{div}^{(n)}\left\{\nabla^{(n+1)}\left[a^{(n)}\right]\right\}$	$\Delta T = \dfrac{\partial^2 T}{\partial x_i \partial x_i}$
	or		$\boldsymbol{\Delta}\boldsymbol{u} = \dfrac{\partial^2 \boldsymbol{u}}{\partial x_j \partial x_j} = \dfrac{\partial^2 u_i}{\partial x_j \partial x_j}\boldsymbol{e}_i$
	Lap		$\Delta^{(n)} a^{(n)} = \dfrac{\partial^2 a^{(n)}}{\partial x_{i_{n+1}} \partial x_{i_{n+1}}} = \dfrac{\partial^2 a_{i_1\cdots i_n}}{\partial x_{i_{n+1}} \partial x_{i_{n+1}}}\boldsymbol{e}_{i_1} \otimes \cdots \otimes \boldsymbol{e}_{i_n}$
curl	$\boldsymbol{\nabla}\wedge$		$\mathbf{curl}\ \boldsymbol{u} = \begin{pmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_3} \end{pmatrix} \wedge \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} \frac{\partial u_3}{\partial x_2} - \frac{\partial u_2}{\partial x_3} \\ \frac{\partial u_1}{\partial x_3} - \frac{\partial u_3}{\partial x_1} \\ \frac{\partial u_2}{\partial x_1} - \frac{\partial u_1}{\partial x_2} \end{pmatrix}$
	or		
	curl		

Table E.2: Differential operators.

Kelvin-Stokes theorem

$$\int_{\mathcal{S}} \mathbf{curl}\ \boldsymbol{u} \cdot \mathbf{d}\boldsymbol{S} = \int_{\partial\mathcal{S}} \boldsymbol{u} \cdot \mathbf{d}\boldsymbol{l}$$

where $\mathbf{d}\boldsymbol{S}$ is the out-warding surface element.

Divergence theorem [Green-Ostrogradski]

$$\boxed{\int_{\Omega} \mathbf{div}^{(n-1)}\left[a^{(n)}\right]\ \mathrm{d}\Omega = \int_{\partial\Omega} a^{(n)} \cdot \mathbf{d}\boldsymbol{S}}$$

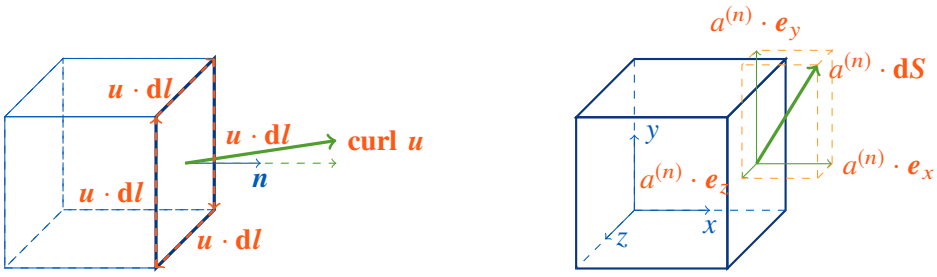
Curl theorem

$$\int_{\Omega} \mathbf{curl}\ \boldsymbol{u}\ \mathrm{d}\Omega = -\int_{\partial\Omega} \boldsymbol{u} \wedge \mathbf{d}\boldsymbol{S}$$

Leibniz theorem

$$\boxed{\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega(t)} a^{(n)}\,\mathrm{d}\Omega = \int_{\Omega(t)} \frac{\partial a^{(n)}}{\partial t}\,\mathrm{d}\Omega + \int_{\partial\Omega(t)} a^{(n)}\boldsymbol{v} \cdot \mathbf{d}\boldsymbol{S}}$$

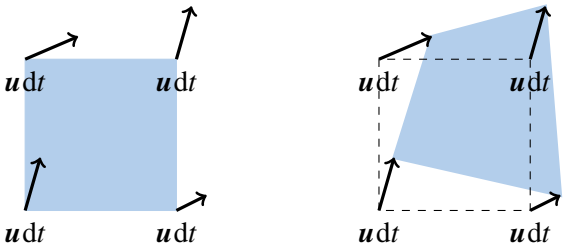
where $\boldsymbol{v}$ is the velocity of the control volume $\Omega(t)$ displacement.



(a) Kelvin-Stokes theorem.

(b) Divergence theorem.

Figure E.1: Illustrations of the Kelvin-Stokes and the divergence theorems.



(a) Fluid volume Ω_f at time t.

(b) Fluid volume Ω_f at time $t + \mathrm{d}t$.

Figure E.2: Illustration of Leibniz theorem.

Bibliography

Adami, S., Hu, X.Y., Adams, N.A., 2012. A generalized wall boundary condition for smoothed particle hydrodynamics. Journal of Computational Physics 231, 7057–7075.

Amicarelli, A., Agate, G., Guandalini, R., 2012. Development and validation of a SPH model using discrete surface elements at boundaries, in: Proceedings of the 7th International SPHERIC Workshop, Prato. pp. 369–374.

Antonopoulos, D., Dougalis, V., Mitsotakis, D., 2017. Error estimates for galerkin approximations of the serre equations. SIAM Journal on Numerical Analysis 55. doi:10.1137/16M1078355.

Antuono, M., Colagrossi, A., 2013. The damping of viscous gravity waves. Wave Motion 50, 197–209.

Antuono, M., Colagrossi, A., Marrone, S., Molteni, D., 2010. Free-surface flows solved by means of sph schemes with numerical diffusive terms. Computer Physics Communications 181, 532–549.

