A dual interface method in cylindrical coordinates for two-phase pipe flows

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op
maandag 13 november 2017 om 15:00 uur

door

Guido Thomas OUD
ingenieur luchtvaart en ruimtevaart

geboren te Spaarndam

This dissertation has been approved by the:

promotors: prof. dr. ir. C. Vuik and prof. dr. ir. R.A.W.M. Henkes
copromotor: dr. ir. D.R. van der Heul

Composition of the doctoral committee:

Rector Magnificus	chairman
prof. dr. ir. C. Vuik	Delft University of Technology, promotor
prof. dr. ir. R.A.W.M. Henkes	Delft University of Technology, promotor
dr. ir. D.R. van der Heul	Delft University of Technology, copromotor

Independent members:

prof. dr. ir. N.G. Deen	Eindhoven University of Technology
prof. dr. S. Hickel	Delft University of Technology
prof. dr. A.E.P. Veldman	University of Groningen
prof. dr. ir. B.G.M. Van Wachem	Imperial College London
prof. dr. ir. C.R. Kleijn	Delft University of Technology (alternate)

The research in this dissertation was funded by Deltares, TNO, Shell and Delft University of Technology.

A dual interface method in cylindrical coordinates for two-phase pipe flows

Dissertation at Delft University of Technology

ISBN 978-94-6295-763-3

Printing and cover design: ProefschriftMaken || www.proefschriftmaken.nl

An electronic version of this dissertation is available at:
`http://repository.tudelft.nl`

Preface

"If you will it, Dude, it is no dream."

— Walter Sobchak, The Big Lebowski

This thesis contains the main results of the PhD project that I have been working on during the past few years. It is the culmination of a life-long interest in the field of mathematics and engineering. In this short preface, I would like to take the opportunity to thank a number of people who have made the realization of this thesis possible. First and foremost, I thank Duncan van der Heul for his guidance and support during this project. Your suggestions during our conversations were often enlightening to me, and you have been a great teacher. Therefore, I look forward to hopefully continuing our scientific cooperation in the future. I thank Kees Vuik and Ruud Henkes, my promotors, for their support and the extensive patience they possess (or must have developed in these years). In particular, I very much appreciated the freedom I was given in occasionally pursuing research topics of my own preference.

Although the process of a PhD project is largely a time of utter solitude, I owe a great deal of appreciation to my roommates for the joyful and helpful distractions they provided. Menel, for keeping us in line, even though I cannot recall the number of times you locked me out of my own office. Manuel, for our shared interests and being an excellent host when I needed a place to spend the night when the last train had left. And Joost, for our shared opinions, whiskey-inspired evenings, and your help in converting me from an experienced Windows user into a Linux noob.

I have been fortunate to work with and alongside fellow MSc- and PhD candidates who made the working environment a pleasant place to be. I enjoyed supervising Coen Hennipman and Ankit Mittal, who extended my work and were able to demonstrate a number of very nice applications.

Most of the evenings at the conferences in Darmstadt and Florence were spent with Fabian Denner, Berend van Wachem and the students of the multiphase group of Imperial College. They were kind enough to let me tag along on those occasions, for which I thank them sincerely.

I would like to thank the industrial partners Deltares, Shell and TNO for their financial support and the suggestions they provided during the regular progress meetings for my project.

I thank my parents and siblings for their love and their never-fading trust throughout this PhD endeavour. Your words of support have helped me through the darker periods and my appreciation cannot be captured in words.

And finally, above all, I thank Esther. You have had to bear so many lonely evenings and weekends while I was trying to get yet another piece of code to work. It takes a special person to endure such times of solitude, and nonetheless remain caring and supportive wholeheartedly. Never forget that you are all my reason...

Guido Oud
Zeist, May 2017

Contents

Chapter 1

Introduction

The numerical modeling of multiphase flow phenomena has been a field of great interest within the computational fluid dynamics community for the past few decades. Examples of multiphase flows in industry and engineering are ubiquitous: boilers, condensers, reactors, mixers, separators and cavitation are but some of the applications. In present day hydrocarbon recovery, the increased effort to produce from fields that are located more remotely and in deeper water has led to more complex transport of oil and gas. One often finds that both phases are transported through a single pipeline to an offshore platform or to an onshore plant. The liquid-gas mixture that is flowing through a pipeline can be in a specific flow regime, as will be further explained in section 1.1. The prevailing flow regime depends on, among others, the liquid and gas flow rates, the pipeline pressure, and the pipe diameter and inclination. Highly unstable flows with slugs can cause significant mechanical stresses on the pipe wall in bends and on its supports. Additionally, unstable flow can create difficulties for the facilities located at the downstream end of the pipeline, such as flooding of the separator or trips of the compressor or pump. Such facilities would perform best with a constant supply of gas and liquid over time. If there is unstable flow, extra capacity should be reserved in the facilities to accommodate the liquid and gas surges, or special equipment with controllers (such as actuated choke valves) needs to be installed to prevent slugs or to mitigate the effects of slugs. For these reasons, there is a clear interest from the industry in numerical tools that are able to accurately simulate two-phase flows in horizontal, inclined, and vertical pipes. Such tools can be very useful in the prediction of the flow conditions that ultimately lead to flow instabilities. The tools can be used to design the pipeline and its operations, and they can tell whether prevention or mitigation measures need to be taken for the liquids management.

1.1 Multiphase flow regimes

The behaviour of the flow of two (or more) immiscible fluids through horizontal circular pipes can be classified into a number of flow regimes. Common variables used in this classification are the flow rates of the phases (often represented by the superficial velocity, which is the ratio of the volumetric flow rate and the cross section of the pipe). The fraction of one fluid in the total (combined) volumetric flow rate is referred to as the (volumetric) quality. A simplified classification of the flow regimes encountered in horizontal pipe flow is shown in figure 1.1.

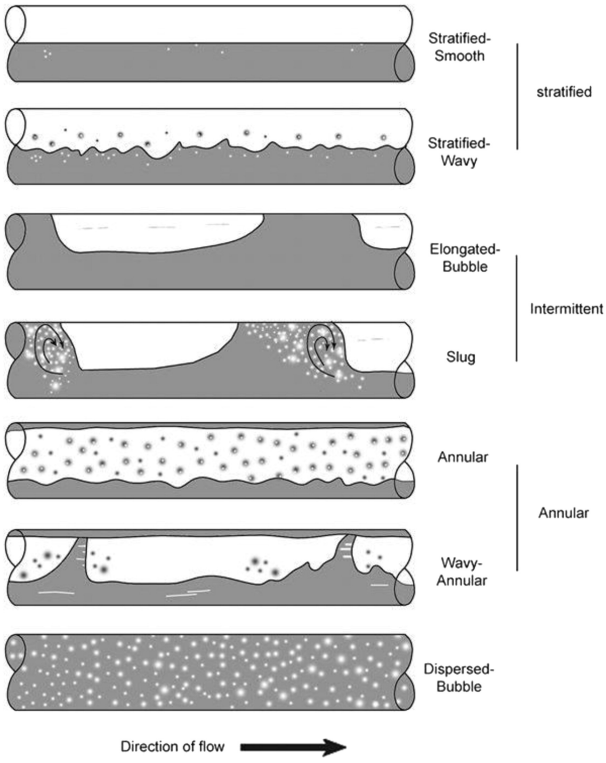


Figure 1.1: Various multiphase flow regimes in horizontal pipe flows. Source: Najmi et al. [63].

For low flow rates of both fluid phases, the presence of gravity leads to the *stratified* flow regime with the lighter phase on top of the heavier phase. As the difference between the bulk velocities of the fluids increases, waves with growing amplitude start to form at the planar interface; this phenomenon is known as a Kelvin-Helmholtz instability of the interface. Eventually, for a sufficiently large velocity difference, the waves reach the upper cylinder wall, and the heavier fluids starts to completely occupy the pipe cross section. The local liquid blockage of the pipe leads to the

so-called *plug-* and *slug flow* regimes. If the gas throughput is further increased the *annular dispersed* flow regime is found. The gas core carries liquid droplets (giving dispersed gas/liquid flow) and there is a liquid film along the wall. Due to gravity, the film along the bottom wall is thicker than along the top wall. If the throughput of the liquid becomes large (and the gas quantity is low or moderate), the *dispersed bubble* flow regime is found. Here liquid forms the continuous phase and it carries the gas bubbles.

For vertical or inclined pipes, similar flow regimes exist, but the transition boundaries are different because gravity now acts at a different angle along the pipe direction. It must be stressed that in practice, besides the superficial velocities, also the fluid properties (density, viscosity, surface tension) and the pipe diameter and the wall roughness are factors that influence the selection of the flow regime. For the pipeline design and operation it is important to know what the flow regime is, as each flow regime will give a specific pressure drop along the pipeline and a specific holdup fraction or volume fraction (which is the fraction of the pipe volume that consists of either liquid or gas). Also the specific dynamics, such as the slug frequency and the slug length, are of importance.

1.2 Computational multiphase flow modeling

The numerical modeling of multiphase flows is a challenging area within the field of computational fluid dynamics. For the engineering design of multiphase flow in pipeline systems, the industry often uses one-dimensional models. This means that the three-dimensional flow equations are averaged over the pipe cross-sectional area, leaving a one-dimensional spatial problem, in which the pipe axis is used as the remaining single coordinate. Here the gas and liquid velocity profiles over the cross section are represented by the bulk gas velocity and the bulk liquid velocity. The averaging leads to closure terms, for which correlations are used, such as for the wall friction, for the interfacial friction, for the gas bubble entrainment in the liquid, and for the liquid droplet entrainment in the gas. There exist also correlations for the transition boundaries between the flow regimes. Both steady state and transient one-dimensional models are used. Some well-known commercial one-dimensional tools that are used in the industry are OLGA and LedaFlow. The main advantage of one-dimensional models is that they are computationally fast, but a disadvantage is that they heavily rely on empirical closure relations. This disadvantage becomes

particularly important in the prediction of slug flow, as the slug formation and its characteristics are inherently multi-dimensional. Therefore the present thesis is focused on the development of two- and three-dimensional CFD models for multiphase flow in pipelines.

Building on proven CFD concepts for single phase flows, various specialized multidimensional algorithms exist that, in principle, are also applicable to multiphase flow, depending on, for example:

- incompressible vs compressible flow.
- laminar vs turbulent flow.
- miscible vs immiscible fluids.
- isothermal vs non-isothermal flows.
- sharp vs diffusive interfaces.

The research described in this thesis is particularly focused on incompressible isothermal flow of immiscible fluids. With the aim of simulating the transport of hydrocarbon fluids, it is important to recognize the impact of these simplifications. For example, compressibility is important in the transport of gas through long pipe lines; this is because the decreasing pressure along the pipeline will cause a certain mass flow rate which experiences a decreased density and therefore and increased velocity. Thermal effects are also important for long pipelines, due to heat transfer between the fluids in the pipe and the ambient and due to the gas expansion. However, the mentioned assumptions simplify the numerical modeling considerably: an advantage of assuming isothermal conditions, for example, is that there is no need to solve the additional energy equation, which reduces the computational effort. Recognizing the limitations of the proposed model, the assumptions mentioned are considered an acceptable premise as an initial step towards a more complete and physically accurate algorithm. Furthermore, the assumptions made are a good approximation for relatively short pipe sections.

The physical scale of the conditions that we want to simulate is large enough to allow for a fully sharp representation of the interface between the immiscible fluids. Proven numerical methods are readily available for the sharp interface treatment. The Level Set (LS) method [66] and the Volume of Fluid (VoF) method [41] are two well-known approaches. They belong to a class known as *Eulerian front-capturing*

methods, in which the interface is represented similar to a contact discontinuity on a fixed computational grid that remains stationary in time. Contrary to *front-tracking* techniques, the interface is not explicitly reconstructed, but it is implicitly defined through an indicator function. Once defined at the initialization, the interface is subsequently advected with the flow field and captured at any later time instant using specially designed algorithms. A detailed description of both the Level Set- and VoF method can be found in section 4.2.2. These methods have been applied very successfully to the simulation of two-phase flows, and experience has shown both their advantages and disadvantages in practical use. The Level Set method is relatively simple to implement, computationally efficient and deals with complex topology changes in a natural way. It is, however, inherently incapable of conserving the liquid and gas volume over time, as this is not an enforced property of the method. This implies that the individual phase volumes may change as time progresses, while the incompressibility of the flow dictates that they should each remain constant. The VoF method, on the other hand, is volume conservative by construction, but it lacks an explicit interface, which leads to complicated and computationally expensive interface reconstructions as it only provides information on the quantity (or volume fraction) in a computational cell but not on its precise location in the cell. Although the impact of most of these drawbacks can be reduced to a certain extent by using advanced techniques, a more recent hybrid approach combines the Level Set and VoF methods with the aim of benefiting from their advantages (the straightforward interface representation of the Level Set method and the conservation property of the VoF method) while minimizing their disadvantages. The general approach in these *coupled* or *dual interface* methods is to combine a flow solver with an interface model that is based on a specifically tuned combination of the Level Set method and the VoF method. The conceptual differences are often small, and the discrepancies mostly lie in the details of the interaction between both methods. Sub-algorithms of the dual interface model may be based on either the Level Set method or the VoF method depending on preference.

A variety of dual interface methods have been proposed, for example in Bourlioux [13], Sussman and Puckett [83] (CLSVOF), Van der Pijl et al. [90] (MCLS), Yang et al. [96] (ACLSVOF) and Sun and Tao [81] (VOSET). In Van der Pijl et al. [90], the Mass-Conserving Level Set (MCLS) method was introduced as a coupled method on a uniform Cartesian grid. A key feature of this method is the use of an analytic function that calculates the VoF value of a computational cell for a given Level Set

function and its gradient. The MCLS method was shown to be robust and to yield accurate results when compared to similar methods.

Due to the success of the MCLS method on a Cartesian grid, the core concept of this method was also used in the present thesis to form the basis for a newly designed interface algorithm in 3D cylindrical coordinates intended for the simulation of flow instabilities in circular pipe sections. A detailed description of the developed algorithm can be found in section 3.3.1, and the numerical coupling between the Level Set method and the VoF method is explained in section 3.3.2 (2D axisymmetric) and section 4.2.3 (3D cylindrical).

Current multiphase pipe flow simulations with CFD seem to favour the use of unstructured grids to capture curved walls and pipe bends. Experience, however, shows that the application of interface capturing methods on such grids is generally cumbersome, and high resolution interfaces are hard to obtain. In particular, conservative advection of the interface is notoriously difficult and computationally demanding compared to advection on structured grids.

When using a structured grid for the simulation of multiphase flow in a pipeline, one can either think of a Cartesian grid or a grid in cylindrical coordinates. The Cartesian grid has the large advantage to give a relatively simple (and well known) discretization of the flow and the interfaces, but introduces some complexity when trying to capture the curved fixed wall. This requires the use of the Immersed Boundary Method (IBM) or the ghost cell method. A very fine grid will be required to capture the boundary layers along the walls. As an alternative, using a cylindrical grid can be considered. This has the advantage that it is fitted to the fixed boundary walls, and it can thus naturally capture the boundary layers along the walls. Disadvantages are the complexity of deriving a discretization of the flow and of the interface, and the presence of a singularity in the grid at the centerline of the pipe. We believe, however, that the use of cylindrical coordinates for multiphase pipe flow has considerable potential, and deserves further exploration: this will be the focus of the present thesis.

1.3 Research objective

The main research objective of the present thesis is to obtain an improved numerical tool that is able to carry out relatively fast, robust, and accurate simulations of physical, two-phase flow instabilities in straight pipe sections. To that end, a dedicated numerical two-phase flow algorithm in cylindrical coordinates will be designed and implemented in Fortran 90. The algorithm is to be used as a numerical test bed in the study of the development of 3D instabilities in pipe flows. The considered flow examples will be all laminar, but the same methods also need to be applicable in a later stage to turbulent multiphase flows using either RANS, LES or DNS. The desire of applicability to turbulent flows places an important condition on the algorithm, as it has to be sufficiently accurate (at least second order and preferably extendable to higher orders) and computationally efficient (well scalable). To accommodate these requirements, the computational domain is restricted to a segment of a straight cylindrical pipe with a certain inclination. In this way, both the computational performance and the numerical accuracy can be optimized, albeit at the cost of less flexibility in choosing the domain. Keeping the geometry fixed allows the use of a structured orthogonal grid that generally leads to a higher order of accuracy than non-structured grids. Additionally, fast and efficient solvers are readily available for structured tessellations.

The literature only contains a very limited number of studies that apply either the Level Set method or the VoF method to the generalized case of curvilinear coordinates [30, 45, 46]. A plausible reason for the lack of studies using structured grids is the presence of a singularity in the cylindrical coordinate system. It is well known that this singularity is a source of numerical problems even in the modeling of single-phase flows. Part of the research described in the present thesis was therefore devoted to analyze, mitigate and possibly solve the problems due to the coordinate singularity in both the flow field model and in the interface model.

Once a suitable model has been established for the flow and for the interface, verification and validation need to be performed. Where possible, verification is done by using the Method of Manufactured Solutions [74]. The validation is done by using well known benchmark tests, such as stationary or rising bubbles. Furthermore, comparisons are made with theoretical predictions if the conditions are such that these apply.

1.4 Outline

This thesis consists of three main research parts.

In the first part (described in chapter 2), the emphasis lies on the modeling of the flow field using the Navier-Stokes equations in cylindrical coordinates. A mimetic approach is used to rigorously derive the finite difference approximations of the differential operators. This approach guarantees that important vector identities hold at the discrete level as well, thereby intrinsically capturing the behaviour of the continuous solutions. A second order finite difference discretization for non-uniform grids is subsequently derived, and the conservation properties are extensively studied using numerical examples. A consistent treatment of the coordinate singularity follows naturally and it is thereafter no longer considered a problem.

In the second part (described in chapter 3), the interface algorithm is presented in the case of rotational symmetry. A detailed description of the algorithm is provided, and various verification tests are performed to demonstrate the accuracy of the proposed model. Validation is done by comparison with results from other software tools (both commercial and open-source), and by using some well-known multiphase benchmark cases.

The third part (described in chapter 4) contains the numerical study of two-phase flow instabilities. The numerical issues in the interface transport due to the singularity are analyzed and mitigated, and a solution strategy is proposed. The ability of the algorithm to capture Kelvin-Helmholtz waves in shear flows is then studied for both core-annular flow and for stratified flow.

Chapter 2

A fully conservative mimetic discretization of the Navier-Stokes equations in cylindrical coordinates with associated singularity treatment

We present a finite difference discretization of the incompressible Navier-Stokes equations in cylindrical coordinates. This currently appears to be the only scheme available that is demonstrably capable of conserving mass, momentum and kinetic energy (in the absence of viscosity) on both uniform and non-uniform grids. Simultaneously, we treat the inherent discretization issues that arise due to the presence of the coordinate singularity at the polar axis. We demonstrate the validity of the conservation claims by performing a number of numerical experiments with the proposed scheme, and we show that it is second order accurate in space using the Method of Manufactured Solutions.

2.1 Introduction

Although it is generally known that the use of cylindrical coordinates in finite difference methods brings along a number of difficulties, it still appears to be the preferred method of choice for turbulent flow simulations in pipe sections. This is likely due to the relative ease with which higher order approximations can be

The content of this chapter is based on the article:

G.T. Oud, D.R. van der Heul, C. Vuik, and R.A.W.M. Henkes. *A fully conservative mimetic discretization of the Navier-Stokes equations in cylindrical coordinates with associated singularity treatment*, Journal of Computational Physics, 325: 314-337, 2016.

implemented, and the growing availability of fast flow solvers that benefit from the orthogonality of the structured cylindrical grid. However, an inherent problem in the use of cylindrical coordinates (r, θ, z) is the calculation of variables that lie on or near the polar axis $r = 0$. Looking at the cylindrical Navier-Stokes equations:

$$\frac{1}{r}\frac{\partial(ru_r)}{\partial r} + \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} + \frac{\partial u_z}{\partial z} = 0, \tag{2.1}$$

and:

$$\begin{aligned}\frac{\partial u_r}{\partial t} + \frac{1}{r}\frac{\partial(ru_r^2)}{\partial r} + \frac{1}{r}\frac{\partial(u_r u_\theta)}{\partial \theta} + \frac{\partial(u_r u_z)}{\partial z} - \frac{u_\theta^2}{r} = \\ -\frac{1}{\rho}\frac{\partial p}{\partial r} + \frac{1}{\rho r}\frac{\partial(r\tau_{rr})}{\partial r} + \frac{1}{\rho r}\frac{\partial \tau_{r\theta}}{\partial \theta} + \frac{1}{\rho}\frac{\partial \tau_{rz}}{\partial z} + g_r,\end{aligned} \tag{2.2}$$

$$\begin{aligned}\frac{\partial u_\theta}{\partial t} + \frac{1}{r}\frac{\partial(ru_r u_\theta)}{\partial r} + \frac{1}{r}\frac{\partial(u_\theta^2)}{\partial \theta} + \frac{\partial(u_\theta u_z)}{\partial z} + \frac{u_r u_\theta}{r} = \\ -\frac{1}{\rho r}\frac{\partial p}{\partial \theta} + \frac{1}{\rho r}\frac{\partial(r\tau_{r\theta})}{\partial r} + \frac{1}{\rho r}\frac{\partial \tau_{\theta\theta}}{\partial \theta} + \frac{1}{\rho}\frac{\partial \tau_{\theta z}}{\partial z} + \frac{\tau_{r\theta}}{\rho r} + g_\theta,\end{aligned} \tag{2.3}$$

$$\begin{aligned}\frac{\partial u_z}{\partial t} + \frac{1}{r}\frac{\partial(ru_r u_z)}{\partial r} + \frac{1}{r}\frac{\partial(u_\theta u_z)}{\partial \theta} + \frac{\partial(u_z^2)}{\partial z} = \\ -\frac{1}{\rho}\frac{\partial p}{\partial z} + \frac{1}{\rho}\frac{\partial(r\tau_{rz})}{r\partial r} + \frac{1}{\rho r}\frac{\partial \tau_{\theta z}}{\partial \theta} + \frac{1}{\rho}\frac{\partial \tau_{zz}}{\partial z} + g_z,\end{aligned} \tag{2.4}$$

with $\mathbf{u} = (u_r, u_\theta, u_z)$ the velocity vector, ρ the flow density, $\mathbf{g} = (g_r, g_\theta, g_z)$ the body force and τ_{ij} the viscous stresses, it would indeed appear that the numerous $1/r$ terms cause the solution to blow up near the polar axis, but, as shown in Morinishi et al. [61], the coordinate singularity is only apparent, and taking rigorous limits shows how the equations actually behave at $r = 0$. From a numerical modeling point of view, however, this asymptotic analysis does not provide a clear solution to the singularity problem. Assuming a staggered (Marker-and-Cell) grid [38], the straightforward finite difference discretization of the Navier-Stokes equations in conservative form requires (among others) the radial velocity u_r at $r = 0$. In the past, this value was usually estimated in some way using neighbouring values. For example, Eggels [28] applied an arithmetic mean for the estimate of $u_r(0, \theta_j, z_k)$ using two opposite values:

$$u_r(0, \theta_j, z_k) = \frac{u_r(r_1, \theta_j, z_k) - u_r(r_1, \theta_j + \pi, z_k)}{2}, \tag{2.5}$$

where the minus sign is necessary because of the orientation (both outwards) of the velocity vectors. This approach, however, yields a multivalued radial velocity. An

improved approximation by Fukagata and Kasagi [33] and Griffin et al. [36] leads to a single-valued radial velocity; this was done by reconstructing the Cartesian velocity components u_x and u_y at $r = 0$ from the neighbouring set of velocities and by defining u_r at $r = 0$ using the decomposition:

$$u_r(0, \theta, z) = u_x \cos\theta + u_y \sin\theta. \tag{2.6}$$

A different approach by Verzicco and Orlandi [92] avoids the problem entirely by solving the equations for the quantity ru_r instead of the velocity u_r. For a more extensive overview of existing methods, see Morinishi et al. [61].