Archambeau, F., Méchitoua, N., Sakiz, M., 2004. Code_saturne: a finite volume method for the computation of turbulent incompressible flows - industrial applications. International Journal on Finite Volumes 1, 1–62.

Bahlali, M.L., Henry, C., Carissimo, B., 2020. On the well-mixed condition and consistency issues in hybrid Eulerian/Lagrangian stochastic models of dispersion. Boundary-Layer Meteorology 174, 275–296.

Beji, S., Battjes, J.A., 1993. Experimental investigation of wave-propagation over a bar. Coastal Engineering 19, 151–162.

Benoit, M., Dias, F., Herterich, J., Scolan, Y.M., 2018. Un cas-test discriminant pour la simulation de la propagation et du run-up de trains de vagues de type tsunami, in: Proceedings of 16th Journées de l'Hydrodynamique, p. 14.

Benson, D.J., 1989. An efficient, accurate, simple ALE method for nonlinear finite element programs. Computer Methods in Applied Mechanics and Engineering 72, 305–350.

Benz, W., 1990. Smoothed particle hydrodynamics: A review. In numerical modeling of nonlinear stellar pulsation: Problems and prospects. Kluwer Academic Dordrecht .

Blondel, F., Audebert, B., Pasutto, T., Stanciu, M., 2013. Condensation models and boundary conditions for non-equilibrium wet steam flows. International Journal on Finite Volumes 10, 1–53.

Bonelle, J., Ern, A., 2014. Analysis of Compatible Discrete Operator schemes for elliptic problems on polyhedral meshes. ESAIM Mathematical Modelling and Numerical Analysis 48, 553–581.

Bonet, J., Lok, T.S.L., 1999. Variational and momentum preservation aspects of Smooth Particle Hydrodynamic formulations. Computer Methods in applied mechanics and engineering 180, 97–115.

Boscheri, W., Balsara, D.S., Dumbser, M., 2014. Lagrangian ADER-WENO finite volume schemes on unstructured triangular meshes based on genuinely multidimensional HLL Riemann solvers. Journal of Computational Physics 267, 112–138.

Bouscasse, B., Colagrossi, A., Marrone, S., Souto-Iglesias, A., 2017. Sph modelling of viscous flow past a circular cylinder interacting with a free surface. Computers & Fluids 146, 190–212.

Bouscasse, B., Marrone, S., Colagrossi, A., Di Mascio, A., 2013. Multi-purpose interfaces for coupling SPH with other solvers, in: 8th SPHERIC International Workshop.

Bradford, S.F., 2011. Nonhydrostatic model for surf zone simulation. Journal of waterway, port, coastal, and ocean engineering 137, 163–174.

Braun, S., Wieth, L., Koch, R., Bauer, H.J., 2015. A Framework for Permeable Boundary Conditions in SPH: Inlet, Outlet, Periodicity, in: 10th SPHERIC International Workshop.

Brezzi, F., Pitkäranta, J., 1984. On the Stabilization of Finite Element Approximations of the Stokes Equations, in: Hackbusch, W. (Ed.), Efficient Solutions of Elliptic Systems. Vieweg+Teubner Verlag. volume 10 of *Notes on Numerical Fluid Mechanics*, pp. 11–19. URL: http://dx.doi.org/10.1007/978-3-663-14169-3_2, doi:10.1007/978-3-663-14169-3_2.

Brezzi, F., Pitkäranta, J., 1984. On the stabilization of finite element approximations of the Stokes equations, in: Efficient solutions of elliptic systems. Springer, pp. 11–19.

Brookshaw, L., 2003. Smooth particle hydrodynamics in cylindrical coordinates. ANZIAM Journal 44, 114–139.

Carré, G., Del Pino, S., Després, B., Labourasse, E., 2009. A cell-centered lagrangian hydrodynamics scheme on general unstructured meshes in arbitrary dimension. Journal of Computational Physics 228, 5160–5183.

Celik, I., Rodi, W., 1984. Simulation of free-surface effects in turbulent channel flows. Physiochemical Hydrodynamics 5, 217–227.

Chan, R., Street, R., 1970. Computer study of finite-amplitude water waves. Journal of Computational Physics 6, 68–94.

Chemartin, L., Lalande, P., Montreuil, E., Delalondre, C., Cheron, B., Lago, F., 2009. Three dimensional simulation of a DC free burning arc. Application to lightning physics. Atmospheric Research 91, 371–380.

Chen, L., Zang, J., Hills, A., Morgan, G., Plummer, A., 2014. Numerical investigation of wave-structure interaction using OpenFOAM. Ocean Engineering 88, 91–109.

Chiron, L., de Leffe, M., Oger, G., Touzé, D.L., 2019. Fast and accurate SPH modelling of 3D complex wall boundaries in viscous and non viscous flows. Computer Physics Communications 234, 93–111. URL: http://www.sciencedirect.com/science/article/pii/S0010465518302832, doi:https://doi.org/10.1016/j.cpc.2018.08.001.

Chorin, A.J., 1968. Numerical solution of the Navier-Stokes equations. Mathematics of computation 22, 745–762.

Colas, C., Ferrand, M., Hérard, J.M., Latché, J.C., Le Coupanec, E., 2019. An Implicit Integral Formulation to Model Inviscid Fluid Flows in Obstructed Media. Computers & Fluids 188, 136–163.

Crespo, A.J., Gómez-Gesteira, M., Dalrymple, R.A., 2007. Boundary conditions generated by dynamic particles in sph methods. Computers, materials and continua 5, 173–184.

Dal Secco, S., Juan, O., Louis-Louisy, M., Lucas, J.Y., Plion, P., Porcheron, L., 2015. Using a genetic algorithm and CFD to identify low NOx configurations in an industrial boiler. Fuel 158, 672–683.

Das, R., Cleary, P.W., 2008. Modelling 3D fracture and fragmentation in a thin plate under high velocity projectile impact using SPH, in: Proc. III^{rd} SPHERIC International Workshop, Lausanne, Suisse.

De Leffe, M., Le Touzé, D., Alessandrini, B., 2009. Normal flux method at the boundary for SPH, in: 4^{th} SPHERIC workshop, Nantes, France. pp. 149 – 156.

Dean, R.G., Dalrymple, R.A., 1991. Water wave mechanics for engineers and scientists. volume 2 of *Advanced Series on Ocean Engineering*. World Scientific, Singapore.

Dehnen, W., Aly, H., 2012. Improving convergence in smoothed particle hydrodynamics simulations without pairing instability. Monthly Notices of the Royal Astronomical Society 425, 1068–1082.

Dehoux, F., Benhamadouche, S., Manceau, R., 2017. An elliptic blending differential flux model for natural, mixed and forced convection. International Journal of Heat and Fluid Flow 63, 190–204.

Di Monaco, A., Manenti, S., Gallati, M., Sibilla, S., Agate, G., Guandalini, R., Maffio, A., 2009. A semi-analytic approach for SPH modelling of solid boundaries, in: 4^{th} SPHERIC workshop, Nantes, France. pp. 165 – 172.

Di Pietro, D.A., Ern, A., 2015. Hybrid high-order methods for variable-diffusion problems on general meshes. Comptes Rendus Mathématique 353, 31–34.

Donea, J., Huerta, A., Ponthot, J.P., guez Ferran, A.R., 2004. Arbitrary Lagrangian-Eulerian Methods, in: Stein, E., de Borst, R., Hughes, T.J.R. (Eds.), Encyclopedia of Computational Mechanics. John Wiley & Sons. chapter 14, pp. 413–437.

Durst, F., Maxworth, T., Pereira, J.C.F., 1989. Piston-driven, unsteady separation at a sudden expansion in a tube: Flow visualization and LDA measurements. Physics of Fluids A 1, 1249.

Dutykh, D., Clamond, D., 2014. Efficient computation of steady solitary gravity waves. Wave Motion 51, 86–99.

Edwards, J.D., Morel, J.E., Knoll, D.A., 2011. Nonlinear variants of the TR/BDF2 method for thermal radiative diffusion. Journal of Computational Physics 230, 1198 – 1214. URL: http://www.sciencedirect.com/science/article/pii/S0021999110005917, doi:https://doi.org/10.1016/j.jcp.2010.10.035.

Espanol, P., Revenga, M., 2003. Smoothed dissipative particle dynamics. Physical Review E 67, 026705.

Eymard, R., Gallouët, T., Guichard, C., Herbin, R., Masson, R., 2014. TP or not TP, that is the question. Computational Geosciences 18, 285–296. doi:10.1007/s10596-013-9392-9.

Eymard, R., Gallouët, T., Herbin, R., 2000. Finite volume methods. Handbook of Numerical Analysis 7, 713–1018.

Eymard, R., Gallouët, T., Herbin, R., 2010. Discretization of heterogeneous and anisotropic diffusion problems on general nonconforming meshes SUSHI: a scheme using stabilization and hybrid interfaces. IMA journal of numerical analysis 30, 1009–1043.

Eymard, R., Gallouët, T., Herbin, R., 2006. Finite Volume Methods. volume 7.

Eymard, R., Henry, G., Herbin, R., Hubert, F., Klöfkorn, R., Manzini, G., 2011. 3D Benchmark on Discretization Schemes for Anisotropic Diffusion Problems on General Grids, Springer. pp. 95–130.

Farhat, C., Geuzaine, P., Grandmont, C., 2001. The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids. Journal of Computational Physics 174, 669–694.

Fatehi, R., Manzari, M.T., 2011. A remedy for numerical oscillations in weakly compressible smoothed particle hydrodynamics. International Journal for Numerical Methods in Fluids 67, 1100–1114.

Feldman, J., Bonet, J., 2007. Dynamic refinement and boundary contact forces in SPH with applications in fluid flow problems. Int. Journal for Numerical Methods in Engineering 72, 295 – 324. Meshfree Methods: Recent Advances and New Applications.

Fenton, J.D., 1985. A fith-order Stokes theory for steady waves. Journal of Waterway, Port, Coastal and Ocean Engineering 111, 216–234.

Ferrand, M., 2009. Etude comparative de quelques modèles algébriques explicites pour les tensions de Reynolds et les flux turbulents de scalaires passifs et actifs et application au cas d'un étang saumâtre. Base documentaire Eureka .

Ferrand, M., 2011. Unified semi-analytical wall boundary conditions for inviscid, laminar and turbulent slightly compressible flows in SPARTACUS-2D combined with an improved time integration scheme on the continuity equation. The University of Manchester (United Kingdom).

Ferrand, M., Fontaine, J., Angelini, O., 2014. An anisotropic diffusion finite volume algorithm using a small stencil, in: Finite Volumes for Complex Applications VII-Elliptic, Parabolic and Hyperbolic Problems, pp. 577–585.

Ferrand, M., Harris, J.C., 2021. Finite volume arbitrary lagrangian-eulerian schemes using dual meshes for ocean wave applications. Computers & Fluids 219, 104860.