Recently, discretizations with improved conservation properties have obtained a growing interest as physically reliable modelling of the finest structures in turbulence requires accurate numerical behaviour of the flow energy. In particular, the construction of numerical schemes that conserve kinetic energy for flows with vanishing viscosity has become an active field of research. At a discrete level, conservation of kinetic energy is an attractive property as it assures an unconditionally stable (spatial) discretization. For Cartesian domains, this has led to a number of schemes that conserve both mass, momentum and kinetic energy on uniform and non-uniform grids, for both low and higher order [60, 91]. For cylindrical grids, not much progress seems to have been made on these aspects. Fukagata and Kasagi [33] suggest a highly conserving discretization, but energy is not conserved exactly. Morinishi et al. [61] introduced a new approach where a radial momentum equation is solved at $r = 0$ after its derivation using l'Hôpital's rule. The authors claim and prove the scheme to be energy conserving in the absence of viscosity for both uniform and non-uniform grids. After a number of tests, however, we noticed disturbances near the radial origin in flows with significant velocity through the origin. Desjardins et al. [23] mention this as well while performing the simulation of an inviscid Lamb-dipole. Furthermore, using a Taylor expansion of the radial momentum equation at the origin, they are able to trace down the problem to an inconsistent discretization, and they ultimately decide to use an averaging method similar to equation 2.6 for improved accuracy, thereby sacrificing exact energy conservation. None of the mentioned singularity treatments above, except for the one of Morinishi et al. [61], appears to conserve energy as they rely on (arithmetic) averaging in obtaining the radial velocity at $r = 0$. Hence, the problem of finding an energy conserving scheme for cylindrical coordinates seems to intrinsically contain the necessity for a satisfactory treatment of the singularity at $r = 0$.

Most of the methods described above use the notion of a *computational* or *logical* Cartesian space and the *physical* cylindrical space, connected through a Jacobian mapping, to solve the governing equations. The problems involving a radial velocity component at the polar axis arise because this mapping is not bijective at the coordinate $r = 0$. Nonetheless, many attempts have been made to derive expressions that include (the inverse of) the Jacobian, an approach we believe to likely be ill-fated. Instead, we propose a discretization of the Navier-Stokes equations using a mimetic method that is applied on the cylindrical grid (i.e. in physical space only). Mimetic discretizations [53] are designed to mimic many of the properties of the analytical operators they approximate. They have shown to be very robust and accurate, but so far they have been surprisingly rarely used in numerical flow modelling. Abba and Bonaventura [1] derive a mimetic finite difference discretization of the Navier-Stokes equations in Cartesian coordinates, while Barbosa and Daube [6] consider cylindrical coordinates. The latter authors have essentially laid the foundations on which we will proceed. They show how the mimetic operators are derived, albeit for uniform cylindrical grids only. However, their averaging procedure seems to require velocity- and vorticity components at locations where they are not defined. It is therefore unlikely that their discretization conserves energy exactly. We will show how to remedy this issue, and then extend the discretization to grids with non-uniform radial and axial node distributions for increased efficiency in turbulence simulations. Using the method of Manufactured Solutions, we show that the resulting discretization is capable of achieving second order accuracy in space and, with a suitable time integration method, is capable of conserving mass, momentum and kinetic energy for both uniform and non-uniform cylindrical grids.

To our knowledge, the proposed method is currently the only approach in the literature that is demonstrated to be fully conservative for cylindrical coordinates. Most related research in this field appears to be focused on increasing spatial accuracy, while we believe that a solid singularity treatment, together with improved conservation properties, should be established first before moving on to increasing global accuracy.

2.2 Construction of the mimetic operators

Whereas traditional finite difference methods generally focus on minimizing the truncation error, the mimetic finite difference method aims to mimic certain prop-

erties of the continuous operators at a discrete level. Nonetheless, results often show that the accuracy and robustness are nearly as good if not better than conventional discretization techniques. The motivation to use the mimetic method stems from the observation that many mathematical descriptions of physical processes contain the vector derivatives gradient, curl and divergence: consider for example Darcy's law of porous media flow, Maxwell's laws of electromagnetism and the Navier-Stokes equations of fluid flow. The vector derivatives satisfy some well known identities like $\mathbf{curl}\,\mathbf{grad} = 0$ and $\mathbf{div}\,\mathbf{curl} = 0$ for scalars and vectors, respectively, as well as a number of decomposition- and integration by parts theorems. The aim of the mimetic approach is to construct a discrete approximation of the analytical vector calculus by defining discrete vector spaces, inner products and operators, such that the aforementioned identities also hold at the discrete level. In that way, a discrete solution is guaranteed to exhibit many of the underlying properties of the analytical solution.

The approach we follow is based on the work of Hyman and Shashkov [44] and Hyman and Shashkov [43], where suitable discrete vector spaces, inner products and derivatives are derived for orthogonal coordinate systems using the finite difference method. Most of the derivations for uniform cylindrical coordinates have already been performed by Barbosa and Daube [6]. We will extend their work by providing the associated expressions for cylindrical grids with non-uniform radial and axial node distributions. An essential result of the rigorous derivation of the mimetic operators is the absence of any problems in the construction of finite difference approximations around the polar axis that are normally encountered when trying to discretize expressions at or near $r = 0$.

We start in section 2.2.1 by defining the discrete vector spaces, depending on the location of the variables on the MAC grid. Then the metric-independent *natural* vector operations are derived in section 2.2.2. These provide a limited set of discrete mappings between the discrete vector spaces, but without inverse. The inverse mappings require additional metric data in the form of inner products, which are defined in section 2.2.3. Finally, the *adjoint* operators are constructed in section 2.2.4 as the formal adjoints to the natural operators with respect to the associated inner products.

2.2.1 Definition of the discrete vector spaces

We start by defining a number of discrete vector spaces based on the location of the variables in the computational grid. We assume a non-uniform radial and axial distribution such that $\Delta r = \Delta r_i$ is a function of the radial index i and $\Delta z = \Delta z_k$ is a function of the axial index k. The angular distribution (identified with index j) is assumed to be uniform. A location in the computational grid is represented by a coordinate with multi-index $\mathbf{I} = (i, j, k) \in \mathbb{N}^3$ in combination with the stride vectors $\mathbf{e}_r$, $\mathbf{e}_\theta$ and $\mathbf{e}_z$ defined as:

$$\mathbf{e}_r = \left(\frac{1}{2}, 0, 0\right), \quad \mathbf{e}_\theta = \left(0, \frac{1}{2}, 0\right), \quad \mathbf{e}_z = \left(0, 0, \frac{1}{2}\right). \tag{2.7}$$

The pressure $p_\mathbf{I}$ is located at the cell centre $\mathbf{r}_\mathbf{I} = (r_i, \theta_j, z_k)$ with coordinates:

$$r_i = \sum_{l=1}^{i} \Delta r_l - \frac{\Delta r_i}{2}, \qquad i \in \{1, \ldots, N_r\}, \tag{2.8}$$

$$\theta_j = \left(j - \frac{1}{2}\right) \Delta\theta, \qquad j \in \{1, \ldots, N_\theta\}, \tag{2.9}$$

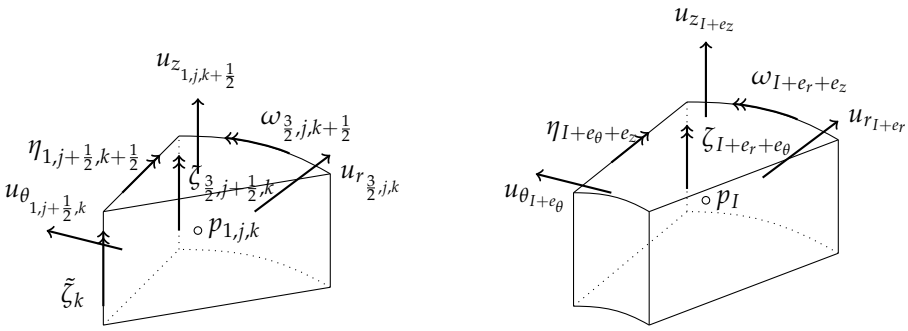
$$z_k = -\frac{L}{2} + \sum_{l=1}^{k} \Delta z_l - \frac{\Delta z_k}{2}, \qquad k \in \{1, \ldots, N_z\}, \tag{2.10}$$

where L is the length of the cylinder. For evaluation at the radial cell boundaries, we define $r_{i\pm\frac{1}{2}}$ as $r_{i\pm\frac{1}{2}} = r_i \pm \frac{\Delta r_i}{2}$. The velocity components $u_{r_{I+e_r}}$, $u_{\theta_{I+e_\theta}}$ and $u_{z_{I+e_z}}$ lie orthogonal to the positive cell faces, and the vorticity components $\eta_{I+e_\theta+e_z}$, $\omega_{I+e_r+e_z}$ and $\zeta_{I+e_r+e_\theta}$ lie along the positive cell edges. Figure 2.1 shows the location of the scalars and vector components in cylindrical coordinates as they are used in this paper. Note the particular axial vorticity $\zeta_{\frac{1}{2},j,k} = \tilde{\zeta}_k$ located at $r = 0$ and independent of angular index j.

For clarity, we use the same notation as in Hyman and Shashkov [44], with calligraphic letters for spaces with vectors and plain letters for spaces with scalars. The following spaces are used:

- The space $\mathcal{HS}$ of discrete vector functions with components that are defined perpendicular to the cell faces at locations $\mathbf{r}_{\mathbf{I}\pm\mathbf{e}_r}$, $\mathbf{r}_{\mathbf{I}\pm\mathbf{e}_\theta}$ and $\mathbf{r}_{\mathbf{I}\pm\mathbf{e}_z}$ in the domain. The discrete velocity $\mathbf{u} = (u_r, u_\theta, u_z)$ on a staggered grid belongs to this space.

- The space $\mathcal{HL}$ of discrete vector functions with components that are defined on the cell ribs at locations $\mathbf{r}_{\mathbf{I}\pm\mathbf{e}_\theta\pm\mathbf{e}_z}$, $\mathbf{r}_{\mathbf{I}\pm\mathbf{e}_r\pm\mathbf{e}_z}$ and $\mathbf{r}_{\mathbf{I}\pm\mathbf{e}_r\pm\mathbf{e}_\theta}$ in the domain. The discrete vorticity $\boldsymbol{\omega} = (\eta, \omega, \zeta)$ belongs to this space.



(a) Cells near the axis at (r_1, θ_j, z_k).

(b) Cells away from the axis at (r_i, θ_j, z_k).

Figure 2.1: Location of the variables in the computational domain.

- The space HC of discrete scalar functions that are defined in the cell centres at locations $\mathbf{r_I}$. The discrete pressure p belongs to this space.
- The space HN of discrete scalar functions that are defined in the cell vertices at locations $\mathbf{r_{I\pm e_r\pm e_\theta\pm e_z}}$, $\mathbf{r_{I\pm e_r\pm e_\theta\pm e_z}}$ and $\mathbf{r_{I\pm e_r\pm e_\theta\pm e_z}}$ in the domain. This space is mentioned for completeness, but it is not used in our approach.

2.2.2 Definition of the natural vector operations

In Hyman and Shashkov [44], expressions are derived for the discrete divergence **D**, the discrete gradient **G** and the discrete curl **C**. They are defined based on a discrete approximation of their coordinate-independent definitions, i.e:

$$\nabla \cdot \mathbf{W} := \lim_{V \to 0} \frac{\oint_{\partial V} (\mathbf{W}, \mathbf{n}) \, \mathrm{d}S}{V}, \tag{2.11}$$

$$(\nabla u, \mathbf{n}) := \frac{\partial u}{\partial n}, \tag{2.12}$$

$$(\mathbf{n}, \nabla \times \mathbf{W}) := \lim_{S \to 0} \frac{\oint_l (\mathbf{W}, \mathbf{l}) \, \mathrm{d}l}{S}, \tag{2.13}$$

for some volume V with boundary ∂V, normal vector $\mathbf{n}$ and surface S with boundary l. The divergence operator **D** is the natural[†] mapping $\mathbf{D} : \mathcal{HS} \to HC$. The gradient operator **G** is the natural mapping $\mathbf{G} : HN \to \mathcal{HL}$, while the curl operator **C** is the natural mapping $\mathbf{C} : \mathcal{HL} \to \mathcal{HS}$. Combined, the natural operators **G**, **C**

[†]The mappings are considered natural because the location of the variables on the staggered grid allows straightforward discrete evaluations of their analytical definitions, i.e. Gauss' theorem (for the divergence) and Stokes' theorem (for the curl).

and **D** form the sequence:

$$HN \xrightarrow{\mathbf{G}} \mathcal{HL} \xrightarrow{\mathbf{C}} \mathcal{HS} \xrightarrow{\mathbf{D}} HC. \tag{2.14}$$

The construction of the natural operators **D** and **C** is shown below (for our purposes, we do not need **G**). The resulting operators satisfy (among others) the well known vector identities (for proofs of this, see Hyman and Shashkov [44]):

$$\mathbf{DC} : \mathcal{HL} \to HC, \quad \mathbf{DC} \equiv 0, \qquad \mathbf{CG} : HN \to \mathcal{HS}, \quad \mathbf{CG} \equiv 0. \tag{2.15}$$

Notice that the resulting expressions, although derived from mimetic principles, are in fact often equal to the classical finite difference discretizations of the gradient, curl and divergence operators in cylindrical coordinates at the computational cell centres. This shows that a mimetic approach does not necessarily lead to different discretization results, but it does provide additional insight and motivation.

The divergence operator D : $\mathcal{HS} \to HC$ The natural divergence operator **D** : $\mathcal{HS} \to HC$ follows from the coordinate-independent formulation of Gauss' divergence theorem:

$$\nabla \cdot \mathbf{W} := \lim_{V \to 0} \frac{1}{V} \oint_{\partial V} (\mathbf{W}, \mathbf{n}) \, \mathrm{d}S, \tag{2.16}$$

where $\mathbf{n}$ is the unit outward normal to the boundary ∂V and $\mathbf{W} : \mathbb{R}^n \to \mathbb{R}^n$ is a differentiable vector field. For a cylindrical cell away from the axis $r = 0$, the discrete approximation $\mathbf{Du}$ to equation 2.16 becomes:

$$\begin{aligned}(\mathbf{Du})_{\mathbf{I}} = \frac{1}{r_i \Delta r_i \Delta\theta \Delta z_k} \Big[& \Delta\theta \Delta z_k (r_{i+\frac{1}{2}} u_{r_{\mathbf{I}+\mathbf{e}_r}} - r_{i-\frac{1}{2}} u_{r_{\mathbf{I}-\mathbf{e}_r}}) + \Delta r_i \Delta z_k (u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}} - u_{\theta_{\mathbf{I}-\mathbf{e}_\theta}}) \\ & + r_i \Delta r_i \Delta\theta (u_{z_{\mathbf{I}+\mathbf{e}_z}} - u_{z_{\mathbf{I}-\mathbf{e}_z}}) \Big] \end{aligned} \tag{2.17}$$

$$= \frac{r_{i+\frac{1}{2}} u_{r_{\mathbf{I}+\mathbf{e}_r}} - r_{i-\frac{1}{2}} u_{r_{\mathbf{I}-\mathbf{e}_r}}}{r_i \Delta r_i} + \frac{u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}} - u_{\theta_{\mathbf{I}-\mathbf{e}_\theta}}}{r_i \Delta\theta} + \frac{u_{z_{\mathbf{I}+\mathbf{e}_z}} - u_{z_{\mathbf{I}-\mathbf{e}_z}}}{\Delta z_k}, \tag{2.18}$$

for some discrete vector $\mathbf{u} = (u_r, u_\theta, u_z) \in \mathcal{HS}$. Notice that this result corresponds with the classical finite difference approximation of the cylindrical divergence:

$$\frac{1}{r} \frac{\partial (r u_r)}{\partial r} + \frac{1}{r} \frac{\partial u_\theta}{\partial \theta} + \frac{\partial u_z}{\partial z}. \tag{2.19}$$

For cells at the polar axis with $i = 1$ where $r_{\frac{1}{2}} = 0$, the discrete operator reduces

without issues to:

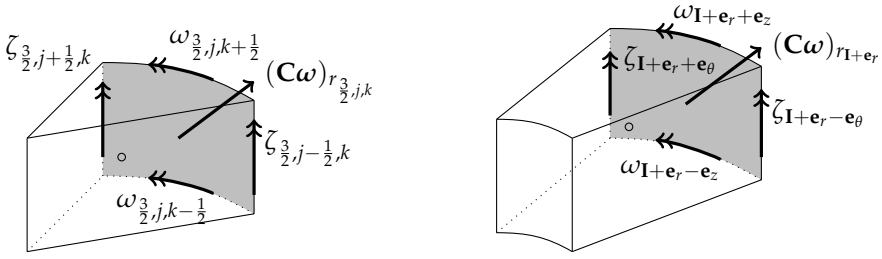
$$(\mathbf{Du})_{\mathbf{I},i=1} = \frac{2u_{r_{\mathbf{I}+\mathbf{e}_r}}}{\Delta r_i} + \frac{2\left(u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}} - u_{\theta_{\mathbf{I}-\mathbf{e}_\theta}}\right)}{\Delta r_i \Delta\theta} + \frac{u_{z_{\mathbf{I}+\mathbf{e}_z}} - u_{z_{\mathbf{I}-\mathbf{e}_z}}}{\Delta z_k}, \tag{2.20}$$

where on the right-hand side the i-component in the multi-index $\mathbf{I}$ is equal to 1.

The curl operator $\mathbf{C} : \mathcal{HL} \to \mathcal{HS}$ The natural curl operator $\mathbf{C} : \mathcal{HL} \to \mathcal{HS}$ follows from the coordinate-independent formulation of Stokes' circulation theorem:

$$(\mathbf{n}, \nabla \times \mathbf{W}) := \lim_{S\to 0} \frac{1}{S} \oint_l (\mathbf{W}, \mathbf{l})\, \mathrm{d}l, \tag{2.21}$$

where S is the surface enclosed by the closed curve l, $\mathbf{n}$ is the unit outward normal to S, $\mathbf{l}$ is the unit tangential vector to l and $\mathbf{W} : \mathbb{R}^n \to \mathbb{R}^n$ is a differentiable vector field. Let $\boldsymbol{\omega} = (\eta, \omega, \zeta)$ be a discrete vector in $\mathcal{HL}$ with its components located as shown in figure 2.1. Then for the radial component $(\mathbf{C}\boldsymbol{\omega})_r$, the curve l around the surface S is defined as the boundary of the gray plane in figure 2.2 for both types of cells.



(a) Radial component at the axis.

(b) Radial component away from the axis.

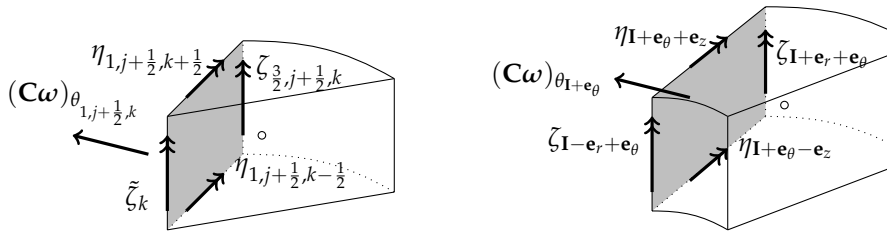
Figure 2.2: Surface for the determination of the radial component of the curl operator $\mathbf{C} : \mathcal{HL} \to \mathcal{HS}$.

The radial component of $\mathbf{C}\boldsymbol{\omega}$ is then approximated as:

$$\begin{aligned}(\mathbf{C}\boldsymbol{\omega})_{r_{\mathbf{I}+\mathbf{e}_r}} = \frac{1}{r_{i+\frac{1}{2}}\Delta\theta\Delta z_k}\Big[& r_{i+\frac{1}{2}}\Delta\theta(-\omega_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_z} + \omega_{\mathbf{I}+\mathbf{e}_r-\mathbf{e}_z}) \\ & + \Delta z_k(\zeta_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_\theta} - \zeta_{\mathbf{I}+\mathbf{e}_r-\mathbf{e}_\theta})\Big] \end{aligned} \tag{2.22}$$

$$= \frac{\zeta_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_\theta} - \zeta_{\mathbf{I}+\mathbf{e}_r-\mathbf{e}_\theta}}{r_{i+\frac{1}{2}}\Delta\theta} - \frac{\omega_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_z} - \omega_{\mathbf{I}+\mathbf{e}_r-\mathbf{e}_z}}{\Delta z_k}. \tag{2.23}$$

For the angular component $(\mathbf{C}\boldsymbol{\omega})_\theta$, the curve l around the surface S is defined as the boundary of the gray plane in figure 2.3.



(a) Angular component at the axis. (b) Angular component away from the axis.

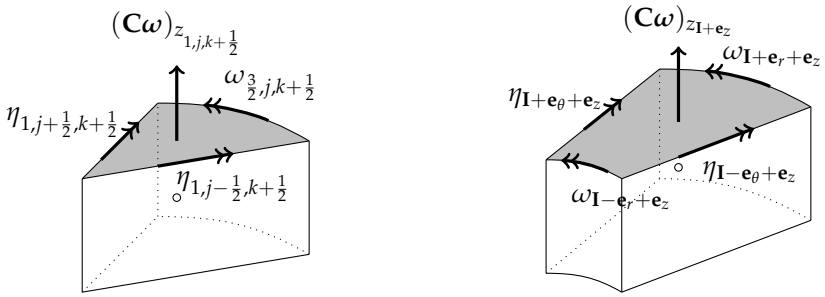
Figure 2.3: Surface for the determination of the angular component of the curl operator $\mathbf{C} : \mathcal{HL} \to \mathcal{HS}$.

The angular component is then approximated as:

$$(\mathbf{C}\boldsymbol{\omega})_{\theta_{\mathbf{I}+\mathbf{e}_\theta}} = \frac{1}{\Delta r_i \Delta z_k}\Big[\Delta r_i(\eta_{\mathbf{I}+\mathbf{e}_\theta+\mathbf{e}_z} - \eta_{\mathbf{I}+\mathbf{e}_\theta-\mathbf{e}_z}) + \Delta z_k(-\zeta_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_\theta} + \zeta_{\mathbf{I}-\mathbf{e}_r+\mathbf{e}_\theta})\Big] \tag{2.24}$$

$$= \frac{\eta_{\mathbf{I}+\mathbf{e}_\theta+\mathbf{e}_z} - \eta_{\mathbf{I}+\mathbf{e}_\theta-\mathbf{e}_z}}{\Delta z_k} - \frac{\zeta_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_\theta} - \zeta_{\mathbf{I}-\mathbf{e}_r+\mathbf{e}_\theta}}{\Delta r_i}. \tag{2.25}$$

Finally, for the axial component $(\mathbf{C}\boldsymbol{\omega})_z$, the curve l around the surface S is defined as the boundary of the gray plane shown in figure 2.4.



(a) Axial component at the axis. (b) Axial component away from the axis.

Figure 2.4: Surface for the determination of the axial component of the curl operator $\mathbf{C} : \mathcal{HL} \to \mathcal{HS}$.

The axial component is approximated as:

$$(\mathbf{C}\boldsymbol{\omega})_{z_{\mathbf{I}+\mathbf{e}_z}} = \frac{1}{r_i \Delta r_i \Delta\theta}\Big[\Delta r_i(-\eta_{\mathbf{I}+\mathbf{e}_\theta+\mathbf{e}_z} + \eta_{\mathbf{I}-\mathbf{e}_\theta+\mathbf{e}_z}) + \Delta\theta(r_{i+\frac{1}{2}}\omega_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_z} - r_{i-\frac{1}{2}}\omega_{\mathbf{I}-\mathbf{e}_r+\mathbf{e}_z})\Big] \tag{2.26}$$

$$= \frac{1}{r_i}\left[\frac{r_{i+\frac{1}{2}}\omega_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_z} - r_{i-\frac{1}{2}}\omega_{\mathbf{I}-\mathbf{e}_r+\mathbf{e}_z}}{\Delta r_i} - \frac{\eta_{\mathbf{I}+\mathbf{e}_\theta+\mathbf{e}_z} - \eta_{\mathbf{I}-\mathbf{e}_\theta+\mathbf{e}_z}}{\Delta\theta}\right]. \tag{2.27}$$

Notice that the non-existence of the angular vorticity element ω for axis cells where $i = 1$ is naturally resolved by the multiplication with $r_{\frac{1}{2}} = 0$. Hence, for axis cells, the expression reduces to:

$$(\mathbf{C}\boldsymbol{\omega})_{z_{\mathbf{I},i=1}} = \frac{2\omega_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_z}}{\Delta r_i} - \frac{2\left(\eta_{\mathbf{I}+\mathbf{e}_\theta+\mathbf{e}_z} - \eta_{\mathbf{I}-\mathbf{e}_\theta-\mathbf{e}_z}\right)}{\Delta r_i \Delta\theta}, \tag{2.28}$$

where on the right-hand side the i-component of the multi-index $\mathbf{I}$ is equal to 1.

2.2.3 Definition of the discrete inner products

To derive the adjoint operators, we need the notion of inner products on the discrete vector spaces. These are defined as approximations of the continuous L^2-inner product, using either the (second order accurate) Midpoint- or Trapezoidal integration rule.

An inner product on HC For the space HC, the inner product $(\cdot,\cdot)_{HC} : HC \times HC \to \mathbb{R}$ is defined as:

$$(u, v)_{HC} := \sum_{\mathbf{I}} u_{\mathbf{I}} v_{\mathbf{I}} r_i \Delta r_i \Delta\theta \Delta z_k, \tag{2.29}$$

where u and v are two discrete scalar functions defined in the cell centres.

An inner product on $\mathcal{HS}$ For the space $\mathcal{HS}$, the inner product $(\cdot,\cdot)_{\mathcal{HS}} : \mathcal{HS} \times \mathcal{HS} \to \mathbb{R}$ is defined as:

$$\begin{aligned}(\mathbf{u}, \mathbf{v})_{\mathcal{HS}} := \sum_{\mathbf{I}} \frac{\Delta r_i \Delta\theta \Delta z_k}{2} \Big[& r_{i-\frac{1}{2}} u_{\mathbf{I}-\mathbf{e}_r} v_{\mathbf{I}-\mathbf{e}_r} + r_{i+\frac{1}{2}} u_{\mathbf{I}+\mathbf{e}_r} v_{\mathbf{I}+\mathbf{e}_r} \\ & + r_i (u_{\mathbf{I}-\mathbf{e}_\theta} v_{\mathbf{I}-\mathbf{e}_\theta} + u_{\mathbf{I}+\mathbf{e}_\theta} v_{\mathbf{I}+\mathbf{e}_\theta}) \\ & + r_i (u_{\mathbf{I}-\mathbf{e}_z} v_{\mathbf{I}-\mathbf{e}_z} + u_{\mathbf{I}+\mathbf{e}_z} v_{\mathbf{I}+\mathbf{e}_z}) \Big].\end{aligned} \tag{2.30}$$

An inner product on $\mathcal{HL}$ The inner product $(\cdot,\cdot)_{\mathcal{HL}} : \mathcal{HL} \times \mathcal{HL} \to \mathbb{R}$ of two vectors $\boldsymbol{\omega} = (\eta, \omega, \zeta)$ and $\bar{\boldsymbol{\omega}} = (\bar{\eta}, \bar{\omega}, \bar{\zeta})$ in $\mathcal{HL}$ is given by:

$$\begin{aligned}(\boldsymbol{\omega}, \bar{\boldsymbol{\omega}})_{\mathcal{HL}} := \sum_{\mathbf{I}, i\neq 1} \frac{\Delta r_i \Delta\theta \Delta z_k}{2} \cdot \Bigg[& r_i \frac{\eta_{\mathbf{I}-\mathbf{e}_\theta-\mathbf{e}_z} \bar{\eta}_{\mathbf{I}-\mathbf{e}_\theta-\mathbf{e}_z} + \eta_{\mathbf{I}+\mathbf{e}_\theta-\mathbf{e}_z} \bar{\eta}_{\mathbf{I}+\mathbf{e}_\theta-\mathbf{e}_z}}{2} \\ & + r_i \frac{\eta_{\mathbf{I}-\mathbf{e}_\theta+\mathbf{e}_z} \bar{\eta}_{\mathbf{I}-\mathbf{e}_\theta+\mathbf{e}_z} + \eta_{\mathbf{I}+\mathbf{e}_\theta+\mathbf{e}_z} \bar{\eta}_{\mathbf{I}+\mathbf{e}_\theta+\mathbf{e}_z}}{2}\end{aligned}$$

$$
\begin{aligned}
&+ r_{i-\frac{1}{2}} \frac{\omega_{\mathbf{I}-\mathbf{e}_r-\mathbf{e}_z}\bar{\omega}_{\mathbf{I}-\mathbf{e}_r-\mathbf{e}_z} + \omega_{\mathbf{I}-\mathbf{e}_r+\mathbf{e}_z}\bar{\omega}_{\mathbf{I}-\mathbf{e}_r+\mathbf{e}_z}}{2} \\
&+ r_{i+\frac{1}{2}} \frac{\omega_{\mathbf{I}+\mathbf{e}_r-\mathbf{e}_z}\bar{\omega}_{\mathbf{I}+\mathbf{e}_r-\mathbf{e}_z} + \omega_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_z}\bar{\omega}_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_z}}{2} \\
&+ \left(\frac{r_{i-1}+r_i}{2}\right)\left(\frac{\zeta_{\mathbf{I}-\mathbf{e}_r-\mathbf{e}_\theta}\bar{\zeta}_{\mathbf{I}-\mathbf{e}_r-\mathbf{e}_\theta} + \zeta_{\mathbf{I}-\mathbf{e}_r+\mathbf{e}_\theta}\bar{\zeta}_{\mathbf{I}-\mathbf{e}_r+\mathbf{e}_\theta}}{2}\right) \\
&+ \left(\frac{r_i+r_{i+1}}{2}\right)\left(\frac{\zeta_{\mathbf{I}+\mathbf{e}_r-\mathbf{e}_\theta}\bar{\zeta}_{\mathbf{I}+\mathbf{e}_r-\mathbf{e}_\theta} + \zeta_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_\theta}\bar{\zeta}_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_\theta}}{2}\right)\Bigg] \\
&+ \sum_{\mathbf{I},i=1} \frac{\Delta r_i \Delta\theta \Delta z_k}{2} \cdot \Bigg[r_i \frac{\eta_{\mathbf{I}-\mathbf{e}_\theta-\mathbf{e}_z}\bar{\eta}_{\mathbf{I}-\mathbf{e}_\theta-\mathbf{e}_z} + \eta_{\mathbf{I}+\mathbf{e}_\theta-\mathbf{e}_z}\bar{\eta}_{\mathbf{I}+\mathbf{e}_\theta-\mathbf{e}_z}}{2} \\
&+ r_i \frac{\eta_{\mathbf{I}-\mathbf{e}_\theta+\mathbf{e}_z}\bar{\eta}_{\mathbf{I}-\mathbf{e}_\theta+\mathbf{e}_z} + \eta_{\mathbf{I}+\mathbf{e}_\theta+\mathbf{e}_z}\bar{\eta}_{\mathbf{I}+\mathbf{e}_\theta+\mathbf{e}_z}}{2} \\
&+ r_{i+\frac{1}{2}} \frac{\omega_{\mathbf{I}+\mathbf{e}_r-\mathbf{e}_z}\bar{\omega}_{\mathbf{I}+\mathbf{e}_r-\mathbf{e}_z} + \omega_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_z}\bar{\omega}_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_z}}{2} \\
&+ \frac{r_i}{2}\widetilde{\zeta}_k\widetilde{\bar{\zeta}}_k + \left(\frac{r_i+r_{i+1}}{2}\right)\left(\frac{\zeta_{\mathbf{I}+\mathbf{e}_r-\mathbf{e}_\theta}\bar{\zeta}_{\mathbf{I}+\mathbf{e}_r-\mathbf{e}_\theta} + \zeta_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_\theta}\bar{\zeta}_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_\theta}}{2}\right)\Bigg].
\end{aligned}
\tag{2.31}
$$

Notice that for $i = 1$, the degeneracy of the cell has been taken into account: for the angular components ω and $\bar{\omega}$, this follows naturally by multiplication with $r_{\frac{1}{2}} = 0$. For the axial components ζ and $\bar{\zeta}$, the expression is altered to accommodate the collapse of the cell face at $r = 0$. It can be verified that all three discrete inner products satisfy the required symmetry, linearity and positive-definiteness properties.

2.2.4 Definition of the adjoint vector operations

The discrete operators derived in section 2.2.2 only allow the trivial successive applications **CG** and **DC**, which are identically zero. Second order operators like **DG** of a scalar and **CC** of a vector are not possible because the range does not equal the domain of the consecutive first order operators. To overcome this, the adjoint operators $\overline{\mathbf{D}}$, $\overline{\mathbf{G}}$ and $\overline{\mathbf{C}}$ are derived using the Support Operator Method [78]. By choosing a *prime* (natural) operator, the associated *derived* (adjoint) operator follows from the discrete inner product in combination with the identities:

$$
\mathbf{D} = -\mathbf{G}^*, \qquad \mathbf{C} = \mathbf{C}^*, \tag{2.32}
$$

where the $*$ denotes the adjoint with respect to the associated inner product. More specifically, starting with the natural operator $\mathbf{D} : \mathcal{HS} \rightarrow HC$, the operator $\overline{\mathbf{G}}$:

$HC \rightarrow \mathcal{HS}$ is defined through:

$$(\mathbf{Du}, v)_{HC} = -\left(\mathbf{u}, \overline{\mathbf{G}} v\right)_{\mathcal{HS}}, \tag{2.33}$$

for any $\mathbf{u} \in \mathcal{HS}$ and $v \in HC$. In a similar way, expressions can be derived for the derived divergence operator $\overline{\mathbf{D}} : \mathcal{HL} \rightarrow HN$ and a derived curl operator $\overline{\mathbf{C}} : \mathcal{HS} \rightarrow \mathcal{HL}$ through:

$$(\mathbf{G}u, \mathbf{v})_{\mathcal{HL}} = -\left(u, \overline{\mathbf{D}}\mathbf{v}\right)_{HN}, \qquad u \in HN, \mathbf{v} \in \mathcal{HL}, \tag{2.34}$$

$$(\mathbf{Cu}, \mathbf{v})_{\mathcal{HS}} = \left(\mathbf{u}, \overline{\mathbf{C}}\mathbf{v}\right)_{\mathcal{HL}}, \qquad \mathbf{u} \in \mathcal{HL}, \mathbf{v} \in \mathcal{HS}. \tag{2.35}$$

With both natural and adjoint discrete operators, it is now possible to discretize combinations like:

$$\mathbf{D}\overline{\mathbf{G}} : HC \rightarrow HC, \qquad \overline{\mathbf{D}}\mathbf{G} : HN \rightarrow HN, \tag{2.36}$$

$$\mathbf{C}\overline{\mathbf{C}} : \mathcal{HS} \rightarrow \mathcal{HS}, \qquad \overline{\mathbf{C}}\mathbf{C} : \mathcal{HL} \rightarrow \mathcal{HL}, \tag{2.37}$$

$$\mathbf{G}\overline{\mathbf{D}} : \mathcal{HL} \rightarrow \mathcal{HL}, \qquad \overline{\mathbf{G}}\mathbf{D} : \mathcal{HS} \rightarrow \mathcal{HS}, \tag{2.38}$$

and even the vector Laplacian. It is shown in Hyman and Shashkov [43] that the discrete operators satisfy several additional important theorems from vector calculus. Just like the natural operators, the adjoint operators $\overline{\mathbf{G}}$, $\overline{\mathbf{C}}$ and $\overline{\mathbf{D}}$ form a sequence (but in reversed direction):

$$HC \xrightarrow{\overline{\mathbf{G}}} \mathcal{HS} \xrightarrow{\overline{\mathbf{C}}} \mathcal{HL} \xrightarrow{\overline{\mathbf{D}}} HN. \tag{2.39}$$

We will now construct the adjoint operators $\overline{\mathbf{G}}$ and $\overline{\mathbf{C}}$ for cylindrical grids with a non-uniform radial and axial distribution (we do not require $\overline{\mathbf{D}}$). In addition, we will prove that the constructed adjoint operators are indeed the formal adjoints of the associated natural operators with respect to their inner products.

The gradient operator $\overline{\mathbf{G}} : HC \rightarrow \mathcal{HS}$ The components of the gradient operator $\overline{\mathbf{G}} : HC \rightarrow \mathcal{HS}$ are defined as:

$$\left(\overline{\mathbf{G}}p\right)_{r_{\mathbf{I}+\mathbf{e}_r}} = \frac{2(p_{\mathbf{I}+2\,\mathbf{e}_r} - p_{\mathbf{I}})}{\Delta r_i + \Delta r_{i+1}}, \tag{2.40}$$

$$\left(\overline{\mathbf{G}}p\right)_{\theta_{\mathbf{I}+\mathbf{e}_\theta}} = \frac{p_{\mathbf{I}+2\,\mathbf{e}_\theta} - p_{\mathbf{I}}}{r_i \Delta\theta}, \tag{2.41}$$

$$\left(\overline{\mathbf{G}}p\right)_{z_{\mathbf{I}+\mathbf{e}_z}} = \frac{2(p_{\mathbf{I}+2\,\mathbf{e}_z} - p_{\mathbf{I}})}{\Delta z_k + \Delta z_{k+1}}, \tag{2.42}$$

for some scalar $p \in HC$.

Proposition 2.2.1. *For any $p \in HC$ and $\mathbf{u} \in \mathcal{HS}$ on an infinite domain, it holds that $(\mathbf{Du}, p)_{HC} = -(\mathbf{u}, \overline{\mathbf{G}}p)_{\mathcal{HS}}$.*

Proof. See appendix A. □

On finite domains, the boundary conditions on HC and $\mathcal{HS}$ need to be chosen consistently in order for expression 2.33 to hold exactly. We assume a finite cylindrical domain of radius R and length L with an associated grid that includes a single layer of ghost cells at both the radial and axial walls. The inner product $(\mathbf{Du}, p)_{HC}$ contains only internal values of p, while the discrete gradient in $(\mathbf{u}, \overline{\mathbf{G}}p)_{\mathcal{HS}}$ also includes values of p in the layer of ghost cells. Consider now a computational cell that shares a face with the cylinder boundary. The contribution of the adjacent ghost cell to $(\mathbf{u}, \overline{\mathbf{G}}p)_{\mathcal{HS}}$ can be removed by actively setting the discrete gradient of p at the boundary to zero. But for consistency in the product $(\mathbf{Du}, p)_{HC}$, the value of the component of $\mathbf{u}$ normal to the boundary then has to be set to zero (compare the contributions in equations A.2 and A.3). This combination of Neumann and Dirichlet boundary conditions for the pressure p and the flow velocity $\mathbf{u}$, respectively, are the well known expressions for the simulation of a solid wall without penetration.

A second case involves periodic wall conditions. For simplicity, we assume that the axial boundaries at $z = \pm L/2$ are periodic, so that the layers of ghost cells at these walls coincide with the first and last slices of the internal cells of the grid. With minor adjustments, the same strategy as the proof for an infinite domain can be used to demonstrate that in this case the inner products are also equal. Hence, on finite domains, proposition 2.2.1 is in particular valid for the common situation of cylindrical pipe flows with solid or periodic walls.

The curl operator $\overline{\mathbf{C}} : \mathcal{HS} \to \mathcal{HL}$ The adjoint curl operator $\overline{\mathbf{C}}$ is derived according to equation 2.21 in a similar way as the natural curl operator $\mathbf{C}$. It will be shown that the resulting expressions for the components of $\overline{\mathbf{C}}$ are indeed the adjoints of the associated components of $\mathbf{C}$ with respect to the discrete inner product $(\cdot, \cdot)_{\mathcal{HL}}$. Throughout the derivation, let $\mathbf{u} = (u_r, u_\theta, u_z)$ be a discrete vector in $\mathcal{HS}$. Then for the radial component of the adjoint curl operator $(\overline{\mathbf{C}}\mathbf{u})_r$, the surface S is defined as the gray plane in figure 2.5.

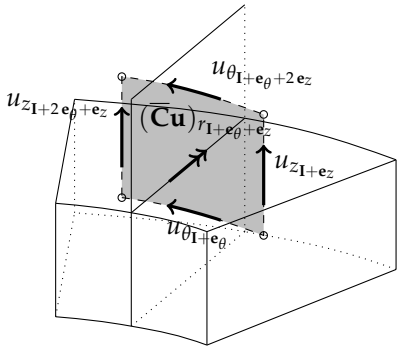


Figure 2.5: Surface for the determination of the radial component of the curl operator $\overline{\mathbf{C}} : \mathcal{HS} \to \mathcal{HL}$.

The component of the discrete curl operator $\overline{\mathbf{C}}$ in the radial direction then follows as:

$$(\overline{\mathbf{C}}\mathbf{u})_{r_{\mathbf{I}+\mathbf{e}_\theta+\mathbf{e}_z}} = \frac{2}{r_i\Delta\theta(\Delta z_k+\Delta z_{k+1})}\left[\frac{\Delta z_k+\Delta z_{k+1}}{2}(u_{z_{\mathbf{I}+2\mathbf{e}_\theta+\mathbf{e}_z}}-u_{z_{\mathbf{I}+\mathbf{e}_z}})\right.$$
$$\left.+\, r_i\Delta\theta(-u_{\theta_{\mathbf{I}+\mathbf{e}_\theta+2\mathbf{e}_z}}+u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}})\right] \tag{2.43}$$
$$= \frac{u_{z_{\mathbf{I}+2\mathbf{e}_\theta+\mathbf{e}_z}}-u_{z_{\mathbf{I}+\mathbf{e}_z}}}{r_i\Delta\theta} - \frac{2(u_{\theta_{\mathbf{I}+\mathbf{e}_\theta+2\mathbf{e}_z}}-u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}})}{\Delta z_k+\Delta z_{k+1}}. \tag{2.44}$$

For the angular component, the surface is defined as in figure 2.6.

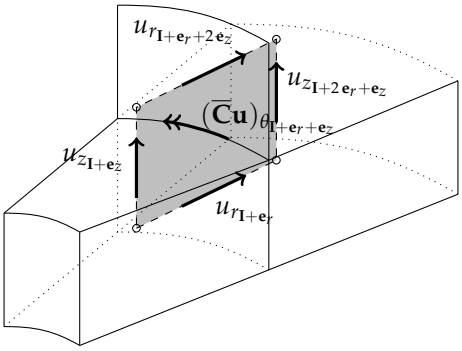


Figure 2.6: Surface for the determination of the angular component of the curl operator $\overline{\mathbf{C}} : \mathcal{HS} \to \mathcal{HL}$.

The component of the discrete curl operator $(\overline{\mathbf{C}}\mathbf{u})_\theta$ in the angular direction is:

$$(\overline{\mathbf{C}}\mathbf{u})_{\theta_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_z}} = \frac{4}{(\Delta r_i+\Delta r_{i+1})(\Delta z_k+\Delta z_{k+1})}\left[\frac{\Delta z_k+\Delta z_{k+1}}{2}\cdot\right. \tag{2.45}$$

$$\left(u_{z_{\mathbf{I}+\mathbf{e}_z}} - u_{z_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_z}}\right) + \frac{\Delta r_i + \Delta r_{i+1}}{2}\left(u_{r_{\mathbf{I}+\mathbf{e}_r+2\,\mathbf{e}_z}} - u_{r_{\mathbf{I}+\mathbf{e}_r}}\right)\Bigg]$$
$$= \frac{2(u_{r_{\mathbf{I}+\mathbf{e}_r+2\,\mathbf{e}_z}} - u_{r_{\mathbf{I}+\mathbf{e}_r}})}{\Delta z_k + \Delta z_{k+1}} - \frac{2(u_{z_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_z}} - u_{z_{\mathbf{I}+\mathbf{e}_z}})}{\Delta r_i + \Delta r_{i+1}}. \tag{2.46}$$

For the axial component for cells away from the axis, the surface is defined as in figure 2.7.

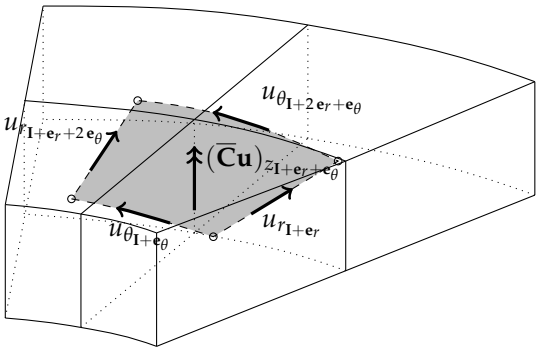


Figure 2.7: Surface for the determination of the axial component of the curl operator $\overline{\mathbf{C}} : \mathcal{HS} \to \mathcal{HL}$.

In this case, the component of the discrete curl operator $(\overline{\mathbf{C}}\mathbf{u})_z$ in the axial direction is:

$$(\overline{\mathbf{C}}\mathbf{u})_{z_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_\theta}} = \frac{4}{(r_i + r_{i+1})(\Delta r_i + \Delta r_{i+1})\Delta\theta}\left[\frac{\Delta r_i + \Delta r_{i+1}}{2}\cdot\right. \tag{2.47}$$
$$\left.\left(u_{r_{\mathbf{I}+\mathbf{e}_r}} - u_{r_{\mathbf{I}+\mathbf{e}_r+2\,\mathbf{e}_\theta}}\right) + \Delta\theta\left(r_{i+1}u_{\theta_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_\theta}} - r_i u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}\right)\right]$$
$$= \frac{2}{r_i + r_{i+1}}\left[\frac{2\left(r_{i+1}u_{\theta_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_\theta}} - r_i u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}\right)}{\Delta r_i + \Delta r_{i+1}} - \frac{u_{r_{\mathbf{I}+\mathbf{e}_r+2\,\mathbf{e}_\theta}} - u_{r_{\mathbf{I}+\mathbf{e}_r}}}{\Delta\theta}\right]. \tag{2.48}$$

For the cells near the axis, the surface S is defined as in figure 2.8.
For these cells, the component of the discrete curl operator $\overline{\mathbf{C}}$ in the axial direction is:

$$(\overline{\mathbf{C}}\mathbf{u})_{z_k} = \frac{8}{N_\theta \Delta r_1^2 \Delta\theta}\sum_{j=1}^{N_\theta}\frac{\Delta r_1}{2}\Delta\theta u_{\theta_{1,j+\frac{1}{2},k}} = \frac{4}{\Delta r_1}\frac{1}{N_\theta}\sum_{j=1}^{N_\theta} u_{\theta_{1,j+\frac{1}{2},k}}. \tag{2.49}$$

Proposition 2.2.2. *For any $\boldsymbol{\omega} \in \mathcal{HL}$ and $\mathbf{u} \in \mathcal{HS}$ on an infinite domain, it holds that $(\mathbf{C}\boldsymbol{\omega}, \mathbf{u})_{\mathcal{HS}} = (\boldsymbol{\omega}, \overline{\mathbf{C}}\mathbf{u})_{\mathcal{HL}}$.*

Proof. See appendix A. □

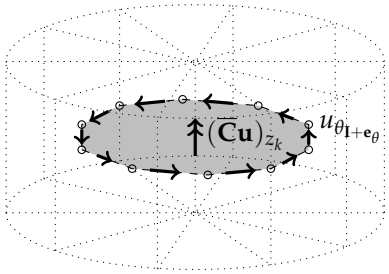
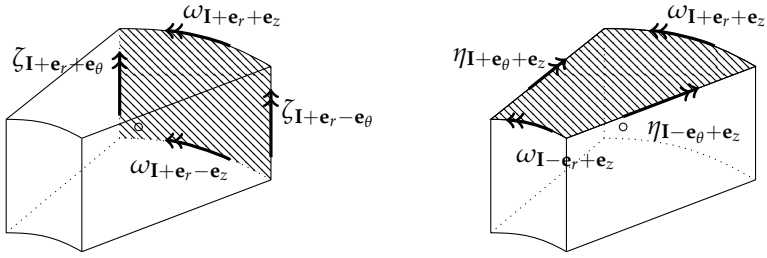


Figure 2.8: Surface for the determination of the axial component of the curl operator $\overline{\mathbf{C}} : \mathcal{HS} \to \mathcal{HL}$.

For finite domains, some care has to be taken for the two inner products to be equal. Reviewing the proof in appendix A shows that we only have to consider the cells that share a face with the boundary. We assume a finite discretized cylindrical domain of radius R and length L. Arguably the simplest case is when the discrete components of $\boldsymbol{\omega}$ are zero at the boundaries. After inspection of the expressions in appendix A, this renders the contribution from the cell face at the boundary to both products $(\boldsymbol{\omega}, \overline{\mathbf{C}}\mathbf{u})_{\mathcal{HL}}$ and $(\mathbf{C}\mathbf{u}, \boldsymbol{\omega})_{\mathcal{HS}}$ equal to zero. Subsequently, both inner products only contain the summation of internal values and they are exactly equal.

If the components of $\boldsymbol{\omega}$ are not zero at the wall, then the proof can be used to derive the values for the components of $\mathbf{u}$ in the surrounding layer of ghost cells. Consider a boundary cell with its positive face at $r = R$ which contains two components of $\boldsymbol{\omega}$, namely ω and ζ, and one component of $\mathbf{u}$, namely u_r, as in figure 2.9a.



(a) Cell near $r = R$. (b) Cell near $z = L/2$.

Figure 2.9: Cells near $r = R$ and $z = L/2$ showing the vorticity components that lie on the (hatched) domain boundaries.