Ferrand, M., Joly, A., Kassiotis, C., Violeau, D., Leroy, A., Morel, F.X., Rogers, B.D., 2017. Unsteady open boundaries for SPH using semi-analytical conditions and Riemann solver in 2D. Computer Physics Communications 210, 29 – 44. URL: http://www.sciencedirect.com/science/article/pii/S0010465516302806, doi:https://doi.org/10.1016/j.cpc.2016.09.009.

Ferrand, M., Laurence, D., Rogers, B., Violeau, D., Kassiotis, C., 2013a. Unified semi-analytical wall boundary conditions for inviscid, laminar or turbulent flows in the meshless SPH method. International Journal for Numerical Methods in Fluids 71, 446–472.

Ferrand, M., Laurence, D., Rogers, B.D., Violeau, D., 2010. Improved time scheme integration approach for dealing with semi-analytical wall boundary conditions in Spartacus2D, in: 5th International SPHERIC SPH Workshop.

Ferrand, M., Laurence, D.R., Rogers, B.D., Violeau, D., Kassiotis, C., 2013b. Unified semi-analytical wall boundary conditions for inviscid, laminar or turbulent flows in the meshless SPH method. International Journal for Numerical Methods in Fluids 71, 446–472. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.3666, doi:10.1002/fld.3666, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.3666.

Ferrant, P., 2000. Fully nonlinear interactions of long-crested wave packets with a three-dimensional body, in: 22nd Symposium on Naval Hydrodynamics, pp. 403–416.

Ferrari, A., Dumbser, M., Toro, E.F., Armanini, A., 2009. A new 3D parallel SPH scheme for free surface flows. Computers & Fluids 38, 1203–1217.

Fonty, T., 2019. Modélisation de l'entraînement d'air dans la méthode SPH .

Fonty, T., Ferrand, M., Leroy, A., Joly, A., Violeau, D., 2019. Mixture model for two-phase flows with high density ratios: A conservative and realizable SPH formulation. International Journal of Multiphase Flow 111, 158–174.

Fonty, T., Ferrand, M., Leroy, A., Violeau, D., 2020. Air Entrainment Modeling in the SPH Method: A Two-Phase Mixture Formulation with Open Boundaries. Flow, Turbulence and Combustion , 1573–1987doi:10.1007/s10494-020-00165-7.

Fourtakas, G., Dominguez, J.M., Vacondio, R., Rogers, B.D., 2019. Local uniform stencil (lust) boundary condition for arbitrary 3-d boundaries in parallel smoothed particle hydrodynamics (sph) models. Computers & Fluids 190, 346–361.

Fromm, J., 1968. A method for reducing dispersion in convective difference schemes. Applied Mathematics Modelling 19, 640–653.

Gauffre, M.C., Benhamadouche, S., Badel, P.B., 2020. Wall-Modeled Large Eddy Simulation of the Flow Through PWR Fuel Assemblies at Re_H = 66 000 - Validation on CALIFS Experimental Setup. Nuclear Technology 206, 255–265.

Ghaïtanellis, A., Violeau, D., Ferrand, M., Abderrezzak, K.E.K., Leroy, A., Joly, A., 2018. A SPH elastic-viscoplastic model for granular flows and bed-load transport. Advances in Water Resources 111, 156–173.

Ghaitanellis, A., Violeau, D., Ferrand, M., Leroy, A., Joly, A., 2015a. Application of the unified semi-analytical wall boundary conditions to multi-phase SPH, in: 10th SPHERIC International Workshop.

Ghaitanellis, A., Violeau, D., Ferrand, M., Leroy, A., Joly, A., 2015b. Application of the unified semi-analytical wall boundary conditions to multi-phase SPH, in: Proceedings of the 10th SPHERIC International Workshop, pp. 16–18.

Gingold, R.A., Monaghan, J.J., 1977. Smoothed Particle hydrodynamic: theory and application to non-spherical stars. Monthly Notices of the Royal Astronomical Society 181, 375–389.

Goldstein, H., Poole, C., Safko, J., 2002. Conservation of the Hamiltonian. Addison Wesley, San Francisco. $3^{rd} edition$.

Grilli, S.T., Horrillo, J., 1997. Numerical generation and absorption of fully nonlinear periodic waves. Journal of Engineering Mechanics 123, 1060–1069.

Hammad, K.J., Ötügen, M.V., Arik, E.B., 1999. A PIV study of the laminar axisymmetric sudden expansion flow. Experiments in Fluids 26, 266–272.

Harlow, F.H., Welch, J.E., 1965. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. Physics of Fluids 8, 2182–2189.

Harten, A., 1982. High resolution schemes for hyperbolic conservation laws. Journal of Computational Physics .

Harten, A., Osher, S., 1987. Uniformly high order accurate nonoscillatory schemes i. SIAM Journal of Numerical Analysis 24, 279–309.

Harten, A., Osher, S., Engquist, B., Chakravarthy, S., 1986. Some results on uniformly high order accurate essentially non-oscillatory schemes. Applied Numerical Mathematics 2, 347–377.

Hayward, A.T.J., 1967. Compressibility equations for liquids: a comparative study. British Journal of Applied Physics 18, 965–977. URL: https://doi.org/10.1088%2F0508-3443%2F18%2F7%2F312, doi:10.1088/0508-3443/18/7/312.

Helluy, Philippe, Golay, Frédéric, Caltagirone, Jean-Paul, Lubin, Pierre, Vincent, Stéphane, Drevard, Deborah, Marcer, Richard, Fraunié, Philippe, Seguin, Nicolas, Grilli, Stephan, Lesage, Anne-Cécile, Dervieux, Alain, Allain, Olivier, 2005. Numerical simulations of wave breaking. ESAIM: M2AN 39, 591–607. URL: https://doi.org/10.1051/m2an:2005024, doi:10.1051/m2an:2005024.