From equations A.10 and A.12, the contributions to the angular component $\omega_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_z}$ come from cells $\mathbf{I}$ and $\mathbf{I} + 2\,\mathbf{e}_z$ alone (since we only sum over internal cells), and together they provide a condition for the value of $u_{z_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_z}}$ in the ghost cell layer

by solving:

$$\frac{r_{i+\frac{1}{2}}\Delta\theta}{2}\left(\Delta z_k + \Delta z_{k+1}\right) u_{z_{\mathbf{I}+\mathbf{e}_z}} = -\frac{r_{i+\frac{1}{2}}\Delta r_i \Delta\theta}{4}\left(\Delta z_k + \Delta z_{k+1}\right) \frac{2(u_{z_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_z}} - u_{z_{\mathbf{I}+\mathbf{e}_z}})}{\Delta r_i + \Delta r_{i+1}}, \tag{2.50}$$

which yields:

$$u_{z_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_z}} = -\frac{\Delta r_{i+1}}{\Delta r_i} u_{z_{\mathbf{I}+\mathbf{e}_z}}, \quad i = N_r. \tag{2.51}$$

Similarly, from equations A.14 and A.16, we obtain that the only contributions to the axial component $\zeta_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_\theta}$ come from cells $\mathbf{I}$ and $\mathbf{I}+2\,\mathbf{e}_\theta$. Equating yields for the angular velocity $u_{\theta_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_\theta}}$ in the ghost cell layer:

$$u_{\theta_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_\theta}} = -\frac{r_i}{r_{i+1}}\frac{\Delta r_{i+1}}{\Delta r_i} u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}, \quad i = N_r. \tag{2.52}$$

The same can be done for the boundaries at $z = \pm L/2$ of the cylindrical grid. There we have the components η and ω of $\boldsymbol{\omega}$ (see figure 2.9b). Collecting terms yields for the velocities in the lower ghost value layer:

$$u_{r_{\mathbf{I}+\mathbf{e}_r-2\,\mathbf{e}_z}} = -\frac{\Delta z_{k-1}}{\Delta z_k} u_{r_{\mathbf{I}+\mathbf{e}_r}}, \quad u_{\theta_{\mathbf{I}+\mathbf{e}_\theta-2\,\mathbf{e}_z}} = -\frac{\Delta z_{k-1}}{\Delta z_k} u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}, \quad k = 1, \tag{2.53}$$

and for the velocities in the upper ghost value layer:

$$u_{r_{\mathbf{I}+\mathbf{e}_r+2\,\mathbf{e}_z}} = -\frac{\Delta z_{k+1}}{\Delta z_k} u_{r_{\mathbf{I}+\mathbf{e}_r}}, \quad u_{\theta_{\mathbf{I}+\mathbf{e}_\theta+2\,\mathbf{e}_z}} = -\frac{\Delta z_{k+1}}{\Delta z_k} u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}, \quad k = N_z. \tag{2.54}$$

Hence, equality of the inner products can be attained by either choosing the components of the vector $\boldsymbol{\omega}$ in $\mathcal{HL}$ to be zero, or, for non-zero components of $\boldsymbol{\omega}$, choosing the boundary values of the vector $\mathbf{u}$ according to expressions 2.51 - 2.54. In section 2.4.2, we will show that when $\boldsymbol{\omega} = \boldsymbol{\omega}(\mathbf{u})$, these conditions have an actual physical meaning: they represent the free-slip (or stress-free) and no-slip boundary conditions for the flow velocity, respectively.

2.3 Mimetic discretization of the flow equations

In this section, we will apply the mimetic discretization techniques from section 2.2 to generate a finite difference discretization of the incompressible Navier- Stokes equations. We will propose spatial discretizations for the convective part and the

viscous part in the next sections. The discretization of the pressure gradient ∇p follows rather trivially: because the pressure is an element of the space HC, the approximation of its gradient is $\overline{\mathbf{G}}p$, with $\overline{\mathbf{G}} : HC \rightarrow \mathcal{HS}$ the adjoint gradient operator as in equations 2.40-2.42.

2.3.1 Discretization of the convective term

The convective term of the vector momentum equations $(\mathbf{u} \cdot \nabla)\mathbf{u}$ is a second order tensor, and it does not allow direct application of the mimetic operators derived in section 2.2. Instead, it can be rewritten as:

$$(\mathbf{u} \cdot \nabla)\mathbf{u} = (\nabla \times \mathbf{u}) \times \mathbf{u} + \frac{1}{2}\nabla(\mathbf{u} \cdot \mathbf{u}), \tag{2.55}$$

occasionally referred to as the rotational formulation, where the right-hand side only consists of first order operators. The term $\frac{1}{2}\nabla(\mathbf{u} \cdot \mathbf{u})$ is added to the pressure gradient, which leaves the term $\mathbf{N}(\mathbf{u}) := (\nabla \times \mathbf{u}) \times \mathbf{u} = \boldsymbol{\omega} \times \mathbf{u}$, where $\boldsymbol{\omega}$ is the flow vorticity, as the remaining convection part. With the vorticity in cylindrical coordinates given by:

$$\nabla \times \mathbf{u} = \begin{pmatrix} \eta \\ \omega \\ \zeta \end{pmatrix} = \begin{pmatrix} \frac{1}{r}\frac{\partial u_z}{\partial \theta} - \frac{\partial u_\theta}{\partial z} \\ \frac{\partial u_r}{\partial z} - \frac{\partial u_z}{\partial r} \\ \frac{1}{r}\frac{\partial (r u_\theta)}{\partial r} - \frac{1}{r}\frac{\partial u_r}{\partial \theta} \end{pmatrix}, \tag{2.56}$$

the convective part $\mathbf{N}$ becomes:

$$\mathbf{N}(\mathbf{u}) = \begin{pmatrix} \omega u_z - \zeta u_\theta \\ \zeta u_r - \eta u_z \\ \eta u_\theta - \omega u_r \end{pmatrix}. \tag{2.57}$$

The discretization of $\mathbf{N}$ requires the approximation of the vorticity components and subsequently an averaging procedure of the velocity components, as velocity and vorticity are members of different discrete spaces ($\mathcal{HS}$ and $\mathcal{HL}$, respectively). As the velocity vector in the staggered grid is an element of $\mathcal{HS}$, we notice that $\boldsymbol{\omega} = \nabla \times \mathbf{u}$ is discretely approximated by $\overline{\mathbf{C}}\mathbf{u}$, where $\overline{\mathbf{C}}$ is the adjoint curl operator $\overline{\mathbf{C}} : \mathcal{HS} \rightarrow \mathcal{HL}$. Then, with the vorticity components known, the discrete approximation of equation 2.57 requires spatial averaging to obtain an estimate at the location of the velocity components. The choice of averaging is restricted by the following considerations:

- The averaging should be sufficiently accurate.
- The averaging should be consistent for the cells near the polar axis.
- The averaging should allow conservation of momentum when subjected to discrete integration.
- The averaging should allow conservation of energy in combination with a discrete inner product.

With this in mind, we propose the following discretization for the radial convective part:

$$\begin{aligned}\mathbf{N}_{r_{\mathbf{I}+\mathbf{e}_r}} = \frac{1}{2(\Delta r_i + \Delta r_{i+1})} \Bigg[& \omega_{\mathbf{I}+\mathbf{e}_r-\mathbf{e}_z} \left(\Delta r_i u_{z_{\mathbf{I}-\mathbf{e}_z}} + \Delta r_{i+1} u_{z_{\mathbf{I}+2\,\mathbf{e}_r-\mathbf{e}_z}} \right) \\ & + \omega_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_z} \left(\Delta r_i u_{z_{\mathbf{I}+\mathbf{e}_z}} + \Delta r_{i+1} u_{z_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_z}} \right) \Bigg] \\ - \frac{1}{4} \frac{r_i + r_{i+1}}{2} \Bigg[& \zeta_{\mathbf{I}+\mathbf{e}_r-\mathbf{e}_\theta} \left(\frac{u_{\theta_{\mathbf{I}-\mathbf{e}_\theta}}}{r_i} + \frac{u_{\theta_{\mathbf{I}+2\,\mathbf{e}_r-\mathbf{e}_\theta}}}{r_{i+1}} \right) \\ & + \zeta_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_\theta} \left(\frac{u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}}{r_i} + \frac{u_{\theta_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_\theta}}}{r_{i+1}} \right) \Bigg]. \end{aligned} \tag{2.58}$$

For the angular convective part:

$$\begin{aligned}\mathbf{N}_{\theta_{\mathbf{I}+\mathbf{e}_\theta}} = \frac{1}{2 r_i^2 \Delta r_i} \Bigg(& r_{i-\frac{1}{2}} \frac{r_{i-1} + r_i}{2} \frac{\Delta r_{i-1} + \Delta r_i}{2} \zeta_{\mathbf{I}-\mathbf{e}_r+\mathbf{e}_\theta} \frac{u_{r_{\mathbf{I}-\mathbf{e}_r}} + u_{r_{\mathbf{I}-\mathbf{e}_r+2\,\mathbf{e}_\theta}}}{2} \\ & + r_{i+\frac{1}{2}} \frac{r_i + r_{i+1}}{2} \frac{\Delta r_i + \Delta r_{i+1}}{2} \zeta_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_\theta} \frac{u_{r_{\mathbf{I}+\mathbf{e}_r}} + u_{r_{\mathbf{I}+\mathbf{e}_r+2\,\mathbf{e}_\theta}}}{2} \Bigg) \\ - \frac{1}{2} \Bigg(& \eta_{\mathbf{I}+\mathbf{e}_\theta-\mathbf{e}_z} \frac{u_{z_{\mathbf{I}-\mathbf{e}_z}} + u_{z_{\mathbf{I}+2\,\mathbf{e}_\theta-\mathbf{e}_z}}}{2} \\ & + \eta_{\mathbf{I}+\mathbf{e}_\theta+\mathbf{e}_z} \frac{u_{z_{\mathbf{I}+\mathbf{e}_z}} + u_{z_{\mathbf{I}+2\,\mathbf{e}_\theta+\mathbf{e}_z}}}{2} \Bigg). \end{aligned} \tag{2.59}$$

Finally, for the axial convective part:

$$\begin{aligned}\mathbf{N}_{z_{\mathbf{I}+\mathbf{e}_z}} = \frac{1}{2} \Bigg(& \eta_{\mathbf{I}-\mathbf{e}_\theta+\mathbf{e}_z} \frac{\Delta z_k u_{\theta_{\mathbf{I}-\mathbf{e}_\theta}} + \Delta z_{k+1} u_{\theta_{\mathbf{I}-\mathbf{e}_\theta+2\,\mathbf{e}_z}}}{\Delta z_k + \Delta z_{k+1}} \\ & + \eta_{\mathbf{I}+\mathbf{e}_\theta+\mathbf{e}_z} \frac{\Delta z_k u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}} + \Delta z_{k+1} u_{\theta_{\mathbf{I}+\mathbf{e}_\theta+2\,\mathbf{e}_z}}}{\Delta z_k + \Delta z_{k+1}} \Bigg) \end{aligned}$$

$$- \frac{1}{2r_i} \left(r_{i-\frac{1}{2}} \omega_{\mathbf{I}-\mathbf{e}_r+\mathbf{e}_z} \frac{\Delta z_k u_{r_{\mathbf{I}-\mathbf{e}_r}} + \Delta z_{k+1} u_{r_{\mathbf{I}-\mathbf{e}_r+2\mathbf{e}_z}}}{\Delta z_k + \Delta z_{k+1}} \right.$$
$$\left. + r_{i+\frac{1}{2}} \omega_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_z} \frac{\Delta z_k u_{r_{\mathbf{I}+\mathbf{e}_r}} + \Delta z_{k+1} u_{r_{\mathbf{I}+\mathbf{e}_r+2\mathbf{e}_z}}}{\Delta z_k + \Delta z_{k+1}} \right). \tag{2.60}$$

In section 2.4 we will show that this choice of discretization leads to conservation of momentum as well as to conservation of kinetic energy in the absence of viscosity. In section 2.5 we will investigate its accuracy.

For comparison with the discretization suggested by Barbosa and Daube [6], the expressions 2.58 - 2.60 for uniform grids reduce to:

$$\mathbf{N}_{r_{\mathbf{I}+\mathbf{e}_r}} = \frac{1}{2} \left(\omega_{\mathbf{I}+\mathbf{e}_r-\mathbf{e}_z} \frac{u_{z_{\mathbf{I}-\mathbf{e}_z}} + u_{z_{\mathbf{I}+2\mathbf{e}_r-\mathbf{e}_z}}}{2} + \omega_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_z} \frac{u_{z_{\mathbf{I}+\mathbf{e}_z}} + u_{z_{\mathbf{I}+2\mathbf{e}_r+\mathbf{e}_z}}}{2} \right)$$
$$- \frac{r_{i+\frac{1}{2}}}{4} \left[\zeta_{\mathbf{I}+\mathbf{e}_r-\mathbf{e}_\theta} \left(\frac{u_{\theta_{\mathbf{I}-\mathbf{e}_\theta}}}{r_i} + \frac{u_{\theta_{\mathbf{I}+2\mathbf{e}_r-\mathbf{e}_\theta}}}{r_{i+1}} \right) \right.$$
$$\left. + \zeta_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_\theta} \left(\frac{u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}}{r_i} + \frac{u_{\theta_{\mathbf{I}+2\mathbf{e}_r+\mathbf{e}_\theta}}}{r_{i+1}} \right) \right], \tag{2.61}$$

$$\mathbf{N}_{\theta_{\mathbf{I}+\mathbf{e}_\theta}} = \frac{1}{2r_i^2} \left(r_{i-\frac{1}{2}}^2 \zeta_{\mathbf{I}-\mathbf{e}_r+\mathbf{e}_r} \frac{u_{r_{\mathbf{I}-\mathbf{e}_r}} + u_{r_{\mathbf{I}-\mathbf{e}_r+2\mathbf{e}_\theta}}}{2} \right.$$
$$\left. + r_{i+\frac{1}{2}}^2 \zeta_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_\theta} \frac{u_{r_{\mathbf{I}+\mathbf{e}_r}} + u_{r_{\mathbf{I}+\mathbf{e}_r+2\mathbf{e}_\theta}}}{2} \right)$$
$$- \frac{1}{2} \left(\eta_{\mathbf{I}+\mathbf{e}_\theta-\mathbf{e}_z} \frac{u_{z_{\mathbf{I}-\mathbf{e}_z}} + u_{z_{\mathbf{I}+2\mathbf{e}_\theta-\mathbf{e}_z}}}{2} \right.$$
$$\left. + \eta_{\mathbf{I}+\mathbf{e}_\theta+\mathbf{e}_z} \frac{u_{z_{\mathbf{I}+\mathbf{e}_z}} + u_{z_{\mathbf{I}+2\mathbf{e}_\theta+\mathbf{e}_z}}}{2} \right), \tag{2.62}$$

$$\mathbf{N}_{z_{\mathbf{I}+\mathbf{e}_z}} = \frac{1}{2} \left(\eta_{\mathbf{I}-\mathbf{e}_\theta+\mathbf{e}_z} \frac{u_{\theta_{\mathbf{I}-\mathbf{e}_\theta}} + u_{\theta_{\mathbf{I}-\mathbf{e}_\theta+2\mathbf{e}_z}}}{2} + \eta_{\mathbf{I}+\mathbf{e}_\theta+\mathbf{e}_z} \frac{u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}} + u_{\theta_{\mathbf{I}+\mathbf{e}_\theta+2\mathbf{e}_z}}}{2} \right)$$
$$- \frac{1}{2r_i} \left(r_{i-\frac{1}{2}} \omega_{\mathbf{I}-\mathbf{e}_r+\mathbf{e}_z} \frac{u_{r_{\mathbf{I}-\mathbf{e}_r}} + u_{r_{\mathbf{I}-\mathbf{e}_r+2\mathbf{e}_z}}}{2} \right.$$
$$\left. + r_{i+\frac{1}{2}} \omega_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_z} \frac{u_{r_{\mathbf{I}+\mathbf{e}_r}} + u_{r_{\mathbf{I}+\mathbf{e}_r+2\mathbf{e}_z}}}{2} \right). \tag{2.63}$$

Although there are global similarities, our radial averaging is quite different. The discretization of Barbosa and Daube [6] seems to require the radial velocity u_r at $r = 0$ for $\mathbf{N}_{\theta_{1,j+\frac{1}{2},k}}$ and the angular vorticity component ω at $r = 0$ for $\mathbf{N}_{z_{1,j,k+\frac{1}{2}}}$, both of which are not defined there. In our discretization, these evaluations at $r = 0$ are resolved using weighted averaging that results in multiplication with the radial

coordinate r. Hence, at $r = 0$, any finite value can be assigned to these components as the resulting product always yields zero.

To demonstrate the improvement of the proposed scheme over the scheme of Barbosa and Daube [6], and in particular to emphasize the effect of the different approaches near the origin, two co-rotating vortices of unit circulation are simulated on a disc of radius $R = 8$ m. Their initial radial- and angular locations are $(\frac{1}{2}, \pi/10)$ and $(\frac{1}{2}, 11\pi/10)$ respectively. The grid is uniform in all directions for the comparison with 512 and 288 cells in radial- and angular direction respectively. The vortices are monitored and figure 2.10 shows their evolution in time for both discretizations in the vicinity of the origin. The results in the left column clearly show the development of a small disturbance around the coordinate origin, while in the right column the contour lines remain smooth. This would suggest that the proposed method leads to more stable and accurate results near the origin.

2.3.2 Discretization of the viscous term

For constant viscosity flows, the viscous part of the Navier-Stokes equations consists of the vector Laplacian $\nu\Delta\mathbf{u} = \nu\nabla^2\mathbf{u}$, where ν is the kinematic viscosity $\nu = \mu/\rho$. Since the vector Laplacian is a second order operator (divergence of a matrix), the term is rewritten using the identity:

$$\Delta\mathbf{u} = \nabla(\nabla \cdot \mathbf{u}) - \nabla \times \nabla \times \mathbf{u}, \tag{2.64}$$

which is a mere combination of first order operators. We make the assumption that the first term on the right-hand side of equation 2.64 can either be neglected in the discretization due to the fact that $\nabla \cdot \mathbf{u} = 0$ is discretely enforced locally in every computational cell, or it can be added to the pressure term. This results in the approximation for the viscous part $\mathbf{V}$ as:

$$\mathbf{V}(\mathbf{u}) = -\nu(\nabla \times \nabla \times \mathbf{u}) = -\nu(\nabla \times \boldsymbol{\omega}), \tag{2.65}$$

where $\boldsymbol{\omega}$ is the flow vorticity as defined in equation 2.56. The discrete approximation of $\mathbf{V}$ is $\mathbf{C}\overline{\mathbf{C}}\mathbf{u}$, where the discrete curl operators $\overline{\mathbf{C}} : \mathcal{HS} \rightarrow \mathcal{HL}$ and $\mathbf{C} : \mathcal{HL} \rightarrow \mathcal{HS}$ are used. No averaging is required, and the derivation of both operators in section 2.2 guarantees that no issues arise for the cells near the polar axis.

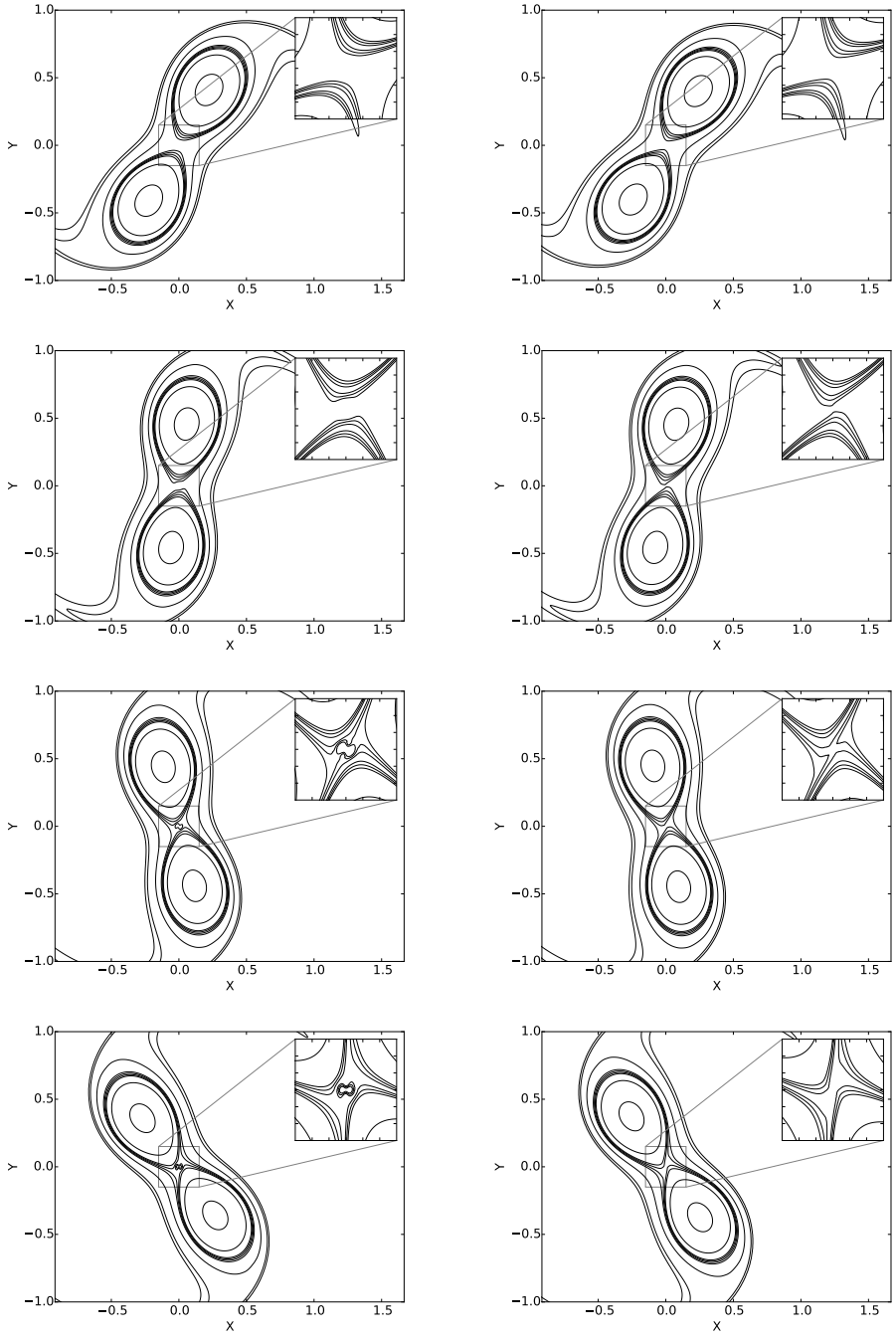


Figure 2.10: Evolution (from top to bottom) of the vorticity contours of two co-rotating vortices. The left column displays the results from the discretization of the non-linear terms as in Barbosa and Daube [6], while the right column displays the results from the proposed discretizations 2.58 - 2.60. Simulation performed by O. Daube.

2.3.3 Temporal discretization

The semi-discrete momentum equations and the continuity equation:

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{N}(\mathbf{u}) - \frac{1}{\rho}\overline{\mathbf{G}}p - \nu\mathbf{C}\overline{\mathbf{C}}\mathbf{u}, \quad \mathbf{D}\mathbf{u} = 0, \qquad \mathbf{u} \in \mathcal{HS},\ p \in HC, \tag{2.66}$$

are discretized in time using the Implicit Midpoint method:

$$\frac{\mathbf{u}^{(n+1)} - \mathbf{u}^{(n)}}{\Delta t} = \mathbf{N}\left(\mathbf{u}^{(n+\frac{1}{2})}\right) - \frac{1}{\rho}\overline{\mathbf{G}}p^{(n+\frac{1}{2})} - \nu\mathbf{C}\overline{\mathbf{C}}\mathbf{u}^{(n+\frac{1}{2})}, \quad \mathbf{D}\mathbf{u}^{(n+\frac{1}{2})} = 0, \tag{2.67}$$

where $\mathbf{u}^{(n+\frac{1}{2})} = \frac{1}{2}\left(\mathbf{u}^{(n)} + \mathbf{u}^{(n+1)}\right)$. Both the momentum- and continuity equations are solved in a coupled way to obtain the solution vector $\left(\mathbf{u}^{(n)}, p^{(n)}\right)$ using an iterative Krylov method. In particular, we iterate to obtain the solution of the *non-linear* equations. The Implicit Midpoint method used is second order accurate in time and unconditionally stable (even for vanishing viscosity), which is desirable as the CFL condition becomes very stringent near $r = 0$ if explicit methods were to be used. Additionally, if the non-linear coupled equations 2.67 are solved to machine precision, the method is capable of conserving kinetic energy. We will elaborate on the latter in section 2.4.3.

2.4 Conservation properties of the discretization

2.4.1 Conservation of mass

The conservation of mass is represented by the discrete continuity equation $\mathbf{D}\mathbf{u} = 0$. Since the discrete system is solved in a coupled way, mass conservation is determined by the accuracy of the solution of 2.67. As iterative methods are often used for this purpose, in practice this implies that the conservation of mass depends on the applied stop criterion of the iterative linear solver.

2.4.2 Conservation of momentum

Conservation of radial, angular and axial momentum requires the discrete evaluation of:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_V u_r \,\mathrm{d}V, \quad \frac{\mathrm{d}}{\mathrm{d}t}\int_V r u_\theta \,\mathrm{d}V, \quad \frac{\mathrm{d}}{\mathrm{d}t}\int_V u_z \,\mathrm{d}V, \tag{2.68}$$

respectively, with V denoting the entire finite cylindrical domain. Proving discrete

conservation of momentum has turned out to be complex due to two main reasons. In the first place, looking at the momentum equations 2.2-2.4, it is clear that the radial and angular momentum equations cannot be trivially written in a conservative formulation, with the time derivative governed solely by flux terms, due to the presence of additional terms that stem from the differentiation of the cylindrical covariant basis vectors which are not constant in space. There exist techniques to rewrite the momentum equations in conservative form (see Vinokur [93]), but they appear to be difficult to integrate into our mimetic approach. A conservative formulation of the governing equations is desirable because it often allows a discrete approximation of a similar form, which in turn leads to discrete conservation almost naturally with only the boundary values contributing. Without conservative formulation, it can be very difficult in practice to obtain conservation numerically, let alone to prove this. The axial momentum equation 2.4, which governs the axial component of the linear momentum, fortunately is in conservative form. Furthermore, multiplication of the angular momentum equation 2.3 with the radial coordinate r yields the conservative formulation for the quantity ru_θ:

$$\begin{aligned}\frac{\partial(ru_\theta)}{\partial t} + \frac{1}{r}\frac{\partial(r^2u_ru_\theta)}{\partial r} + \frac{1}{r}\frac{\partial(ru_\theta^2)}{\partial \theta} + \frac{\partial(ru_\theta u_z)}{\partial z} = \\ -\frac{1}{\rho r}\frac{\partial(rp)}{\partial \theta} + \frac{1}{\rho r}\frac{\partial(r^2\tau_{r\theta})}{\partial r} + \frac{1}{\rho r}\frac{\partial(r\tau_{\theta\theta})}{\partial \theta} + \frac{1}{\rho}\frac{(r\partial\tau_{\theta z})}{\partial z},\end{aligned} \tag{2.69}$$

which governs the evolution of the axial component of the angular momentum vector, which is a conserved quantity in the absence of any external torque applied to the z-axis. For the radial momentum equation, unfortunately, no procedure seems to exist that converts expression 2.2 into a conservative formulation. Nonetheless, we will demonstrate by numerical verification that our proposed discretization is capable of conserving radial, angular and axial momentum.

A second difficulty in our approach lies in the rotational formulation of the convective terms. This choice of formulation makes conservation of momentum much less obvious due to the fact that part of the convection is absorbed into an updated pressure variable $\tilde{p}$, which becomes $\tilde{p} = p + \frac{1}{2}\rho\mathbf{u}\cdot\mathbf{u}$. Analytically it holds that:

$$\boldsymbol{\omega}\times\mathbf{u} = \nabla\cdot(\mathbf{u}\mathbf{u}) - \mathbf{u}\nabla\cdot\mathbf{u} - \nabla\left(\frac{1}{2}\mathbf{u}\cdot\mathbf{u}\right), \tag{2.70}$$

where the first term on the right hand side is the conservative formulation and the second term involves the flow divergence (which hence vanishes). The third term

needs to be balanced by the new pressure $\tilde{p}$, but since this is done implicitly ($\tilde{p}$ is only defined in cell centres), it is impossible to do a term-by-term comparison and therefore to rigorously demonstrate momentum conservation. Equation 2.70, however, shows the dependence of the equality on the flow divergence, and this dependence is observed numerically as well.

For the discrete approximation of the quantities in expression 2.68, the Trapezoidal rule is used for the spatial integration. This yields the discrete quantities M_r, M_θ and M_z at time $t^{(n)}$ defined as:

$$M_r^{(n)} = \int_V u_r^{(n)} \,\mathrm{d}V \approx \sum_{\mathbf{I}} \frac{\Delta r_i \Delta\theta \Delta z_k}{2} \left(r_{i-\frac{1}{2}} u^{(n)}_{r_{\mathbf{I}-\mathbf{e}_r}} + r_{i+\frac{1}{2}} u^{(n)}_{r_{\mathbf{I}+\mathbf{e}_r}} \right), \tag{2.71}$$

$$M_\theta^{(n)} = \int_V r u_\theta^{(n)} \,\mathrm{d}V \approx \sum_{\mathbf{I}} \frac{r_i^2 \Delta r_i \Delta\theta \Delta z_k}{2} \left(u^{(n)}_{\theta_{\mathbf{I}-\mathbf{e}_\theta}} + u^{(n)}_{\theta_{\mathbf{I}+\mathbf{e}_\theta}} \right), \tag{2.72}$$

$$M_z^{(n)} = \int_V u_z^{(n)} \,\mathrm{d}V \approx \sum_{\mathbf{I}} \frac{r_i \Delta r_i \Delta\theta \Delta z_k}{2} \left(u^{(n)}_{z_{\mathbf{I}-\mathbf{e}_z}} + u^{(n)}_{z_{\mathbf{I}+\mathbf{e}_z}} \right). \tag{2.73}$$

We monitor the values of M_r, M_θ and M_z at every time step, and discrete conservation implies that $M_\alpha^{(n)} = M_\alpha^{(0)}$ for any value of $\alpha \in \{r, \theta, z\}$.

Some attention is required at the boundaries. For our purposes, we will only consider no-slip and free-slip (or stress-free) boundaries. The no-penetration condition $\mathbf{u} \cdot \mathbf{n} = 0$, where $\mathbf{n}$ is the unit normal to the wall, guarantees that the convective parts of the momentum equations do not contribute to any change in momentum. The contribution of the viscous part, which requires boundary conditions for the surrounding layer of ghost cells, is determined by the value of the vorticity at the walls, since:

$$\int_V (\nabla \times \boldsymbol{\omega}) \,\mathrm{d}V = \oint_{\partial V} (\mathbf{n} \times \boldsymbol{\omega}) \,\mathrm{d}S, \tag{2.74}$$

where $\mathbf{n}$ is the unit normal to the wall. For free-slip walls, we can enforce the right-hand side of equation 2.74 to vanish by choosing the discrete velocity boundary conditions in such a way that the resulting vorticity components at the wall are zero. This implies that at the wall $r = R$ both the angular and axial components ω and ζ of the discrete vorticity $\overline{\mathbf{C}}\mathbf{u}$ become zero. From equations 2.46 and 2.48, the boundary conditions for the angular and axial velocities then follow as:

$$u_{\theta_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_\theta}} = \frac{r_i}{r_{i+1}} u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}} \quad \text{and} \quad u_{z_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_z}} = u_{z_{\mathbf{I}+\mathbf{e}_z}}, \quad i = N_r. \tag{2.75}$$

Notice that during this derivation, the radial velocity u_r is considered to be zero due to the no-penetration condition. At the walls $z = \pm L/2$, the boundary values for the radial and angular velocities follow from equations 2.44 and 2.46:

$$u_{r_{\mathbf{I}+\mathbf{e}_r-2\,\mathbf{e}_z}} = u_{r_{\mathbf{I}+\mathbf{e}_r}} \quad \text{and} \quad u_{\theta_{\mathbf{I}+\mathbf{e}_\theta-2\,\mathbf{e}_z}} = u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}, \quad k = 1, \tag{2.76}$$

$$u_{r_{\mathbf{I}+\mathbf{e}_r+2\,\mathbf{e}_z}} = u_{r_{\mathbf{I}+\mathbf{e}_r}} \quad \text{and} \quad u_{\theta_{\mathbf{I}+\mathbf{e}_\theta+2\,\mathbf{e}_z}} = u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}, \quad k = N_z. \tag{2.77}$$

In this case, the axial velocity u_z is considered to be zero. Combined, equations 2.75-2.77 form the free-slip boundary conditions for the velocity, and they follow rather naturally from the construction of the discrete vorticity. Furthermore, with all vorticity components zero at the walls, proposition 2.2.2 holds on a finite domain due to the reasoning in section 2.2.4.

No-slip boundary conditions for finite difference methods are commonly derived by interpolation of the associated velocity to the wall, equating it to zero and subsequently obtaining a value for the velocity component in the ghost cell. We, however, will proceed along a different path, and derive expressions for the ghost values by instead demanding that proposition 2.2.2 remains valid on a finite domain, thereby assuring that the global mimetic structure of the discretization is not impaired. The procedure has in fact already been presented at the end of section 2.2.4 in the case of non-zero vorticity at the walls. We will repeat the resulting expressions here for completeness: at the wall at $r = R$, it holds that:

$$u_{z_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_z}} = -\frac{\Delta r_{i+1}}{\Delta r_i} u_{z_{\mathbf{I}+\mathbf{e}_z}}, \; u_{\theta_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_\theta}} = -\frac{r_i}{r_{i+1}} \frac{\Delta r_{i+1}}{\Delta r_i} u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}, \quad i = N_r, \tag{2.78}$$

while at the walls at $z = \pm L/2$:

$$u_{r_{\mathbf{I}+\mathbf{e}_r-2\,\mathbf{e}_z}} = -\frac{\Delta z_{k-1}}{\Delta z_k} u_{r_{\mathbf{I}+\mathbf{e}_r}}, \quad u_{\theta_{\mathbf{I}+\mathbf{e}_\theta-2\,\mathbf{e}_z}} = -\frac{\Delta z_{k-1}}{\Delta z_k} u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}, \quad k = 1, \tag{2.79}$$

$$u_{r_{\mathbf{I}+\mathbf{e}_r+2\,\mathbf{e}_z}} = -\frac{\Delta z_{k+1}}{\Delta z_k} u_{r_{\mathbf{I}+\mathbf{e}_r}}, \quad u_{\theta_{\mathbf{I}+\mathbf{e}_\theta+2\,\mathbf{e}_z}} = -\frac{\Delta z_{k+1}}{\Delta z_k} u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}, \quad k = N_z. \tag{2.80}$$

Notice that all but the condition for $u_{\theta_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_\theta}}$ at $r = R$ coincide with the linear interpolation on a non-uniform grid of the velocity component at the wall. The condition for $u_{\theta_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_\theta}}$ resembles the linear interpolation of the angular momentum ru_θ instead, and a closer look reveals that it is close to the linear interpolation of u_θ

since:

$$u_{\theta_{\mathbf{I}+2\mathbf{e}_r+\mathbf{e}_\theta}} = -\frac{r_i}{r_{i+1}}\frac{\Delta r_{i+1}}{\Delta r_i}u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}} = -\frac{\Delta r_{i+1}}{\Delta r_i}\left(1-\frac{\Delta r_i+\Delta r_{i+1}}{2r_{i+1}}\right)u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}, \tag{2.81}$$

for $i = N_r$. Hence, for sufficiently fine grids it is expected that the conventional linear interpolation of u_θ to the wall is found, but the presence of this extra term must be kept in mind for coarser grids. Finally, equations 2.78-2.80 constitute the no-slip boundary conditions for the velocity that assure that proposition 2.2.2 is valid on a finite domain.

2.4.3 Conservation of kinetic energy

Summation after taking the inner product of the momentum equations with their respective velocities gives the temporal evolution of the kinetic energy K:

$$\frac{\mathrm{d}K}{\mathrm{d}t} + (\mathbf{N}(\mathbf{u}),\mathbf{u})_{L^2(D)} + \frac{1}{\rho}(\mathbf{G}p,\mathbf{u})_{L^2(D)} - \nu(\mathbf{V}(\mathbf{u}),\mathbf{u})_{L^2(D)} = 0, \tag{2.82}$$

where $K = (\mathbf{u},\mathbf{u})_{L^2(D)}/2$ is the L^2-inner product of the velocity vector $\mathbf{u}$ over a suitable domain D, $\mathbf{N}$ is the skew-symmetric convective operator and $\mathbf{V}$ is the symmetric viscous operator. Because $\mathbf{N}$ is skew-symmetric, and with the gradient $\mathbf{G}$ as the formal (negative) adjoint of the divergence $\mathbf{D}$, the second and third terms of equation 2.82 vanish and the expression reduces to:

$$\frac{\mathrm{d}K}{\mathrm{d}t} = \nu(\mathbf{V}(\mathbf{u}),\mathbf{u})_{L^2(D)} = -\nu(\nabla\times\boldsymbol{\omega},\mathbf{u})_{L^2(D)}. \tag{2.83}$$

Thus, in the absence of viscosity and with appropriate boundary conditions, the kinetic energy is constant and therefore conserved. We will verify that the proposed discretization satisfies the same conditions, which leads to discrete conservation of kinetic energy as well. The first assumption that $\mathbf{G}$ is the negative adjoint of $\mathbf{D}$ is true by default, as this is precisely how the discrete operator $\mathbf{G}$ was constructed (see section 2.2). Secondly, it must be shown that the discretization of the convective part does not contribute to the change in kinetic energy.

Proposition 2.4.1. *The proposed discretization 2.58 - 2.60 for the convective part* $\mathbf{N}$ *of the Navier-Stokes equations assures that* $(\mathbf{N}(\mathbf{u}),\mathbf{u})_{\mathcal{HS}} = 0$.

Proof. After performing the inner product 2.30 over the entire computational domain, we collect all terms that contain the radial component of the vorticity $\eta_{\mathbf{I}+\mathbf{e}_\theta+\mathbf{e}_z}$

for any random choice of $\mathbf{I}$. This comprises the contributions to the inner product from cells $\mathbf{I}$, $\mathbf{I}+2\,\mathbf{e}_\theta$, $\mathbf{I}+2\,\mathbf{e}_z$ and $\mathbf{I}+2\,\mathbf{e}_\theta+2\,\mathbf{e}_z$. From these four cells, there are angular contributions from $\mathbf{N}_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}$ and $\mathbf{N}_{\theta_{\mathbf{I}+\mathbf{e}_\theta+2\,\mathbf{e}_z}}$ and axial contributions from $\mathbf{N}_{z_{\mathbf{I}+\mathbf{e}_z}}$ and $\mathbf{N}_{z_{\mathbf{I}+2\,\mathbf{e}_\theta+\mathbf{e}_z}}$, which sum up to:

$$
\begin{aligned}
&- C\left(\Delta z_k \overbrace{\frac{u_{z_{\mathbf{I}+\mathbf{e}_z}}+u_{z_{\mathbf{I}+2\,\mathbf{e}_\theta+\mathbf{e}_z}}}{2}}^{\text{from } \mathbf{N}_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}} u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}} + \Delta z_{k+1} \overbrace{\frac{u_{z_{\mathbf{I}+\mathbf{e}_z}}+u_{z_{\mathbf{I}+2\,\mathbf{e}_\theta+\mathbf{e}_z}}}{2}}^{\text{from } \mathbf{N}_{\theta_{\mathbf{I}+\mathbf{e}_\theta+2\,\mathbf{e}_z}}} u_{\theta_{\mathbf{I}+\mathbf{e}_\theta+2\,\mathbf{e}_z}}\right) \\
&+ C\left(\Delta z_k + \Delta z_{k+1}\right)\left(\overbrace{\frac{1}{2}\frac{\Delta z_k u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}+\Delta z_{k+1}u_{\theta_{\mathbf{I}+\mathbf{e}_\theta+2\,\mathbf{e}_z}}}{\Delta z_k+\Delta z_{k+1}}}^{\text{from } \mathbf{N}_{z_{\mathbf{I}+\mathbf{e}_z}}} u_{z_{\mathbf{I}+\mathbf{e}_z}}\right. \\
&\qquad\left.+\underbrace{\frac{1}{2}\frac{\Delta z_k u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}+\Delta z_{k+1}u_{\theta_{\mathbf{I}+\mathbf{e}_\theta+2\,\mathbf{e}_z}}}{\Delta z_k+\Delta z_{k+1}}}_{\text{from } \mathbf{N}_{z_{\mathbf{I}+2\,\mathbf{e}_\theta+\mathbf{e}_z}}} u_{z_{\mathbf{I}+2\,\mathbf{e}_\theta+\mathbf{e}_z}}\right),
\end{aligned}
\tag{2.84}
$$

where $C=\frac{1}{2}r_i\Delta r_i\Delta\theta$. It can be seen that these terms add up to zero.

Then we collect all terms that contain the angular component of the vorticity $\omega_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_z}$ for any random $\mathbf{I}$. This comprises the contributions to the inner product from cells $\mathbf{I}$, $\mathbf{I}+2\,\mathbf{e}_r$, $\mathbf{I}+2\,\mathbf{e}_z$ and $\mathbf{I}+2\,\mathbf{e}_r+2\,\mathbf{e}_z$. From these four cells, there are radial contributions from $\mathbf{N}_{r_{\mathbf{I}+\mathbf{e}_r}}$ and $\mathbf{N}_{r_{\mathbf{I}+\mathbf{e}_r+2\,\mathbf{e}_z}}$ and axial contributions from $\mathbf{N}_{z_{\mathbf{I}+\mathbf{e}_z}}$ and $\mathbf{N}_{z_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_z}}$, which sum up to:

$$
\begin{aligned}
&C\left(\Delta r_i+\Delta r_{i+1}\right)\left(\Delta z_k \overbrace{\frac{1}{2}\frac{\Delta r_i u_{z_{\mathbf{I}+\mathbf{e}_z}}+\Delta r_{i+1}u_{z_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_z}}}{\Delta r_i+\Delta r_{i+1}}}^{\text{from } \mathbf{N}_{r_{\mathbf{I}+\mathbf{e}_r}}} u_{r_{\mathbf{I}+\mathbf{e}_r}}\right. \\
&\qquad\left.+\Delta z_{k+1}\underbrace{\frac{1}{2}\frac{\Delta r_i u_{z_{\mathbf{I}+\mathbf{e}_z}}+\Delta r_{i+1}u_{z_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_z}}}{\Delta r_i+\Delta r_{i+1}}}_{\text{from } \mathbf{N}_{r_{\mathbf{I}+\mathbf{e}_r+2\,\mathbf{e}_z}}} u_{r_{\mathbf{I}+\mathbf{e}_r+2\,\mathbf{e}_z}}\right) \\
&- C\left(\Delta z_k+\Delta z_{k+1}\right)\left(\Delta r_i \overbrace{\frac{1}{2}\frac{\Delta z_k u_{r_{\mathbf{I}+\mathbf{e}_r}}+\Delta z_{k+1}u_{r_{\mathbf{I}+\mathbf{e}_r+2\,\mathbf{e}_z}}}{\Delta z_k+\Delta z_{k+1}}}^{\text{from } \mathbf{N}_{z_{\mathbf{I}+\mathbf{e}_z}}} u_{z_{\mathbf{I}+\mathbf{e}_z}}\right. \\
&\qquad\left.+\Delta r_{i+1}\underbrace{\frac{1}{2}\frac{\Delta z_k u_{r_{\mathbf{I}+\mathbf{e}_r}}+\Delta z_{k+1}u_{r_{\mathbf{I}+\mathbf{e}_r+2\,\mathbf{e}_z}}}{\Delta z_k+\Delta z_{k+1}}}_{\text{from } \mathbf{N}_{z_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_z}}} u_{z_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_z}}\right),
\end{aligned}
\tag{2.85}
$$

with $C = \frac{1}{2} r_{i+\frac{1}{2}} \Delta\theta$. After some algebraic manipulation, it follows that these terms also add up to zero.

Finally, we collect all terms that contain the axial component of the vorticity $\zeta_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_\theta}$ for any random $\mathbf{I}$, which comprises the contributions to the inner product from cells $\mathbf{I}$, $\mathbf{I}+2\,\mathbf{e}_r$, $\mathbf{I}+2\,\mathbf{e}_\theta$ and $\mathbf{I}+2\,\mathbf{e}_r+2\,\mathbf{e}_\theta$. From these four cells, there are radial contributions from $\mathbf{N}_{r_{\mathbf{I}+\mathbf{e}_r}}$ and $\mathbf{N}_{r_{\mathbf{I}+\mathbf{e}_r+2\,\mathbf{e}_\theta}}$ and angular contributions from $\mathbf{N}_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}$ and $\mathbf{N}_{\theta_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_\theta}}$, which sum up to:

$$
\begin{aligned}
&- \frac{C}{4} \frac{r_i + r_{i+1}}{2} \Bigg[\overbrace{\left(\frac{u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}}{r_i} + \frac{u_{\theta_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_\theta}}}{r_{i+1}} \right)}^{\text{from } \mathbf{N}_{r_{\mathbf{I}+\mathbf{e}_r}}} u_{r_{\mathbf{I}+\mathbf{e}_r}} \\
&\qquad + \underbrace{\left(\frac{u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}}{r_i} + \frac{u_{\theta_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_\theta}}}{r_{i+1}} \right)}_{\text{from } \mathbf{N}_{r_{\mathbf{I}+\mathbf{e}_r+2\,\mathbf{e}_\theta}}} u_{r_{\mathbf{I}+\mathbf{e}_r+2\,\mathbf{e}_\theta}} \Bigg] \\
&+ \frac{C}{4} \frac{r_i + r_{i+1}}{2} \Bigg[\overbrace{\frac{u_{r_{\mathbf{I}+\mathbf{e}_r}} + u_{r_{\mathbf{I}+\mathbf{e}_r+2\,\mathbf{e}_\theta}}}{r_i}}^{\text{from } \mathbf{N}_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}} u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}} \\
&\qquad + \underbrace{\frac{u_{r_{\mathbf{I}+\mathbf{e}_r}} + u_{r_{\mathbf{I}+\mathbf{e}_r+2\,\mathbf{e}_\theta}}}{r_{i+1}}}_{\text{from } \mathbf{N}_{\theta_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_\theta}}} u_{\theta_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_\theta}} \Bigg],
\end{aligned}
\tag{2.86}
$$

with $C = \frac{1}{2} r_{i+\frac{1}{2}} (\Delta r_i + \Delta r_{i+1}) \Delta\theta \Delta z_k$. It can be seen that these terms sum up to zero as well. Since this holds for all elements of the vorticity vector, the inner product yields exactly zero. □

The Implicit Midpoint time integration method applied to the semi-discrete Navier-Stokes without viscosity or external forces gives:

$$
\frac{\mathbf{u}^{(n+1)} - \mathbf{u}^{(n)}}{\Delta t} = -\mathbf{N}\left(\mathbf{u}^{(n+\frac{1}{2})}\right) - \overline{\mathbf{G}} p^{(n+\frac{1}{2})}, \qquad \mathbf{D}\mathbf{u}^{(n+\frac{1}{2})} = 0, \tag{2.87}
$$

with $\mathbf{u}^{(n+\frac{1}{2})} = \frac{1}{2}(\mathbf{u}^{(n)} + \mathbf{u}^{(n+\frac{1}{2})})$. Notice that the inner product of the left-hand side of equation 2.87 with $\mathbf{u}^{(n+\frac{1}{2})}$ yields:

$$
\left(\frac{\mathbf{u}^{(n+1)} - \mathbf{u}^{(n)}}{\Delta t}, \mathbf{u}^{(n+\frac{1}{2})} \right)_{\mathcal{HS}} = \frac{1}{2\Delta t} \left[\left(\mathbf{u}^{(n+1)}, \mathbf{u}^{(n+1)} \right)_{\mathcal{HS}} - \left(\mathbf{u}^{(n)}, \mathbf{u}^{(n)} \right)_{\mathcal{HS}} \right], \tag{2.88}
$$

while taking the inner product with the right-hand side gives:

$$-\left(\mathbf{N}(\mathbf{u}^{(n+\frac{1}{2})}),\mathbf{u}^{(n+\frac{1}{2})}\right)_{\mathcal{HS}}-\left(\overline{\mathbf{G}}p^{(n+\frac{1}{2})},\mathbf{u}^{(n+\frac{1}{2})}\right)_{\mathcal{HS}}=\left(p^{(n+\frac{1}{2})},\mathbf{D}\mathbf{u}^{(n+\frac{1}{2})}\right)_{HN} \quad (2.89)$$

by proposition 2.4.1 and the definition of the gradient operator $\overline{\mathbf{G}}$. Combining equations 2.88 and 2.89 then shows that:

$$\left(\mathbf{u}^{(n+1)},\mathbf{u}^{(n+1)}\right)_{\mathcal{HS}}-\left(\mathbf{u}^{(n)},\mathbf{u}^{(n)}\right)_{\mathcal{HS}}=2\Delta t\left(p^{(n+\frac{1}{2})},\mathbf{D}\mathbf{u}^{(n+\frac{1}{2})}\right)_{HN}. \quad (2.90)$$

If we define the kinetic energy at time t^n as $K^{(n)}:=\frac{1}{2}\left(\mathbf{u}^{(n)},\mathbf{u}^{(n)}\right)_{\mathcal{HS}}$, then:

$$\frac{K^{(n+1)}-K^{(n)}}{\Delta t}=\left(p^{(n+\frac{1}{2})},\mathbf{D}\mathbf{u}^{(n+\frac{1}{2})}\right)_{HN}, \quad (2.91)$$

and conservation of kinetic energy depends solely on the value of $\mathbf{D}\mathbf{u}$ in the domain. Hence, for vanishing flow divergence $\mathbf{D}\mathbf{u}=0$, it follows that the discrete kinetic energy is conserved.