Herant, M., 1994. Dirty tricks for sph. Memorie della Societa Astronomica Italiana 65, 1013.

Herant, M., Benz, W., 1992. Postexplosion hydrodynamics of SN 1987A. The Astrophysical Journal 387, 294–308.

Hérard, J.M., Martin, X., Preprint: hal-01166478. An integral approach to compute compressible fluid flows in domains containing obstacles. International Journal on Finite Volumes .

Hérault, A., Bilotta, G., Dalrymple, R.A., 2010. SPH on GPU with CUDA. Journal of Hydraulic Research 48, 74–79.

Hérault, A., Bilotta, G., Dalrymple, R.A., 2014. Achieving the best accuracy in an SPH implementation, in: 9th SPHERIC International Workshop.

Hervouet, J., 2007a. Hydrodynamics of Free Surface Flows: Modelling with the Finite Element Method. John Wiley.

Hervouet, J.M., 2007b. Hydrodynamics of free surface flows: modelling with the finite element method. Wiley and Sons, Ltd.

Hirt, C.W., Amsden, A.A., Cook, J.L., 1974. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. Journal of Computational Physics 14, 227–253.

Hirt, C.W., Nichols, B.D., 1981. Volume of fluid (VOF) method for the dynamics of free boundaries. Journal of Computational Physics 39, 201–225.

Huseby, M., Grue, J., 2000. An experimental investigation of higher-harmonic wave forces on a vertical cylinder. Journal of Fluid Mechanics 414, 75–103.

Inutsuka, S., 1994. Godunov-type SPH. Memorie della Societa Astronomica Italiana 65, 1027.

Inutsuka, S.i., 2002. Reformulation of smoothed particle hydrodynamics with Riemann solver. arXiv preprint astro-ph/0206401 .

Issa, R., Lee, E.S., Violeau, D., Laurence, D., 2005. Incompressible separated flows simulations with the smoothed particle hydrodynamics gridless method. International Journal for Numerical Methods in Fluids 47, 1101–1106. URL: http://dx.doi.org/10.1002/fld.864, doi:10.1002/fld.864.

Issa, R.I., Gosman, A., Watkins, A., 1986. The computation of compressible and incompressible recirculating flows by a non-iterative implicit scheme. Journal of Computational Physics 62, 66–82.

Jacobsen, N.G., Fuhrman, D.R., Fredsøe, J., 2012. A wave generation toolbox for the open-source CFD library: OpenFOAM. International Journal for Numerical Methods in Fluids 70, 1073–1088.

Jeffrey, A., 1976. Quasilinear Hyperbolic Systems and Waves. Pitman, London.

J.P. Borisi, D.B., 1973. Flux corrected transport i: Shasta, a fluid transport algorithm that works. Journal of computational physics 11, 38–69.

Kassiotis, C., Ferrand, M., Violeau, D., Rogers, B.D., Stansby, P.K., Benoit, M., et al., 2011. Coupling SPH with a 1-D Boussinesq-type wave model, in: 6th International SPHERIC Workshop, pp. 241–247.

Kleefsman, K., Fekken, G., Veldman, A., Iwanowski, B., Buchner, B., 2005. A volume-of-fluid based simulation method for wave impact problems. Journal of Computational Physics 206, 363–393. URL: http://www.sciencedirect.com/science/article/pii/S0021999104005170, doi:10.1016/j.jcp.2004.12.007.

Kostorz, W., Esmail-Yakas, A., 2021a. Semi-analytical smoothed-particle hydrodynamics correction factors for polynomial kernels and piecewise-planar boundaries. International Journal for Numerical Methods in Engineering .

Kostorz, W., Esmail-Yakas, A., 2021b. Semi-analytical treatment of spherical solids in smoothed-particle hydrodynamics fluid simulations. International Journal for Numerical Methods in Engineering 122, 1732–1745.

Kulasegaram, S., Bonet, J., Lewis, R.W., Profit, M., 2004a. A variational formulation based contact algorithm for rigid boundaries in two-dimensional SPH applications. Computational Mechanics 33, 316–325.

Kulasegaram, S., Bonet, J., Lewis, R.W., Profit, M., 2004b. A variational formulation based contact algorithm for rigid boundaries in two-dimensional SPH applications. Computational Mechanics 33, 316–325.

Kunz, P., Hirschler, M., Huber, M., Nieken, U., 2016. Inflow/outflow with dirichlet boundary conditions for pressure in isph. Journal of Computational Physics 326, 171–187.

Lamb, H., 1945. Hydrodynamics. Dover Books.

Lastiwka, M., Basa, M., Quinlan, N.J., 2009. Permeable and non-reflecting boundary conditions in SPH. International Journal for Numerical Methods inf Fluids , 709–724.

Launder, B.E., Spalding, D.B., 1972. Mathematical Models of Turbulence. London: Academic Press .

Layton, W., Trenchea, C., 2012. Stability of two IMEX methods, CNLF and BDF2-AB2, for uncoupling systems of evolution equations. Applied Numerical Mathematics 62, 112–120.

Leonard, B.P., 1990. The ultimate conservative difference scheme appliedto unsteady one-dimensional advection. Computer Methods in Applied Mechanics and Engineering .

Leroy, A., 2014a. A new incompressible SPH model : towards industrial applications. Modeling and Simulation. Ph.D. thesis. Universitée Paris-Est.

Leroy, A., 2014b. A new incompressible SPH model: towards industrial applications. Ph.D. thesis. Université Paris-Est.

Leroy, A., Violeau, D., Ferrand, M., Fratter, L., Joly, A., 2016. A new open boundary formulation for incompressible SPH. Computers & Mathematics with Applications 72, 2417 – 2432. URL: http://www.sciencedirect.com/science/article/pii/S0898122116305107, doi:https://doi.org/10.1016/j.camwa.2016.09.008.

Leroy, A., Violeau, D., Ferrand, M., Joly, A., 2015. Buoyancy modelling with incompressible SPH for laminar and turbulent flows. International Journal for Numerical Methods in Fluids 78, 455–474. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.4025, doi:10.1002/fld.4025, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.4025.

Leroy, A., Violeau, D., Ferrand, M., Kassiotis, C., 2014. Unified semi-analytical wall boundary conditions applied to 2-D incompressible SPH. Journal of Computational Physics 261, 106 – 129. URL: http://www.sciencedirect.com/science/article/pii/S0021999113008437, doi:https://doi.org/10.1016/j.jcp.2013.12.035.

Leveque, R.J., 1992. Numerical methods for conservation laws. volume 132. Springer.

Li, M.K., Zhang, A.M., Ming, F.R., Sun, P.N., Peng, Y.X., 2020. An axisymmetric multiphase SPH model for the simulation of rising bubble. Computer Methods in Applied Mechanics and Engineering 366, 113039.

Lin, P., 2008. Numerical modeling of water waves. CRC Press.

Lind, S., Xu, R., Stansby, P., Rogers, B., 2012. Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves. Journal of Computational Physics 231, 1499 – 1523. URL: http://www.sciencedirect.com/science/article/pii/S0021999111006279, doi:https://doi.org/10.1016/j.jcp.2011.10.027.

Longuet-Higgins, M.S., Cokelet, E., 1976. The deformation of steep surface waves on water – I. A numerical method of computation. Proceedings of the Royal Society A 350, 1–26.

Lu, H., Park, Y.S., Cho, Y.S., 2017. Modeling of long waves generated by bottom-tilting wave maker. Coastal Engineering 122, 1–9.

Lucy, L.B., 1977. A Numerical Approach to Testing the Fission Hypothesis. The Astronomical Journal 82, 1013–1924.

Ma, G., Shi, F., Kirby, J.T., 2012. Shock-capturing non-hydrostatic model for fully dispersive surface wave processes. Ocean Modelling 43-44, 22–35.

Ma, L., Nmira, F., Consalvi, J., 2019. Verification and validation of a variable–density solver for fire safety applications. Numerical Heat Transfer, Part B: Fundamentals 76, 107–129.

Madsen, P.A., Fuhrman, D.R., Schäffer, H.A., 2008. On the solitary wave paradigm for tsunamis. Journal of Geophysical Research 113, C12012.

Mahmood, O., Kassiotis, C., Violeau, D., Rogers, B.D., Ferrand, M., 2012. Absorbing inlet/outlet boundary conditions for SPH 2-D turbulent free-surface flows, in: 7th SPHERIC International Workshop.

Marongiu, J.C., Leboeuf, F., Caro, J., 2009. Low Mach number numerical schemes for the SPH-ALE method. Application in free surface flows in Pelton turbines, in: 4^{th} SPHERIC workshop, Nantes, France. pp. 323 – 330.

Marongiu, J.C., Leboeuf, F., Caro, J., Parkinson, E., 2010. Free surface flows simulations in Pelton turbines using an hybrid SPH-ALE method. Journal of Hydraulic Research 48, 40–49. URL: https://doi.org/10.1080/00221686.2010.9641244, doi:10.1080/00221686.2010.9641244, arXiv:https://doi.org/10.1080/00221686.2010.9641244.

Marongiu, J.C., Leboeuf, F., Parkinson, E., 2007. Numerical simulation of the flow in a Pelton turbine using the meshless method SPH and a new simple solid boundary treatment, in: 7th European Conference on Turbomachinery Fluid Dynamics and Thermodynamics, Athens, Greece.

Marrone, S., Antuono, M., Colagrossi, A., Colicchio, G., Le Touzé, D., Graziani, G., 2011. δ-sph model for simulating violent impact flows. Computer Methods in Applied Mechanics and Engineering 200, 1526–1542.

Mayrhofer, A., Ferrand, M., Kassiotis, C., Violeau, D., Morel, F.X., 2015. Unified semi-analytical wall boundary conditions in SPH: analytical extension to 3-D. Numerical Algorithms 68, 15–34.

Mayrhofer, A., Laurence, D., Rogers, B., Violeau, D., 2014. Large eddy simulation with SPH: Mission impossible, in: Proc. 9th international SPHERIC workshop, pp. 216–222.

Mayrhofer, A., Rogers, B.D., Violeau, D., Ferrand, M., 2012. Study of differential operators in the context of the semi-analytical wall boundary conditions, in: 7th SPHERIC International Workshop, pp. 149–156.

Mayrhofer, A., Rogers, B.D., Violeau, D., Ferrand, M., 2013. Investigation of wall bounded flows using SPH and the unified semi-analytical wall boundary conditions. Computer Physics Communications 184, 2515–2527.

Méchitoua, N., Boucker, M., Laviéville, J., Hérard, J., Pigny, S., Serre, G., 2003. Neptune-CFD: an Unstructured FV Solver for two-phase Water/Vapor Flows Based on an Elliptic Oriented Fractional Step Method. NURETH-10, Seoul, Oct. 5 9.

Minier, J.P., 2016. Statistical descriptions of polydisperse turbulent two-phase flows. Physics Reports 665, 1–122.