For viscous flows, it is well known that the temporal decay of kinetic energy due to viscous dissipation on a suitable domain is given by:

$$\frac{\mathrm{d}K}{\mathrm{d}t}=-2\nu\mathcal{E}, \quad (2.92)$$

where $\mathcal{E}=\frac{1}{2}\int_V|\nabla\times\mathbf{u}|^2\,\mathrm{d}V$ is the flow enstrophy. Notice that this expression follows from applying the self-adjointness of the curl operator to the right-hand side of equation 2.83. At the discrete level, the fully discretized momentum equations without external forces read:

$$\frac{\mathbf{u}^{(n+1)}-\mathbf{u}^{(n)}}{\Delta t}=-\mathbf{N}\left(\mathbf{u}^{(n+\frac{1}{2})}\right)-\overline{\mathbf{G}}p^{(n+\frac{1}{2})}-\nu\mathbf{C}\overline{\mathbf{C}}\mathbf{u}^{(n+\frac{1}{2})}. \quad (2.93)$$

Taking the inner product with $\mathbf{u}^{(n+\frac{1}{2})}$ once more now yields:

$$\frac{K^{(n+1)}-K^{(n)}}{\Delta t}=-\nu\left(\mathbf{C}\overline{\mathbf{C}}\mathbf{u}^{(n+\frac{1}{2})},\mathbf{u}^{(n+\frac{1}{2})}\right)_{\mathcal{HS}}, \quad (2.94)$$

where we have used the fact that the convective term does not contribute to the evolution of the kinetic energy, and the pressure term vanishes because $\mathbf{D}\mathbf{u}=0$ as

noted above. By proposition 2.2.2, the term on the right-hand side is equal to:

$$\nu \left(\mathbf{C}\overline{\mathbf{C}}\mathbf{u}^{(n+\frac{1}{2})}, \mathbf{u}^{(n+\frac{1}{2})} \right)_{\mathcal{HS}} = \nu \left(\overline{\mathbf{C}}\mathbf{u}^{(n+\frac{1}{2})}, \overline{\mathbf{C}}\mathbf{u}^{(n+\frac{1}{2})} \right)_{\mathcal{HL}}, \tag{2.95}$$

which is the discrete representation of twice the enstrophy. Hence, with the flow divergence $\mathbf{Du}$ equal to zero, the resulting expression:

$$\frac{K^{(n+1)} - K^{(n)}}{\Delta t} = -2\nu \left(\overline{\mathbf{C}}\mathbf{u}^{(n+\frac{1}{2})}, \overline{\mathbf{C}}\mathbf{u}^{(n+\frac{1}{2})} \right)_{\mathcal{HL}}, \tag{2.96}$$

is the discrete analogue of expression 2.92.

2.5 Numerical validation

In this section, we will numerically demonstrate the conservation properties of the proposed discretization on both uniform and non-uniform grids (section 2.5.1). Furthermore, we will demonstrate the formal accuracy of the proposed method (section 2.5.2). For this, we use the Method of Manufactured Solutions, which we believe is among the most rigorous procedures to determine the general numerical accuracy.

2.5.1 Conservation properties

We are interested in the temporal evolution of mass, momentum and kinetic energy (in absence of viscosity). In sections 2.4.1 and 2.4.2 it was observed that conservation of mass and momentum are both determined by the accuracy of the solution of the continuity equation $\mathbf{Du} = 0$. In our simulations, we therefore set the stopping criterion for the iterative linear solver (based on the relative residual) to 10^{-15} to enforce conservation of mass and momentum up to machine precision. To investigate the energy conservation capacity of the discretization, we consider a flow in a cylinder of radius 1 m and length 1 m. The velocities are initialized as random numbers from the interval $[-\frac{1}{2}, \frac{1}{2}]$, and subsequently the flow field is made divergence-free by a projection step, where the initial pressure is also calculated. On the solid walls at $r = R$ and $z = \pm\frac{L}{2}$, a free-slip or stress-free boundary condition is imposed as described by equations 2.75-2.77. The free-slip condition assures that the vorticity at the wall is zero and therefore it does not contribute to any change in momentum. The fluid viscosity is set to zero, and we perform the simulation up to time $T = 10$ seconds, while observing the discrete mass, momentum and kinetic energy at every time step. A small time step Δt of 10^{-5} seconds is chosen purely to guarantee

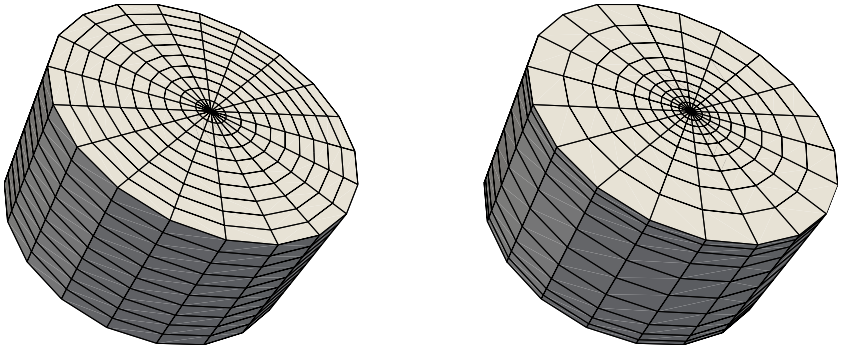
convergence of our current non-linear solver, as the conservation properties are independent of the magnitude of the time step.

To verify the proposed discretization on grids with non-uniform node distributions, the simulations are performed on two grids: a uniform grid, where $r_{i+\frac{1}{2}} = (i+\frac{1}{2})R/N_r$ and $z_{k+\frac{1}{2}} = -\frac{L}{2} + (k-\frac{1}{2})L/N_z$, and a grid with a non-uniform distributions defined by:

$$r_{i+\frac{1}{2}} = R\left(\frac{e^{\alpha i R/N_r} - 1}{e^{\alpha R} - 1}\right), \qquad i = 0, \ldots, N_r, \tag{2.97}$$

$$z_{k+\frac{1}{2}} = \frac{L}{2}\frac{\tanh(\beta i/N_z)}{\tanh(\beta L/2)}, \qquad k = 0, \ldots, N_z, \tag{2.98}$$

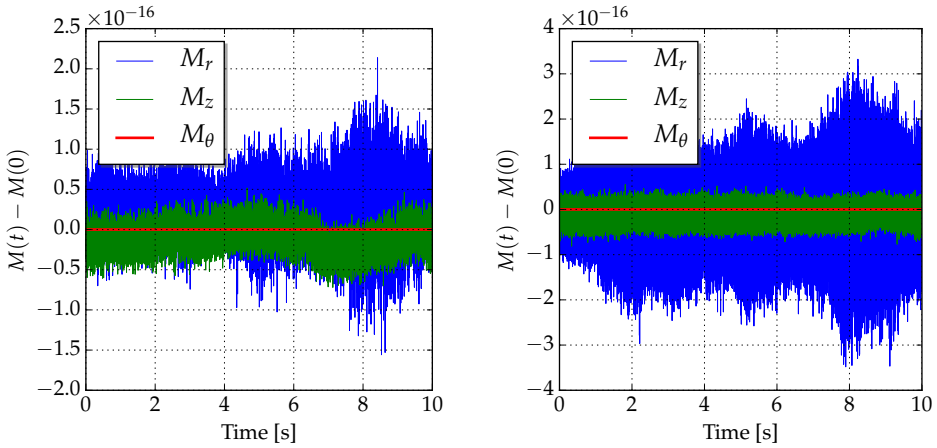
with $\alpha = 2$ and $\beta = 3$ as depicted in figure 2.11. Although perhaps not very applicable in practice, we have chosen this distribution in particular to demonstrate the validity of our conservation claims in the case of a severe variety in cell size throughout the computational domain. The number of grid cells are $10 \times 20 \times 10$ in radial, angular and axial direction, respectively.



(a) Uniform distribution. (b) Non-uniform distribution.

Figure 2.11: The uniform and non-uniform node distributions.

For the inviscid computations, we also calculate the resulting flow field using the spatial discretization of Morinishi et al. [61] for comparison. Because it is known that their axis treatment may introduce instabilities, we use the averaging procedure of Fukagata and Kasagi [33] for the radial velocity at $r = 0$, which does not conserve kinetic energy exactly. We find that the proposed method conserves mass, momentum and kinetic energy up to machine precision, as expected. Conservation of mass and momentum is obtained for the discretization technique of Morinishi et al. [61] (this is also proven in their paper). However, the kinetic energy is not



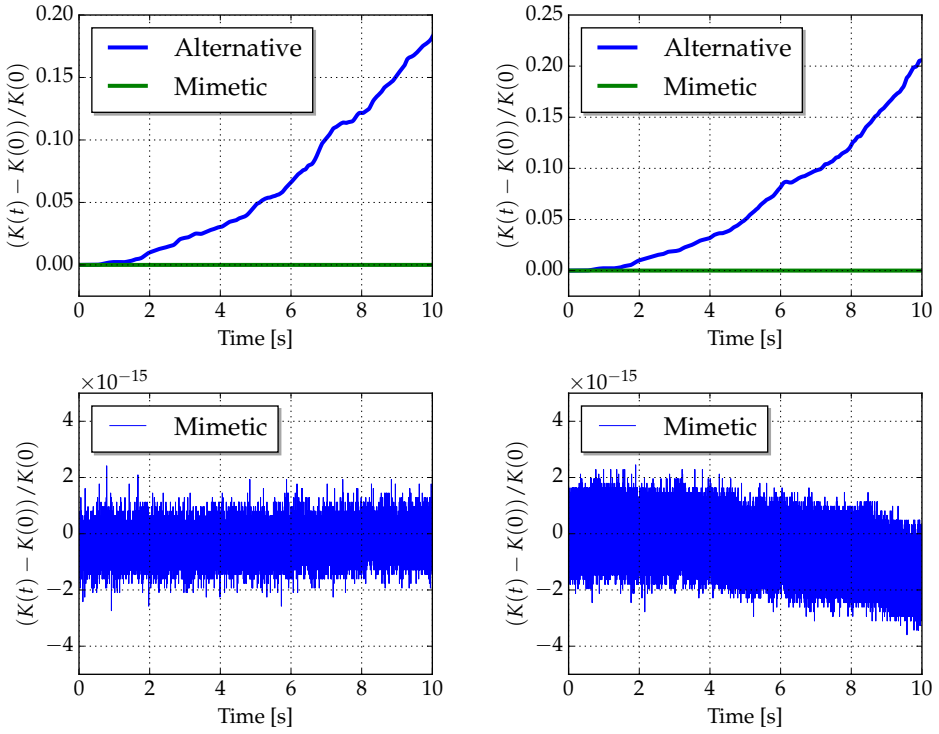
(a) Uniform grid.

(b) Non-uniform grid.

Figure 2.12: Momentum evolution on both uniform and non-uniform grid.

conserved in time, which is likely due to the handling of the radial velocity at the polar axis. First, figure 2.12 shows the evolution of the discrete radial, angular and axial momenta M_α of equations 2.71-2.73 using the proposed discretization. On both uniform and non-uniform grids, all three momenta are conserved to machine precision. The radial and axial momenta are of the order of machine precision initially, while the angular momentum maintains a larger non-zero value. Figure 2.13 then shows the kinetic energy during the simulation for both methods on the two grid types: the combined method of Morinishi et al. [61] and Fukagata and Kasagi [33] is referred to as the 'alternative' method.

For viscous flows, the kinetic energy decays at a rate determined by the flow enstrophy $\mathcal{E}$ as defined in section 2.4.3. To test this, we simulate a swirling flow with an angular velocity $u_\theta(r) = R(r - r^2)$. No-slip boundary conditions are applied for the velocity, and the dynamic viscosity has a value of $\mu = 0.01$ kg/m/s. Both the kinetic energy and the flow enstrophy are monitored in time. Figure 2.14 shows the monotone decay of the kinetic energy in time for both uniform and non-uniform grids based on the proposed discretization. We have used three levels of refinement to demonstrate that the uniform- and non-uniform results converge to the same rate of energy decay. As a consistency check, we have compared the energy decay in time at semi-integer time levels by explicitly calculating both sides of equation



(a) Uniform grid.

(b) Non-uniform grid.

Figure 2.13: Normalized kinetic energy evolution on both a uniform and a non-uniform grid. The mimetic results are shown separately in the lower graphs for a better impression of the actual magnitude.

2.96, i.e. the time derivative of the kinetic energy:

$$\frac{\mathrm{d}K}{\mathrm{d}t}(t^{n+\frac{1}{2}}) \approx \frac{K^{(n+1)} - K^{(n)}}{\Delta t}, \tag{2.99}$$

and the scaled flow enstrophy, calculated as:

$$-2\nu\mathcal{E}(t^{n+\frac{1}{2}}) \approx -2\nu \left(\overline{\mathbf{C}}\mathbf{u}^{(n+\frac{1}{2})}, \overline{\mathbf{C}}\mathbf{u}^{(n+\frac{1}{2})}\right)_{\mathcal{HL}}. \tag{2.100}$$

For the proposed approach, we find that the difference of the two terms is in the order of machine precision for both the uniform and the non-uniform grid.

2.5.2 Spatial accuracy

In order to demonstrate the formal accuracy of the proposed discretization, we utilize the Method of Manufactured Solutions (MMS) [74]. With a properly chosen

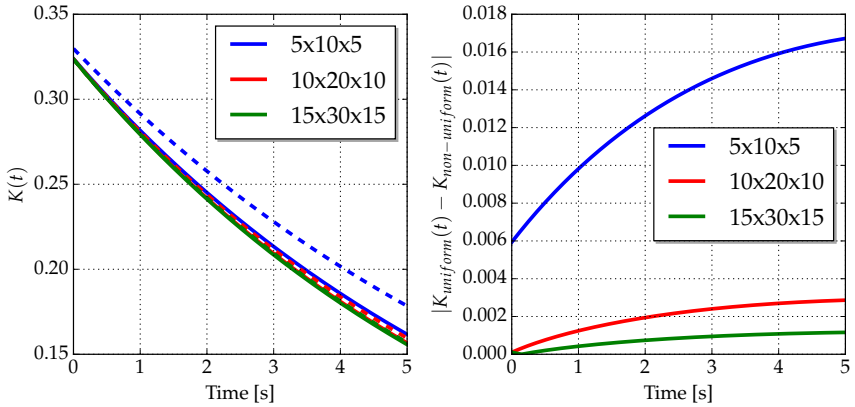


Figure 2.14: Kinetic energy decay for a viscous flow for different levels of grid refinement, and the error between the results on uniform (solid line) and non-uniform (dashed line) grids.

solution, the MMS is capable of testing virtually all terms of the discretization, and, if necessary, even individually. This is generally more challenging than the comparison with known exact solutions, which, if they exist in closed form at all, often follow from applying highly simplifying assumptions and therefore they may not involve all the terms in the Navier-Stokes equations.

One of the recommended properties of manufactured solutions is that they are sufficiently smooth, so that they do not prevent the theoretical order of accuracy to be obtained. Therefore, we choose the following time-dependent solutions for the velocity components and the pressure:

$$u_r(r,\theta,z,t) = (r-R)\left(z-\frac{L}{2}\right)\left(z+\frac{L}{2}\right)\cos(rz)\sin(\theta)\sin(t), \tag{2.101}$$

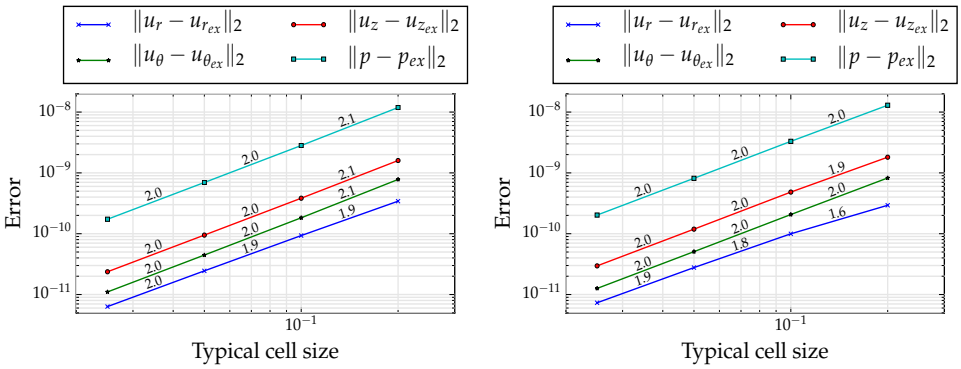
$$u_\theta(r,\theta,z,t) = (r-R)\left(z-\frac{L}{2}\right)\left(z+\frac{L}{2}\right)\sin(r+\theta+z)\sin(t), \tag{2.102}$$

$$u_z(r,\theta,z,t) = (r-R)\left(z-\frac{L}{2}\right)\left(z+\frac{L}{2}\right)\cos(r\theta z)\sin(t) \tag{2.103}$$

$$p(r,\theta,z,t) = \cos\left(\frac{2\pi r}{R}\right)\cos\left(\frac{2\pi\left(z+\frac{L}{2}\right)}{L}\right)\sin(t), \tag{2.104}$$

where $R = 1$ m is the cylinder radius and $L = 1$ m its length. The manufactured velocity solutions satisfy no-slip boundary conditions at the walls, while there is a non-zero radial flow through the origin. The pressure satisfies homogeneous Neu-

mann conditions at all walls. Using a symbolic computer algebra program, the solutions 2.101-2.104 are inserted in the Navier-Stokes equations 2.67, and the output is added to the right-hand side of the momentum equations. No-slip boundary conditions are applied, and initially all velocities and the pressure are set to zero. We then perform simulations until a certain time T on four grids with $(5 \times 10 \times 5)$, $(10 \times 20 \times 10)$, $(20 \times 40 \times 20)$ and $(40 \times 80 \times 40)$ cells. Both a uniform and a non-uniform grid as in equations 2.97 and 2.98 are used. Since the computations are unsteady, a very small time step of $\Delta t = 10^{-7}$ s is chosen to assure that the temporal error is negligible compared to the spatial error. The fixed time step is reduced for each grid refinement in order to keep the Courant number approximately constant.



(a) Global error on a uniform grid.

(b) Global error on a non-uniform grid.

Figure 2.15: Grid convergence of the proposed discretization using the MMS. For clarity, the pressure error is scaled down by a factor of 10^5.

The global error is determined by approximating the discrete L^2-norm at time T for all three velocity components by calculating:

$$\|u - u_{ex}\|_2 = \sqrt{\sum_{i,j,k} \left(u_{i,j,k}^{(T)} - u_{ex}(r_i, \theta_j, z_k, T) \right)^2 r_i \Delta r_i \Delta\theta \Delta z_k}, \tag{2.105}$$

and:

$$\|p - p_{ex}\|_2 = \sqrt{\sum_{i,j,k} \left(p_{i,j,k}^{(T)} - p_{ex}(r_i, \theta_j, z_k, T) \right)^2 r_i \Delta r_i \Delta\theta \Delta z_k}, \tag{2.106}$$

where $u_{i,j,k}^{(T)}$ and $p_{i,j,k}^{(T)}$ are the calculated solutions for any of the velocity components and the pressure at location (r_i, θ_j, z_k) at time T, u_{ex} and p_{ex} the associated exact solutions value and $r_i \Delta r_i \Delta\theta \Delta z_k$ the volume of the cell (i, j, k). The results for both uniform- and non-uniform grids are shown in figure 2.15, where the error against

the typical cell size is shown. For both uniform and non-uniform grids, the global error of all flow variables displays second order behaviour as the grid size goes to zero.

2.6 Concluding remarks

The aim of this study was to construct a spatial finite difference discretization of the cylindrical Navier-Stokes equations that conserves mass, momentum and kinetic energy, while simultaneously treating the classical problem near and at the origin $r = 0$. We have used the mimetic finite difference method to derive the vector identities like the gradient, the curl and the divergence. We have shown that the proposed averaging procedure is able to conserve mass and momentum and, in the absence of viscosity, also kinetic energy on both uniform and non-uniform grids. The mimetic approach combined with the averaging also yields a natural treatment of the singularity at $r = 0$. Accuracy tests show second order convergence in space on both grids with uniform and non-uniform node distribution.

For Large Eddy Simulations (LES) or Direct Numerical Simulations (DNS), it is likely that the overall second order accuracy of the proposed method is too restrictive. Therefore, future work should investigate the possibility of higher order approximations while maintaining the conservation properties.

The proposed spatial discretization alone is not sufficient for kinetic energy conservation. The time advancement of the semi-discrete equations is equally important, as any damping present may destroy the inherent conservation properties. In this chapter, we have used an implicit method known to conserve quadratic invariants to demonstrate the capabilities of the proposed spatial discretization. Besides being implicit, our approach hinges on the solution of the non-linear equations, which makes the computational procedure even more costly. In practice, both this and an implicit system may be unfeasible for large grids, and other (explicit) methods could be considered that (nearly) conserve energy; Verstappen and Veldman [91], for example, propose a class of time integration methods for this purpose. Here the trade-off in the method is clearly between computational cost and the desired level of conservation.

Acknowledgments

We thank Arthur Veldman for helpful suggestions. Olivier Daube is greatly acknowledged for fruitful discussions and for performing the rotating vortices test case in section 2.3.1.

Chapter 3

A dual interface capturing method for the simulation of incompressible immiscible two-phase pipe flows

In this chapter we present a dual interface capturing method for the simulation of incompressible immiscible two-phase flow in a cylindrical pipe with a circular cross section. The algorithm is developed as a research tool for the advancement of turbulence models for stratified two-phase flow and to improve the understanding of turbulence induced interface instabilities. To be able to perform computations on high resolution meshes, the algorithm is formulated to maximize (parallel) efficiency and accuracy in exchange for generic applicability. The cylindrical coordinates Navier-Stokes equations are discretized using a staggered finite difference method, while the model for the interface is based on the Mass Conserving Level Set (MCLS) method proposed in Van der Pijl et al. [90]. That method uses a dual formulation in terms of congruent Volume of Fluid (VoF) and Level Set (LS) fields. This approach combines the advantages of both the underlying VoF and LS methods: (nearly) exact mass conservation and an unambiguous definition of the interface location that can be used in turbulence models. The approach distinguishes itself from other dual approaches by cleverly avoiding an explicit reconstruction of the interface in the simultaneous advancement of both fields to reduce computational work. The algorithm is numerically validated through a set of well known benchmark cases, using experimental and numerical results from literature as well as reference solutions obtained by expert users of commercial and open-source simulation suites.

The content of this chapter is based on the article:

G.T. Oud, D.R. van der Heul, C. Vuik, and R.A.W.M. Henkes. *A dual interface capturing method for the simulation of incompressible immiscible two-phase pipe flows*, to be submitted.

3.1 Introduction

The numerical modeling of multiphase flow phenomena has been a highly active field of research for the past few decades. Examples of multiphase flows in industry and engineering are ubiquitous: boilers, condensers, reactors, mixers, separators and (cavitating) propellers are but some of the applications. In the oil and gas industry the increased effort for hydrocarbon recovery from more remote and deeper production fields has led to more complex pipeline transport. One often finds that both phases are transported through a single pipeline to an offshore platform or to an onshore plant. The liquid-gas mixture flowing through a pipeline can be in a specific flow regime, such as stratified flow or slug flow, depending on, among others, the liquid and gas flow rates and the pipe geometry. Unstable flows with slugs cause significant mechanical stresses on the pipe wall and its supports, and can also cause difficulties for the facilities at the end of the pipeline, such as flooding of the separator or trips of the compressor. It is therefore desirable to be able to predict what kind of conditions lead to slug flow, in both horizontal and inclined pipes.

3.1.1 Stratified multiphase pipe flow

To get a better understanding of the physical mechanisms underlying this transition process and to accurately model the influence of turbulence, a highly accurate and strictly mass conserving computational model will be developed specifically for the simulation of two-phase flow in a straight segment of a cylindrical pipe with a circular cross section. By limiting the applicability to this very simple geometry, we can optimize the accuracy and efficiency, at the cost of genericity, to enable very high resolution simulations (eventually DNS and LES). The flow is considered to be incompressible and isothermal and the two phases to be immiscible and separated by a sharp interface. Hence, the flow system can be accurately described by a two-phase flow model consisting of the variable density/viscosity Navier-Stokes equations and a separate model for the evolution of the interface. In this application, controlling numerical dissipation is very important to be able to distinguish physical from numerical amplification of interface disturbances. Simultaneously, the mass should be accurately conserved, as loss of (moving) mass generally implies loss of momentum (i.e. artificial damping). We believe that an optimal accuracy and (parallel) efficiency can be achieved by using a simple second order finite difference discretization for the cylindrical coordinate Navier-Stokes equations on an orthogo-

nal structured grid, together with a dedicated mass conserving, finite volume based interface model that exploits all symmetries and regularity of the control volumes on that same grid.

3.1.2 Dual interface capturing models

Nearly all interface models that do not impose any restrictions on changes in the topology of the interface are based on either the Level Set (LS) method [66] or the Volume of Fluid (VoF) method [41]. Individually, however, each suffers from drawbacks: the LS method is inherently incapable of conserving mass over time, while the VoF method, lacking an explicit interface, often requires complicated and computationally expensive interface reconstructions. Although the impact of most of these drawbacks can be reduced to a certain extent, yet another approach tries to combine both methods with the aim of benefiting from their advantages (the explicit interface of the LS method and the conservation property of the VoF method) while eliminating their disadvantages. Existing concepts involving the coupling of the level set method and the VoF method include Bourlioux [13], the CLSVOF algorithm [83], the VOSET algorithm [81] and the recent CLSMOF method [47], where besides the VoF also the centroids of the cells are included in the interface reconstruction. Because the interface is described by congruent LS and VoF fields, we refer to these methods as *dual* interface capturing methods.

Dual methods will outperform either VoF and LS methods, but none of the methods presented in the literature matches our requirements with respect to strict mass conservation and computational efficiency. Therefore, we have developed a new dual interface capturing method based on the Mass-Conserving Level Set (MCLS) method introduced by Van der Pijl et al. [90] for uniform Cartesian grids. A key feature of this method is the use of an invertible function that relates the VoF and LS fields, without a time-consuming explicit interface reconstruction that characterizes other dual methods. A more detailed review of the general MCLS algorithm can be found in section 3.3.1. The MCLS method was shown to be robust and to yield accurate results when compared to similar methods.

Although the results of the original Cartesian MCLS algorithm by Van der Pijl et al. [90] look encouraging, several of the sub-algorithms can be improved upon. This becomes clear in the recent comparison by Denner et al. [22], where the original MCLS algorithm slightly underperforms in the case of a rising bubble. Therefore,

our approach has been to only retain the original concept of the MCLS but formulate new sub-algorithms specifically for a structured tessellation of a cylindrical domain with circular cross section. Furthermore, we present a new formulation of a dimensionally split VoF advection scheme that is mass conserving to machine precision. Therefore, contrary to other dual methods mass is preserved exactly without the need for ad-hoc corrections. The algorithm is first verified and validated for solutions with symmetry with respect to the axis of symmetry. Although this is formally a simplification, many commonly used test cases for the assessment of multiphase flow models can in fact be performed in an axisymmetric setting. The algorithm is certainly not restricted to the axisymmetric case, and if the equations are carefully discretized, the pseudo-singularity at the centre of the cylinder does not lead to numerical problems [67].

3.1.3 Outline

We present an efficient dual interface model for a structured tessellation of cylindrical geometries with the following key features:

- The coupling between LS and VoF fields is likely to be more efficient than in other dual methods that require the solution of a system of equations (like Sussman and Puckett [83]) because of a relatively simple invertible functional relationship between the LS and the VoF fields.
- Individual volume fractions are conserved to machine precision without any need for volume truncation or complicated redistribution algorithms like in Van der Pijl et al. [90] or Sussman and Puckett [83]. This is achieved through the cylindrical implementation of a conservative dimensionally split algorithm for the VoF advection that nonetheless allows CFL numbers up to 0.5.
- Second order accuracy of the interface position is obtained in a non-trivial imposed flow field that is constant in time. This is one order of magnitude more accurate than Eulerian VoF schemes, which are generally limited to first order accuracy only [5].

We start in section 3.2 where the discretization of the Navier-Stokes equations in space and time is presented. Section 3.3 describes our adapted MCLS algorithm in a cylindrical geometry. The coupling procedure between the LS and the VoF is explained, and an updated interface advection algorithm which provides improved conservation properties when compared to the original scheme is presented. The

algorithm is tested and validated using a number of common benchmark cases in section 3.4. We conclude with a summary of the results and some closure remarks in section 3.5.

3.2 Calculation of the flow field

We consider the cylindrical domain:

$$\left\{ (r,\theta,z) \;\middle|\; 0 \le r \le R, 0 \le \theta < 2\pi, -\frac{L}{2} \le z \le \frac{L}{2} \right\}, \tag{3.1}$$

on which the axisymmetric Navier-Stokes equations are solved for the unknown velocity vector $\mathbf{u} = (u_r, u_\theta, u_z)$ and the pressure scalar field p. Under the assumption of rotational symmetry (i.e. all angular derivatives of any order vanish), the flow equations in conservative form are given by:

$$\frac{1}{r}\frac{\partial(ru_r)}{\partial r} + \frac{\partial u_z}{\partial z} = 0, \tag{3.2}$$

and:

$$\frac{\partial u_r}{\partial t} + \frac{1}{r}\frac{\partial(ru_r^2)}{\partial r} + \frac{\partial(u_r u_z)}{\partial z} - \frac{u_\theta^2}{r} = -\frac{1}{\rho}\frac{\partial p}{\partial r} + \frac{1}{\rho}\frac{\partial(r\tau_{rr})}{r\partial r} + \frac{1}{\rho}\frac{\partial \tau_{rz}}{\partial z} + g_r, \tag{3.3}$$

$$\frac{\partial u_\theta}{\partial t} + \frac{1}{r}\frac{\partial(ru_r u_\theta)}{\partial r} + \frac{\partial(u_\theta u_z)}{\partial z} + \frac{u_r u_\theta}{r} = \frac{1}{\rho}\frac{\partial(r\tau_{r\theta})}{r\partial r} + \frac{1}{\rho}\frac{\partial \tau_{\theta z}}{\partial z} + g_\theta, \tag{3.4}$$

$$\frac{\partial u_z}{\partial t} + \frac{1}{r}\frac{\partial(ru_r u_z)}{\partial r} + \frac{\partial(u_z^2)}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial z} + \frac{1}{\rho}\frac{\partial(r\tau_{rz})}{r\partial r} + \frac{1}{\rho}\frac{\partial \tau_{zz}}{\partial z} + g_z, \tag{3.5}$$

together with a set of boundary- and initial conditions. The viscous stresses are given by:

$$\tau_{rr} = 2\mu\frac{\partial u_r}{\partial r}, \quad \tau_{r\theta} = \mu\left(\frac{\partial u_\theta}{\partial r} - \frac{u_\theta}{r}\right), \quad \tau_{rz} = \mu\left(\frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r}\right), \tag{3.6}$$

$$\tau_{\theta\theta} = \frac{2\mu u_r}{r}, \quad \tau_{\theta z} = \mu\frac{\partial u_\theta}{\partial z}, \quad \tau_{zz} = 2\mu\frac{\partial u_z}{\partial z} \tag{3.7}$$

3.2.1 Spatial discretization

The axisymmetric Navier Stokes equations 3.2-3.5 are discretized in space using the second order accurate finite difference approach of Morinishi et al. [61]. The general presentation of this approach allows for non-uniform grids in all coordinate

directions, but for efficiency this has only been applied to the radial direction. In the absence of viscosity, this choice of discretization should, in addition to mass and momentum, also result in spatial conservation of kinetic energy, which is highly preferable for a possible future transition to turbulent flows. A Marker and Cell (MAC) arrangement of the flow unknowns according to Harlow and Welch [38] is used, in which the velocity components lie on the cell faces and the pressure is located in the cell centres (see figure 3.1). The staggered arrangement of the velocity components implies that there is a radial velocity located at $r = 0$. However, the rotational symmetry implies that at $r = 0$:

$$\frac{\partial u_r}{\partial r}(0, \theta, z) = 0. \tag{3.8}$$

Using central differences, it follows that in our discretization $u_r(0, \theta, z) = 0$ for all test cases.

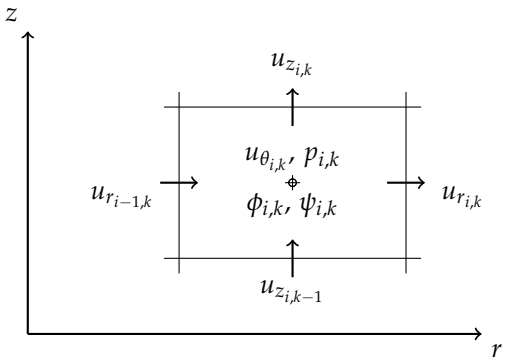


Figure 3.1: Staggered grid layout in axisymmetric domain: velocity components u_r and u_z at the cell faces, and u_θ lies orthogonal to the $r - z$ plane. The pressure p, level set function ϕ and VoF function ψ are located at the cell centre.

3.2.2 Temporal discretization

The advancement of the flow field in time requires the fluid densities ρ and viscosities μ, which in our approach are both determined using the interface variables (the level set function ϕ and the Volume of Fluid function ψ). The transport of these interface variables in turn depends on the underlying flow field, and hence there exists a two-way coupling in the advection of the flow field and the interface. Performing the advection for both simultaneously and hence implicitly is the only way to completely resolve the coupling issue. Aside from the complexity that

the implicit treatment introduces, however, this approach is generally too computationally demanding. Instead, the issue is remedied by staggering both the flow variables and the interface variables in time. In this way, the problem becomes decoupled at the cost of a temporal splitting error. The flow velocity $\mathbf{u}$ is evaluated at integer time levels $n, n+1$, etc, while the pressure p, the level set ϕ and the VoF value ψ are evaluated at semi-integer time levels $n+\frac{1}{2}, n+\frac{3}{2}$, etc. The flow field is advanced using a common projection scheme that consists of a predictor step in which an updated velocity field is calculated that may not be divergence-free. This newly obtained velocity field is subsequently made divergence-free using an updated pressure field that results from a Poisson equation.

The choice of a cylindrical computational grid imposes the potential problem that angular velocities close to $r = 0$ can cause a severe time step restriction when using explicit methods. The original MCLS algorithm contains a mixed IMEX time integration method, which, besides being only first order accurate, is unstable in the case of vanishing viscosity. Even though stable and efficient IMEX algorithms are readily available nowadays [48, 49], the time step restriction due to external blocks remains, and therefore the second order Implicit Midpoint integration method is used for the temporal advancement of the momentum equations as it provides unconditional stability without any artificial damping, even in the case of vanishing viscosity. The latter is a valuable property since many analytical results are based on inviscid assumptions. Therefore being able to do inviscid calculations will help the proper validation and verification of the implementation of the algorithms. The set of non-linear discrete momentum equations at time $t = t^{(n)}$:

$$\frac{\mathbf{u}^{(*)} - \mathbf{u}^{(n)}}{\Delta t} + f(\mathbf{u}^{(*)}, \rho^{(n+\frac{1}{2})}, \mu^{(n+\frac{1}{2})}) + \frac{1}{\rho^{(n-\frac{1}{2})}} G p^{(n-\frac{1}{2})} = \mathbf{0}, \tag{3.9}$$

where G is the discrete gradient operator, is linearized using Newton linearization, and solved by a preconditioned GMRES algorithm to find a tentative velocity $\mathbf{u}^{(*)}$. Subsequently, a Poisson equation is solved for an updated pressure $p^{(n+\frac{1}{2})}$ using an ICCG(0) method:

$$D\left(\frac{1}{\rho^{(n+\frac{1}{2})}} G p^{(n+\frac{1}{2})}\right) = \frac{1}{\Delta t} D\mathbf{u}^{(*)} + D\left(\frac{1}{\rho^{(n-\frac{1}{2})}} G p^{(n-\frac{1}{2})}\right), \tag{3.10}$$

where D is the discrete divergence operator. Finally, the tentative velocity $\mathbf{u}^{(*)}$ is

made divergence-free using the newly found pressure $p^{(n+\frac{1}{2})}$ through:

$$\mathbf{u}^{(n+1)} = \mathbf{u}^{(*)} - \Delta t \left(\frac{1}{\rho^{(n+\frac{1}{2})}} G p^{(n+\frac{1}{2})} - \frac{1}{\rho^{(n-\frac{1}{2})}} G p^{(n-\frac{1}{2})} \right), \tag{3.11}$$

which yields the new velocity vector $\mathbf{u}^{(n+1)}$ that satisfies $D\mathbf{u}^{(n+1)} = 0$. The Jacobi preconditioner used for the GMRES algorithm is a well balanced solution between performance and computational effort. For small to moderate time steps, the main diagonal of the momentum matrix is dominated by the value $1/\Delta t$, and preconditioning is hardly necessary as convergence is achieved after only a few iterations. Its effectiveness becomes apparent at larger time steps (which is possible due to the stability of the time integration method) when the diagonal dominance is lost. The simplicity and ease of implementation of the Jacobi preconditioner give the MCLS algorithm an overall good GMRES convergence rate at very low additional computational cost.

The density ρ required in the predictor equation is calculated as the volume-weighted average of the two (constant) densities ρ_0 and ρ_1 using the VoF value ψ:

$$\rho = \psi\rho_1 + (1 - \psi)\rho_0. \tag{3.12}$$

The viscosity μ follows from a step function $H_s = H_s(\phi(\mathbf{x}))$ that is smoothed using the level set function ϕ:

$$\mu = \mu_0 + H_s(\phi)(\mu_1 - \mu_0), \tag{3.13}$$

where:

$$H_s(\phi) = \begin{cases} 0, & \phi < -\alpha \\ \frac{1}{2}\left(1 + \sin\frac{\pi\phi}{2\alpha}\right) & -\alpha \leq \phi \leq \alpha \\ 1, & \phi > \alpha \end{cases} \tag{3.14}$$

For the band width α of the smoothed step function generally 1.5 cell widths are used. The value of the viscosity $\mu_{i,k}$ is calculated in the cell centre as a function of the local level set value $\phi_{i,k}$. Harmonic averaging based on the level set values is used when the value of the viscosity is needed at other locations than the cell centre.

3.2.3 Treatment of the interface jump conditions

The assumption of a sharp interface that separates two incompressible, immiscible fluids leads to jump conditions for the flow variables at the interface. Both density and viscosity are constant in each fluid phase, and the jump value of their difference normal to the interface is denoted by $[\rho]$ and $[\mu]$, respectively. Because only viscous flows are considered, the velocity is continuous across the interface and hence $[\mathbf{u}] = 0$. The pressure is not continuous but satisfies:

$$[p] = \sigma\kappa + 2[\mu]\mathbf{n}^T \cdot \nabla\mathbf{u} \cdot \mathbf{n}, \tag{3.15}$$

with σ the surface tension coefficient, κ the interface curvature and $\mathbf{n}$ the interface normal vector. The Ghost Fluid method [51] is used to implement the pressure jump as an additional term on the right hand side of the pressure Poisson equation 3.10. Although the density is modeled as a discontinuous quantity, the viscosity is smeared out over the interface using a smoothed Dirac function so that $[\mu] = 0$. The motivation for this is that the implementation of the pressure jump, which reduces to:

$$[p] = \sigma\kappa, \tag{3.16}$$

is greatly simplified. Contrary to the Continuous Surface Force (CSF) model [14], which is implemented in the original Cartesian MCLS algorithm, the Ghost Fluid method treats the interface as truly sharp, whereas the CSF model spreads out the interface over an arbitrary finite support. The accuracy of the pressure jump of the CSF model is limited to first order due to the smearing of the interface, while the Ghost Fluid method can obtain second order as verified by Francois et al. [32]: the latter is verified for our implementation in section 3.4.1. One of the objectives of our improved MCLS algorithm was to eliminate empirical parameters that could be case-specific for optimal performance. Although in practice the interface thickness in case of the CSF model is kept constant, the Ghost Fluid method requires no parameters at all and therefore has our preference.

3.3 Representation of the interface

The interface is represented by a linearized level set of a signed distance function. The level set method is a commonly used method in multiphase flow modeling because of its relatively simple implementation and its ability to naturally deal with changes in the topology of the interface. Its main drawback, however, is the in-

evitable loss of enclosed volume within the computational domain, and hence for multiphase flows with constant densities, this implies a loss of mass of one or of both fluids. This is due to the fact that the conservative treatment of the level set function does not imply conservation of volume within an iso-contour of that function. To counteract this phenomenon, generally high order level set representation and advection schemes are used. A different approach is to couple the level set method to the Volume of Fluid method, which is a method based on volume conservation. We will start by briefly summarizing both the level set method and the Volume of Fluid method.

Let $\mathbb{R}_{\geq 0}$ be the set of non-negative real numbers. Then at every time step, the level set function $\phi : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ in an arbitrary computational cell $\Omega \subset \mathbb{R}^n$ is represented by its spatial linearization around the cell centre $\mathbf{x}_0$:

$$\phi(\mathbf{x}, t) \approx \phi(\mathbf{x}_0, t) + \nabla\phi(\mathbf{x}_0, t) \cdot (\mathbf{x} - \mathbf{x}_0), \qquad \mathbf{x} \in \Omega, \tag{3.17}$$

and the linear interface $\Gamma \subset \mathbb{R}^{n-1} \times \mathbb{R}_{\geq 0}$ becomes the level set:

$$\Gamma(t) = \{\mathbf{x} \in \Omega \mid \phi(\mathbf{x}_0, t) + \nabla\phi(\mathbf{x}_0, t) \cdot (\mathbf{x} - \mathbf{x}_0) = 0\}. \tag{3.18}$$

The level set function is generally taken to be a signed distance function to the interface, although other choices have been successfully applied as well.

In addition, a discontinuous colour function $\chi : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \{0, 1\}$ is defined on Ω as:

$$\chi(\mathbf{x}, t) = \begin{cases} 0, & \mathbf{x} \in \text{ fluid 0 at time } t \\ 1, & \mathbf{x} \in \text{ fluid 1 at time } t \end{cases} \tag{3.19}$$

The Volume of Fluid (VoF) value ψ of a computational cell Ω is now defined as:

$$\psi_\Omega(t) = \frac{1}{|\Omega|} \int_\Omega \chi(\mathbf{x}, t) d\Omega. \tag{3.20}$$

From its definition, it is clear that $\psi_\Omega(t) \in [0, 1]$. In the finite difference discretization, both the level set function ϕ and the VoF function ψ are located in the cell centres. For clarity, any reference to the spatial location will be denoted by subscript indices while any reference to time will be denoted by superscript indices. Hence, the discrete analogues of $\phi(\mathbf{x}, t_n)$ and $\psi(\mathbf{x}, t_n)$ are $\phi_{i,k}^{(n)}$ and $\psi_{i,k}^{(n)}$.

3.3.1 General description of the MCLS algorithm

In this section, a brief description of the MCLS algorithm is given (see also Van der Pijl et al. [90]). Both the level set function $\phi^{(n-\frac{1}{2})}$ and the VoF function $\psi^{(n-\frac{1}{2})}$ are advected by a flow field $\mathbf{u}^{(n)}$ using the conservative advection algorithm of section 3.3.3 to become $\widetilde{\phi}$ and $\psi^{(n+\frac{1}{2})}$, respectively. Based on the knowledge that the volume enclosed by any iso-contour of $\widetilde{\phi}$ may significantly differ from the volume enclosed by the same iso-contour of $\phi^{(n-\frac{1}{2})}$, the MCLS algorithm essentially assures global mass conservation by altering the newly obtained level set function $\widetilde{\phi}$ such that locally (i.e. cell-wise) mass is conserved up to a user-defined tolerance. The VoF value is used as a reference, as it is known to be conservative by definition and through the use of a conservative advection scheme. As both the level set function and the VoF function are advected using a coupled directionally split algorithm, the value of $\widetilde{\phi}$ is usually a fairly good initial guess, and the changes to $\widetilde{\phi}$ are therefore often small so that the coupling can be done relatively fast. More specifically, given $\widetilde{\phi}$ and $\psi^{(n+\frac{1}{2})}$, a comparison is made in every cell of the computational domain between the VoF values that follow from the newly obtained level set $\widetilde{\phi}$ and the VoF values $\psi^{(n+\frac{1}{2})}$. The specific process of obtaining the associated VoF value from a level set value and its gradient in a computational cell is explained in section 3.3.2. Essentially, a function f has been formulated such that $\psi = f(\phi, \nabla\phi)$ exactly for any linear interface within a cell. Therefore, using this notation, during the comparison sweep the value of $|f(\widetilde{\phi}, \nabla\widetilde{\phi}) - \psi^{(n+\frac{1}{2})}|$, where the gradient $\nabla\widetilde{\phi}$ is obtained from central differences of $\widetilde{\phi}$, is calculated for every cell. If this value exceeds a user-defined value ϵ (the allowable volume fraction loss), the local value of $\widetilde{\phi}$ is adjusted iteratively until it complies with the prescribed tolerance, i.e. until:

$$|f(\widetilde{\phi}, \nabla\widetilde{\phi}) - \psi^{(n+\frac{1}{2})}| \leq \epsilon. \tag{3.21}$$

The adjustment of the level set field is a minimization problem in multiple coupled variables (both the level set value and its gradient) involving a non-linear function f. To solve it, an alternating method is used, in which initially the gradient is kept fixed while an updated level set value is found through:

$$\widetilde{\phi} = g\left(\psi^{(n+\frac{1}{2})}, \nabla\widetilde{\phi}\right), \tag{3.22}$$

where g is the inverse of the function f with respect to the first argument and consists of a robust scalar root finding algorithm. After the level set value has been updated throughout the domain, the global level set gradient is recalculated,

and a comparison sweep is re-initiated. This process is repeated until a global level set field with associated gradient is found that satisfies equation 3.21 in every computational cell, at which point the temporary level set $\widetilde{\phi}$ becomes the definitive level set $\phi^{(n+\frac{1}{2})}$ with its gradient $\nabla\phi^{(n+\frac{1}{2})}$. Figure 3.2 contains a schematic overview of the MCLS algorithm.

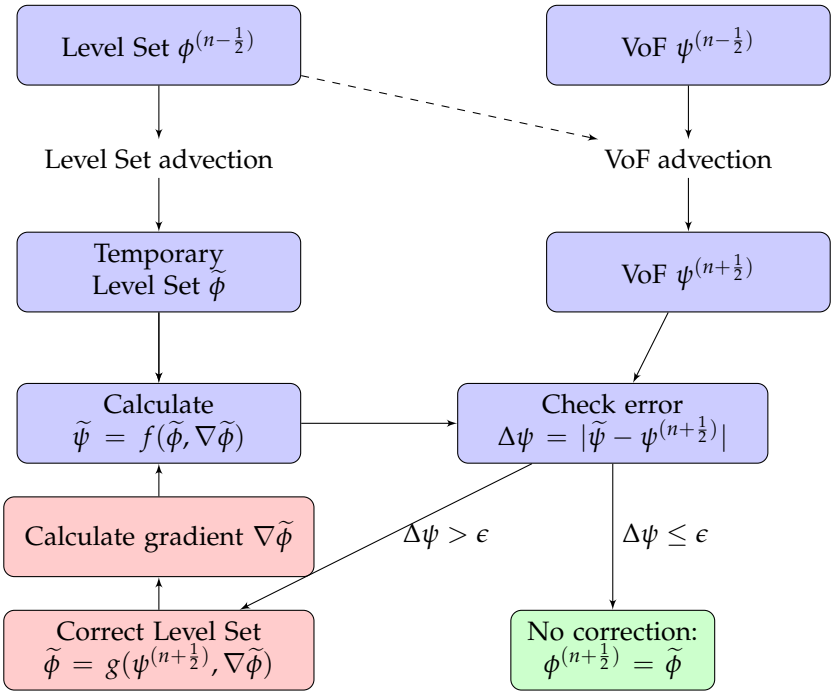


Figure 3.2: Schematic overview of the MCLS algorithm.