Mola, A., Heltai, L., DeSimone, A., 2013. A stable and adaptive semi-Lagrangian potential model for unsteady and nonlinear ship-wave interactions. Engineering Analysis with Boundary Elements 37, 128 – 143. URL: http://www.sciencedirect.com/science/article/pii/S0955799712001907, doi:https://doi.org/10.1016/j.enganabound.2012.09.005.

Monaghan, J., Kajtar, J., 2009a. SPH particle boundary forces for arbitrary boundaries. Computer Physics Communications 180, 1811 – 1820.

Monaghan, J.J., 1992. Smoothed Particle Hydrodynamics. Annual review of Astronomy and Astrophysics 30, 543–574.

Monaghan, J.J., 1994. Simulating free surface flows with SPH. Journal of Computational Physics 110, 399–406.

Monaghan, J.J., 1999. SPH without a tensile instability. Journal of Computational Physics 159, 290–311.

Monaghan, J.J., 2005. Smoothed Particle Hydrodynamics. Reports on Progress in Physics 68, 1703–1759.

Monaghan, J.J., Gingold, R.A., 1983. Shock simulation by the particle method sph. Journal of computational physics 52, 374–389.

Monaghan, J.J., Kajtar, J.B., 2009b. Sph particle boundary forces for arbitrary boundaries. Computer physics communications 180, 1811–1820.

Morris, J.P., Fox, P.J., Zhu, Y., 1997. Modeling Low Reynolds Number Incompressible Flows Using SPH. Journal of Computational Physics 136, 214–226.

Neuhauser, M., Leboeuf, F., Marongiu, J.C., Parkinson, E., Robb, D., 2014. Simulations of Rotor–Stator Interactions with SPH-ALE. Springer Singapore, Singapore. pp. 349–361. URL: https://doi.org/10.1007/978-981-4451-42-0_29, doi:10.1007/978-981-4451-42-0_29.

Oger, G., Doring, M., Alessandrini, B., Ferrant, P., 2007. An improved SPH method: Towards higher order convergence. Journal of Computational Physics 225, 1472 – 1492.

Oger, G., Marrone, S., Le Touzé, D., De Leffe, M., 2016. SPH accuracy improvement through the combination of a quasi-Lagrangian shifting transport velocity and consistent ALE formalisms. Journal of Computational Physics 313, 76–98.

Orlanski, I., 1976. A simple boundary condition for unbounded hyperbolic flows. Journal of Computational Physics 21, 251 – 269. doi:https://doi.org/10.1016/0021-9991(76)90023-1.

Patankar, S.V., 1980. Numerical heat transfer and fluid flow .

Patsis, P.A., Athanassoula, E., 2000. SPH simulations of gas flow in barred galaxies. Effect of hydrodynamical and numerical parameters. aap 358, 45–56.

P.D. Lax, B.W., 1960. Systems of conservation laws. Communications in Pure and Applied Mathematics 13, 217–237.

Pesce, C.P., Tannuri, E.A., Casetta, L., 2006. The Lagrange equations for systems with mass varying explicitly with position: some applications to offshore engineering. Journal of the Brazilian Society of Mechanical Sciences and Engineering 28, 496–504.

Pilliod, J.E., Puckett, E.G., 2004. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. Journal of Computational Physics 199, 465 – 502. URL: http://www.sciencedirect.com/science/article/pii/S0021999104000920, doi:https://doi.org/10.1016/j.jcp.2003.12.023.

Pope, S.B., 2000. Turbulent Flows. Cambridge University Press.

R. Beam, R.W., 1976. An implicit finite-difference algorithm for hyperbolic systems in conservation-law form. Journal of Computational Physics 22.

Re, B., Dobryzynski, C., Guardone, A., 2017. An interpolation-free ALE scheme for unsteady inviscid flows computations with large boundary displacements over three-dimensional adaptive grids. Journal of Computational Physics 340, 26–54.

Relaño Castillo, A., 2012. AxisSPH: devising and validating an axisymmetric smoothed particle hydrodynamics code .

Rhie, C., Chow, W.L., 1983. Numerical study of the turbulent flow past an airfoil with trailing edge separation. AIAA journal 21, 1525–1532.

Robinson, M.J., 2009. Turbulence and viscous mixing using smoothed particle hydrodynamics. Ph.D. thesis. Monash University.

Shao, Y.L., Faltinsen, O.M., 2014. A harmonic polynomial cell (HPC) method for 3D Laplace equation with application in marine hydrodynamics. Journal of Computational Physics 274, 312–332.

Storti, M.A., Garelli, L., Paz, R.R., 2012. A finite element formulation satisfying the discrete geometric conservation law based on averaged Jacobians. International Journal for Numerical Methods in Fluids 69, 1872–1890. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.2669, doi:10.1002/fld.2669, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.2669.

Sun, P., Colagrossi, A., Marrone, S., Zhang, A., 2017. The δplus-SPH model: Simple procedures for a further improvement of the SPH scheme. Computer Methods in Applied Mechanics and Engineering 315, 25–49.

Sun, P.N., Le Touzé, D., Oger, G., Zhang, A.M., 2021. An accurate sph volume adaptive scheme for modeling strongly-compressible multiphase flows. part 2: Extension of the scheme to cylindrical coordinates and simulations of 3d axisymmetric problems with experimental validations. Journal of Computational Physics 426, 109936.

Svyatskiy, D., 2008. Benchmark on Anisotropic Problems.Nonlinear monotone finite volume method, in: Finite Volumes for Complex Applications V , pp. 935–947.

Sweby, P.K., 1984. High resolution schemes using flux limiters for hyperbolic conservation laws. SIAM Journal of Numerical Analysis 21, 995–1011.

Tait, P.G., 1888. Report on some of the physical properties of fresh water and sea water. Report on the Voyage of Challenger, Physics and Chemistry 2, 1–76.

Team, D., 2021. code_saturne 7.0 Theory Guide. EDF R&D. URL: https://code-saturne.org/cms/sites/default/files/docs/7.0/theory.pdf.

Teles, M.J., Pires-Silva, A.A., Benoit, M., 2013. Numerical modelling of wave current interactions at a local scale. Ocean Modelling 68, 72–87.

Temam, R., 1968. Une méthode d'approximation de la solution des équations de navier-stokes. Bulletin de la Société Mathématique de France 96, 115–152.

Thomas, P.D., Lombard, C.K., 1979. Geometric conservation law and its application to flow computations on moving grids. AIAA Journal 17, 1030–1037. doi:10.2514/3.61273, arXiv:https://doi.org/10.2514/3.61273.

Toro, E.F., 2009. Riemann solvers and numerical methods for fluid dynamics. Springer, London.

Toro, E.F., 2013. Riemann solvers and numerical methods for fluid dynamics: a practical introduction. Springer Science & Business Media.

Vacondio, R., Altomare, C., De Leffe, M., Hu, X., Le Touzé, D., Lind, S., Marongiu, J.C., Marrone, S., Rogers, B.D., Souto-Iglesias, A., 2021. Grand challenges for smoothed particle hydrodynamics numerical schemes. Computational Particle Mechanics 8, 575–588.

Vacondio, R., Rogers, B.D., Stansby, P.K., Mignosa, P., 2011. SPH modeling of shallow flow with open boundaries for practical flood simulation. Journal of Hydraulic Engineering 138, 530–541.

Versteeg, H.K., Malalasekera, W., 2007. An introduction to computational fluid dynamics the finite volume method. 2^{nd} ed., Pearson Education Ltd., Harlow (GB) New York Boston [etc.]. URL: http://www.sudoc.fr/11762473X.

Vila, J.P., 1999. On particle weighted methods and smooth particle hydrodynamics. Mathematical Models and Methods in Applied Sciences 9, 161–209.

Violeau, D., 2012. Fluid Mechanics and the SPH Method: Theory and Applications. Oxford University Press. URL: http://ukcatalogue.oup.com/product/academic/earthsciences/hydrology/9780199655526.do.

Violeau, D., Issa, R., 2007. Numerical modelling of complex turbulent free-surface flows with the SPH method: an overview. International Journal for Numerical Methods in Fluids 53, 277–304. URL: http://dx.doi.org/10.1002/fld.1292, doi:10.1002/fld.1292.

Violeau, D., Issa, R., Benhamadouche, S., Saleh, K., Chorda, J., Maubourguet, M.M., 2008. Modelling a fish passage with SPH and Eulerian codes: the influence of turbulent closure, in: Proc. III^{rd} SPHERIC International Workshop, Lausanne, Suisse. pp. 85–91.

Violeau, D., Leroy, A., 2014. On the maximum time step in weakly compressible SPH. Journal of Computational Physics 256, 388–415.

Violeau, D., Leroy, A., Mayrhofer, A., 2014. Exact computation of SPH wall renormalising integrals in 3-D. by D. Violeau, A. Hérault, and A. Joly. Paris , 95–102.

Wall, C., Pierce, C.D., Moin, P., 2002. A semi-implicit method for resolution of acoustic waves in low Mach number flows. Journal of Computational Physics 181, 545–563.

Wang, Y., Hutter, K., 2001. Comparisons of numerical methods with respect to convectively dominated problems. International Journal for Numerical Methods in Fluids .

Wei, G., Kirby, J.T., Grilli, S.T., Subramanya, R., 1995. A fully nonlinear Boussinesq model for surface waves. Part 1. Highly nonlinear unsteady waves. Journal of Fluid Mechanics 294, 71–92.

Wendland, H., 1995a. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. Advances in Computational Mathematics 4, 389–396.

Wendland, H., 1995b. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. Advances in Computational Mathematics 4, 1414–1421.

Werdelmann, B., Koch, R., Krebs, W., Bauer, H.J., 2021. An approach for permeable boundary conditions in sph. Journal of Computational Physics 444, 110562.

Wroniszewski, P.A., Verschaeve, J.C.G., Pedersen, G.K., 2014. Benchmarking of Navier-Stokes codes for free surface simulations by means of a solitary wave. Coastal Engineering 91, 1–17.

X. Zhang, C.S., 2011. Maximum-principle-satisfying and positivity-preserving high order schemes for conservation laws: Survey and new developments. Proceedings of the Royal Society A 471.

Xu, R., 2009. An Improved Incompressible Smoothed Particle Hydrodynamics Method and Its Application in Free-Surface Simulations. Ph.D. thesis. University of Manchester Institute of Science and Technology. Manchester, UK.

Yee, H.C., 1989. A Class of High-Resolution Explicit and Implicit Shock-Capturing Methods. NASA.

Zalesak, S., 1979. Fully multidimensional flux corrected transport algorithms for fluids. Journal of computational physics 31, 335–362.

Zhang, H., Reggio, M., Trépanier, J., Camarero, R., 1993. Discrete form of the GCL for moving meshes and its implementation in CFD schemes. Computers & Fluids 22, 9 – 23. URL: http://www.sciencedirect.com/science/article/pii/004579309390003R, doi:https://doi.org/10.1016/0045-7930(93)90003-R.

Zijlema, M., Stelling, G.S., 2008. Efficient computation of surf zone waves using the nonlinear shallow water equations with non-hydrostatic pressure. Coastal Engineering 55, 780–790.