If $v^{(n+\frac{1}{2})} = \sum_i v_i^{(n+\frac{1}{2})}$ is the sum of the cell-wise volumes $v_i^{(n+\frac{1}{2})}$ of one of the two fluids present at time $t = t^{(n+\frac{1}{2})}$, then the volume loss of that fluid during the coupling procedure is bounded by:

$$
\begin{aligned}
|v^{(n+\frac{1}{2})} - v^{(n-\frac{1}{2})}| &\leq \sum_i |v_i^{(n+\frac{1}{2})} - v_i^{(n-\frac{1}{2})}| && (3.23)\\
&= \sum_i |f(\phi_i^{(n+\frac{1}{2})}, \nabla\phi_i^{(n+\frac{1}{2})}) - \psi_i^{(n+\frac{1}{2})}|\Omega_i && (3.24)\\
&\leq \epsilon \sum_i \Omega_i && (3.25)\\
&= \epsilon V, && (3.26)
\end{aligned}
$$

where Ω_i is the volume of the computational cell with index i and $V = \sum_i \Omega_i$ is the total volume of the cylinder. In practice, however, one only has to sum over the

interface cells which usually account for around 10% of the total number of cells. For the test cases in this chapter, the value of ϵ is set to 10^{-10}.

3.3.2 Coupling of the level set and VoF

For a given interface that is described by the zero level set of the function ϕ, the colour function χ from equation 3.20 can be defined as $\chi = H(\phi)$, and the associated VoF value $\psi_\Omega \in [0,1]$ belonging to some cell Ω can be calculated as:

$$\psi_\Omega = \frac{1}{|\Omega|} \int_\Omega H(\phi(\mathbf{x}))\, \mathrm{d}\mathbf{x}, \tag{3.27}$$

where H is the Heaviside function. For arbitrary ϕ this integral can generally not be evaluated in a straightforward way, but if ϕ is restricted to the subset of linear functions, then exact solutions can be readily found. More specifically, the VoF value ψ_Ω associated to a linear interface can be calculated as:

$$\psi_\Omega = \frac{1}{|\Omega|} \int_\Omega H\left[\phi(\mathbf{x}_0) + \nabla\phi(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0)\right] \mathrm{d}\mathbf{x}. \tag{3.28}$$

In the MCLS approach, this integral is evaluated analytically through a function f, so that $\psi_\Omega = f(\phi(\mathbf{x}_0), \nabla\phi(\mathbf{x}_0))$. For axisymmetric domains with $\mathbf{x}_0 = (r_0, z_0)$, the integral becomes:

$$\psi_\Omega = \frac{2\pi}{|\Omega|} \int_\Omega H\left[\phi(\mathbf{x}_0) + \phi_r(\mathbf{x}_0)(r - r_0) + \phi_z(\mathbf{x}_0)(z - z_0)\right] r\, \mathrm{d}r\, \mathrm{d}z, \tag{3.29}$$

and this is solved in a robust way using the expression of the centroid of a simple polygon, i.e.:

$$\psi_\Omega = \frac{2\pi}{6|\Omega|} \sum_{i=0}^{N-1} (r_i + r_{i+1})(r_i z_{i+1} - r_{i+1} z_i), \tag{3.30}$$

where N is the number of vertices (r_i, z_i) of the polygon that is enclosed by the linear interface and the cell edges, and $|\Omega| = 2\pi r_0 \Delta r \Delta z$ is the volume of the entire computational cell. The availability of the level set function greatly helps in finding the vertices of the enclose volume efficiently. Consider a computational cell (i,k) centred around the coordinates (r_i, z_k) as in figure 3.3.
The linear interface is constructed using the centred value $\phi_{i,k}$ and the first derivatives D_r and D_z that follow from central differences:

$$D_r = \frac{\phi_{i+1,k} - \phi_{i-1,k}}{\Delta r_i + \frac{1}{2}\left(\Delta r_{i-1} + \Delta r_{i+1}\right)}, \qquad D_z = \frac{\phi_{i,k+1} - \phi_{i,k-1}}{\Delta z_k + \frac{1}{2}\left(\Delta z_{k-1} + \Delta z_{k+1}\right)}. \tag{3.31}$$

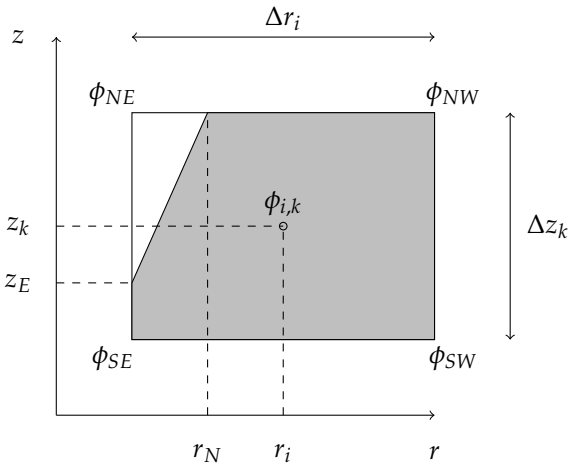


Figure 3.3: Calculation of the VoF value for a given level set field.

Then the linearized level set values at the corners of the cell, $\phi_{SE}, \phi_{SW}, \phi_{NW}$ and ϕ_{NE} are determined through:

$$\phi_{SE} = \phi_{i,k} - D_r \frac{\Delta r_i}{2} - D_z \frac{\Delta z_k}{2}, \qquad \phi_{SW} = \phi_{i,k} + D_r \frac{\Delta r_i}{2} - D_z \frac{\Delta z_k}{2}, \tag{3.32}$$

$$\phi_{NE} = \phi_{i,k} + D_r \frac{\Delta r_i}{2} + D_z \frac{\Delta z_k}{2}, \qquad \phi_{NE} = \phi_{i,k} - D_r \frac{\Delta r_i}{2} + D_z \frac{\Delta z_k}{2}. \tag{3.33}$$

Whether or not a cell contains an interface is now easily checked by considering the maximum and minimum of the four corner values. Furthermore, the vertices of the polygon on the cell edges are quickly found by linear interpolation. In figure 3.3, the radial coordinate r_N located on the northern cell edge is calculated as:

$$r_N = r_i - \left(\frac{1}{2} - \frac{|\phi_{NE}|}{|\phi_{NE}| + |\phi_{NW}|} \right) \Delta r_i. \tag{3.34}$$

Similarly, the axial coordinate on the western cell edge follows from:

$$z_W = z_k - \left(\frac{1}{2} - \frac{|\phi_{SE}|}{|\phi_{SE}| + |\phi_{NE}|} \right) \Delta z_k. \tag{3.35}$$

The set of vertices is subsequently used in equation 3.30 to find the corresponding volume. Finally, the signs of the corner values are used to determine whether the correct polygon volume is calculated, or whether the complement is actually required. This relatively simple sequence of calculations, with only the four cor-

ner LS values required, results in a cheap function f such that $\psi = f(\phi, \nabla\phi)$ as described in section 3.3.1. Because of this, its inverse $g = f^{-1}$ from equation 3.22 that is needed to locally correct the level set value $\phi_{i,k}$ can be obtained iteratively in an very efficient way compared to methods that require the solution of a system of equations (least squares-based) to obtain the interface.

3.3.3 Advection of the interface

Both the level set function ϕ and the VoF function ψ are advected with the underlying flow field $\mathbf{u}$:

$$\frac{\partial\phi}{\partial t} + \mathbf{u}\cdot\nabla\phi = 0, \qquad \frac{\partial\psi}{\partial t} + \mathbf{u}\cdot\nabla\psi = 0. \tag{3.36}$$

For both processes, the second order directional split algorithm of Weymouth and Yue [94] is used. For the level set ϕ centred in cell (i,k), it is given by:

$$\frac{\phi^{(*)} - \phi^{(n)}}{\Delta t} = \frac{(r\widetilde{F}^{(n)})_{i-\frac{1}{2}} - (r\widetilde{F}^{(n)})_{i+\frac{1}{2}}}{r_i\Delta r} + \phi^{(n)}\frac{(ru_r^{(n)})_{i+\frac{1}{2}} - (ru_r^{(n)})_{i-\frac{1}{2}}}{r_i\Delta r}, \tag{3.37}$$

$$\frac{\phi^{(n+1)} - \phi^{(*)}}{\Delta t} = \frac{\widetilde{G}^{(*)}_{k-\frac{1}{2}} - \widetilde{G}^{(*)}_{k+\frac{1}{2}}}{\Delta z} + \phi^{(n)}\frac{u^{(n)}_{z_{k+\frac{1}{2}}} - u^{(n)}_{z_{k-\frac{1}{2}}}}{\Delta z}, \tag{3.38}$$

with $\widetilde{F}_{i\pm\frac{1}{2}}$ and $\widetilde{G}_{k\pm\frac{1}{2}}$ the scalar level set fluxes in radial and axial direction, respectively, and the time level $t^{(*)}$ an intermediary time level between $t^{(n)}$ and $t^{(n+1)}$. The fluxes are determined in a similar way to the upwind approach of Sussman and Puckett [83]: for cell (i,k), the flux $\widetilde{F}_{i+\frac{1}{2}}$ at the positive face at $r_{i+\frac{1}{2}}$ is given by:

$$\widetilde{F}_{i+\frac{1}{2}} = \begin{cases} u_{r_{i+\frac{1}{2}}}\left(\phi_{i,k} + \frac{1}{2}\left(\Delta r - u_{r_{i+\frac{1}{2}}}\Delta t\right)\right)\frac{\phi_{i+1,k} - \phi_{i-1,k}}{2\Delta r}, & u_{r_{i+\frac{1}{2}}} \geq 0 \\ u_{r_{i+\frac{1}{2}}}\left(\phi_{i+1,k} - \frac{1}{2}\left(\Delta r + u_{r_{i+\frac{1}{2}}}\Delta t\right)\right)\frac{\phi_{i+2,k} - \phi_{i,k}}{2\Delta r}, & u_{r_{i+\frac{1}{2}}} < 0, \end{cases} \tag{3.39}$$

while for the axial direction the flux $\widetilde{G}_{k+\frac{1}{2}}$ is given by:

$$\widetilde{G}_{k+\frac{1}{2}} = \begin{cases} u_{z_{k+\frac{1}{2}}}\left(\phi_{i,k} + \frac{1}{2}\left(\Delta z - u_{z_{k+\frac{1}{2}}}\Delta t\right)\right)\frac{\phi_{i,k+1} - \phi_{i,k-1}}{2\Delta z}, & u_{z_{k+\frac{1}{2}}} \geq 0 \\ u_{z_{k+\frac{1}{2}}}\left(\phi_{i,k+1} - \frac{1}{2}\left(\Delta z + u_{z_{k+\frac{1}{2}}}\Delta t\right)\right)\frac{\phi_{i,k+2} - \phi_{i,k}}{2\Delta z}, & u_{z_{k+\frac{1}{2}}} < 0 \end{cases} \tag{3.40}$$

For the VoF function, we denote the radial and axial fluxes by $F_{i\pm\frac{1}{2}}$ and $G_{k\pm\frac{1}{2}}$,

respectively, and the advection scheme is given by:

$$\psi^{(*)} - \psi^{(n)} = \frac{1}{|\Omega|}\left(F^{(n)}_{i-\frac{1}{2}} - F^{(n)}_{i+\frac{1}{2}}\right) + H(\phi^{(n)})\Delta t \frac{(ru_r^{(n)})_{i+\frac{1}{2}} - (ru_r^{(n)})_{i-\frac{1}{2}}}{r_i \Delta r}, \tag{3.41}$$

$$\psi^{(n+1)} - \psi^{(*)} = \frac{1}{|\Omega|}\left(G^{(*)}_{k-\frac{1}{2}} - G^{(*)}_{k+\frac{1}{2}}\right) + H(\phi^{(n)})\Delta t \frac{u^{(n)}_{z_{k+\frac{1}{2}}} - u^{(n)}_{z_{k-\frac{1}{2}}}}{\Delta z}, \tag{3.42}$$

with H the Heaviside function and $|\Omega| = r_i \Delta r \Delta z$ the volume of the cell. The time-explicit second term on the right hand side guarantees that volume is conserved up to machine precision as long as the CFL number of the flow field, calculated as $\Delta t(u_r/\Delta r + u_z/\Delta z)$, remains less than 0.5. After application of the divergence theorem, the radial VoF flux $F^{(n)}_{i+\frac{1}{2}}$ becomes:

$$F^{(n)}_{i+\frac{1}{2}} = r_{i+\frac{1}{2}} \int_{t_n}^{t_n+\Delta t} \int_{z_{k-\frac{1}{2}}}^{z_{k+\frac{1}{2}}} \chi(r_{i+\frac{1}{2}}, z, t) u_r(r_{i+\frac{1}{2}}, z, t)\, \mathrm{d}z\, \mathrm{d}t, \tag{3.43}$$

where χ is the colour function. To rewrite the time integral into a spatial integral, it is observed that the continuity equation restricted to the radial direction:

$$\frac{1}{r}\frac{\partial}{\partial r}(ru_r) = 0, \quad r > 0, \tag{3.44}$$

implies that, at some fixed time $t \in [t_n, t_n + \Delta t]$, the characteristics are curved and satisfy:

$$r(t) u_r(r, t) = r_{i+\frac{1}{2}} u_r(r_{i+\frac{1}{2}}, z, t). \tag{3.45}$$

Integration then yields:

$$\int_{r(t)}^{r_{i+\frac{1}{2}}} r\, \mathrm{d}r = \int_t^{t_n+\Delta t} r_{i+\frac{1}{2}} u_r(r_{i+\frac{1}{2}}, z, \tau)\, \mathrm{d}\tau, \tag{3.46}$$

so that:

$$r^2(t) = r^2_{i+\frac{1}{2}} - 2\int_t^{t_n+\Delta t} r_{i+\frac{1}{2}} u_r(r_{i+\frac{1}{2}}, z, \tau)\, \mathrm{d}\tau, \tag{3.47}$$

for some $t \in [t_n, t_n + \Delta t]$. If we now assume $u_r(r_{i+\frac{1}{2}}, z, t)$ to be constant in time during the interval $[t_n, t_n + \Delta t]$ and equal to $u_r(r_{i+\frac{1}{2}}, z, t_n)$, then the integral reduces to:

$$r^2(t) = r^2_{i+\frac{1}{2}} - 2 r_{i+\frac{1}{2}} u_r(r_{i+\frac{1}{2}}, z, t_n)(t_n + \Delta t - t). \tag{3.48}$$

A change of variable finally allows us to rewrite expression 3.43 as:

$$F^{(n)}_{i+\frac{1}{2}} = \int_{z_{k-\frac{1}{2}}}^{z_{k+\frac{1}{2}}} \int_{r^*}^{r_{i+\frac{1}{2}}} \chi(r,z,t_n) r \,\mathrm{d}r \,\mathrm{d}z, \tag{3.49}$$

with:

$$r^* = \sqrt{r_{i+\frac{1}{2}} \left(r_{i+\frac{1}{2}} - 2u_r(r_{i+\frac{1}{2}},z,t_n)\Delta t \right)}. \tag{3.50}$$

The flux $F^{(n)}_{i+\frac{1}{2}}$ is subsequently calculated by substitution of $H(\phi)$ for the colour function χ and the techniques described in section 3.3.2. The resulting expression for $F^{(n)}_{i+\frac{1}{2}}$ appears to be different from the radial VoF flux proposed in Sussman and Puckett [83]. In our approach, the dimension of the volume that is fluxed is based on the divergence-free velocity per flux direction which is constant in time but not in space. In Sussman and Puckett [83], the radial velocity used for the flux is considered constant both in time and space, i.e.:

$$u_r(r,z,t) = u_r(r_{i+\frac{1}{2}},z,t) \qquad \text{for any } r \in [r_{i-\frac{1}{2}}, r_{i+\frac{1}{2}}], \tag{3.51}$$

which yields:

$$r^* = r_{i+\frac{1}{2}} - u_r(r_{i+\frac{1}{2}},z,t_n)\Delta t. \tag{3.52}$$

The axial VoF flux is given by:

$$G^{(*)}_{k+\frac{1}{2}} = \int_{t_n}^{t_n+\Delta t} \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \chi(r,z_{k+\frac{1}{2}},t) u_z(r,z_{k+\frac{1}{2}},t) r \,\mathrm{d}r \,\mathrm{d}t, \tag{3.53}$$

and since the continuity equation restricted to the axial direction is:

$$\frac{\partial u_z}{\partial z} = 0, \tag{3.54}$$

the resulting VoF flux based on a divergence-free velocity field is in this case equal to the flux from Sussman and Puckett [83]:

$$G^{(*)}_{k+\frac{1}{2}} = \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \int_{z^*}^{z_{k+\frac{1}{2}}} \chi(r,z,t_n) r \,\mathrm{d}z \,\mathrm{d}r, \tag{3.55}$$

where:

$$z^* = z_{k+\frac{1}{2}} - u_z(r,z_{k+\frac{1}{2}},t_n)\Delta t. \tag{3.56}$$

To verify the correct implementation and accuracy of the interface advection scheme,

a torus is placed in a divergence-free velocity field $(u_r, u_\theta, u_z) = (r, 0, -2(z+2))$ that is constant in time. By choosing a velocity field that is exactly represented at the discrete level, it is guaranteed that any errors that arise stem from the interface advection scheme alone. As pointed out by Weymouth and Yue [94], it is necessary to have a flow field that contains shear to show the improved behaviour of the implemented advection scheme compared to other non-strictly-conserving schemes. The geometry is shown in figure 3.4, where the calculated shape of the cross section of the torus is demonstrated at a number of time instances.

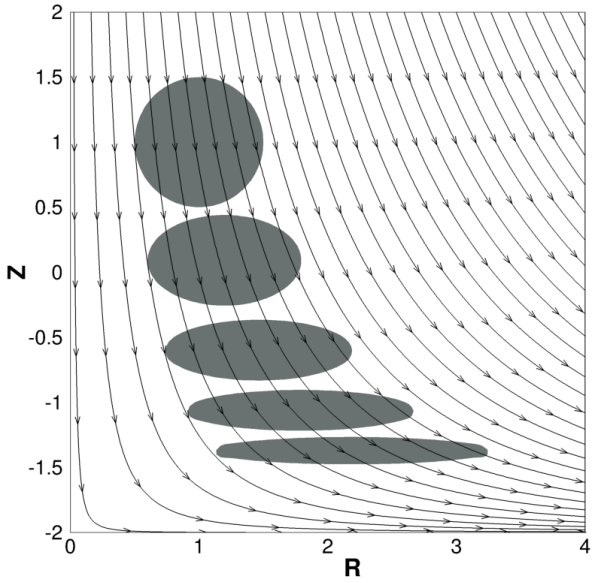


Figure 3.4: The advection of a torus by a linear divergence-free velocity field.

The accuracy of the spatial fluxes is obtained by comparing the cell-wise VoF values to the 'exact' values at some predefined time $t^{(*)}$. The exact VoF values are obtained using the exact solution of the velocity field, and several refinements are used to approximate the exact cell-wise VoF values. The error:

$$\|\psi(t^*) - \psi_{ex}(t^*)\|_1 = \sum_{i,k} |\psi_{i,k}^{(*)} - \psi_{ex_{i,k}}^{(*)}| r_i \Delta r \Delta z, \tag{3.57}$$

is shown in figure 3.5, and demonstrates that the advection scheme has second order accuracy in space. Additionally, the volume loss per time step is analyzed for three different values of the maximal allowable volume loss parameter ϵ (10^{-4}, 10^{-7} and 10^{-10}). For all cases it it observed that during a simulation of sufficient temporal length, the actual volume loss per time step correctly satisfies the bound

in 3.26.

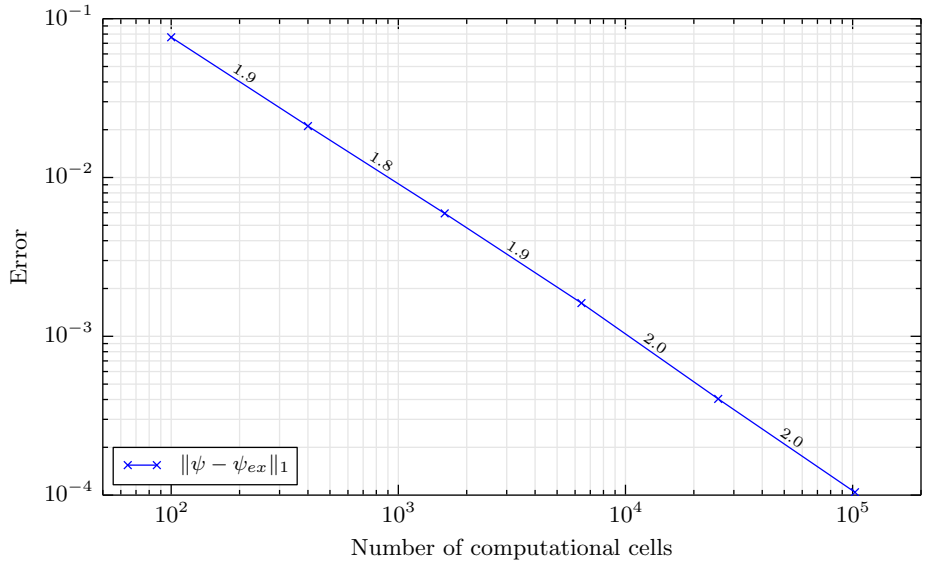


Figure 3.5: The accuracy of the VoF field with grid refinement. Second order convergence is obtained in the case of a torus in an axisymmetric corner flow.

Conservation of mass (equivalent to conservation of volume in the incompressible case) in a simulation model is of paramount importance for the results to have a clear physical meaning. Generic VoF methods frequently suffer from volume loss during advection; even if the advection scheme is conservative, there is no intrinsic mechanism that prevents computational cells from overflowing (VoF greater than one) or from draining (VoF less than zero). These nonphysical occurrences need to be subsequently corrected through processes that usually either change the total volume (by adding or removing volume) or redistribute the total volume to eliminate invalid VoF values. The advection scheme used in our algorithm is to our knowledge the only split flux scheme that is conservative and which simultaneously prevents nonphysical VoF values up to rounding errors.

3.3.4 Level set reinitialization

Due to dissipation during advection, the level set function ϕ quickly loses its signed distance property $\|\nabla\phi\| = 1$. This property is desirable as it generally guarantees a nicely behaving level set function with accurate finite difference results when, for example, approximating its gradient or curvature. Additionally, the determination

of the regularized fluid viscosities depends on the signed distance property. A reinitialization process is therefore performed after every few iterations with the aim of reconstructing the level set function ϕ to a distance function by having its gradient satisfy $\|\nabla\phi\| = 1$. This is done according to the method proposed by Sussman et al. [84], and involves solving the following evolution equation at time t_n:

$$\frac{\partial\phi}{\partial\tau} = \text{sign}(\phi_n)\left(1 - \|\nabla\phi\|_2\right), \quad \phi(0) = \phi_n, \tag{3.58}$$

in which τ is a pseudo time. This equation is integrated over this pseudo time until a steady state solution is obtained. This is accomplished by rewriting the equation to a hyperbolic transport equation:

$$\frac{\partial\phi}{\partial\tau} + \mathbf{q}\cdot\nabla\phi = \text{sign}(\phi_n), \quad \text{with} \quad \mathbf{q} = \text{sign}(\phi_n)\frac{\nabla\phi}{\|\nabla\phi\|_2}, \tag{3.59}$$

and it is solved using first order ENO spatial fluxes [77] and Euler forward integration in time. The sub-cell fix of Russo and Smereka [73] is applied to make the fluxes truly upwind and therefore to reduce any movement of the interface. Since the advection velocity is unity, the number of time steps for a given CFL number is chosen such that the level set is reinitialized in a neighbourhood with a radius of about five cells around the interface. This 'narrow band' approach significantly reduces the required computational effort compared with reinitialization of the level set function on the whole domain.

3.3.5 Calculation of the interface curvature

The interface conditions described in section 3.2.3 require the curvature of the interface. As demonstrated by, among others, Francois et al. [32], the accuracy of most surface tension-dominated computations is generally more dependent on the accuracy of the interface curvature calculation than on the surface tension model used. Having both a level set and a VoF function available, there are two obvious ways to compute the interface curvature. The curvature κ based on the level set ϕ is relatively straightforward to determine in cell centres by using a finite difference approximation of:

$$\kappa(\phi) = \nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right). \tag{3.60}$$

The actual interface curvature is found by an approximation that uses the cell centred curvature values. In the original MCLS algorithm, the curvature is calculated

in this way. Despite its apparent simplicity, however, the method generally yields a rather non-uniform interface curvature field unless advanced methods are used to obtain the interface curvature from the cell centre values. This non-uniformity is a major source of parasitic currents (see section 3.4.1). Improved approximation methods involve, among others, smoothing of the obtained interface curvature by a diffusive Poisson process [90], approximating the interface through curve fitting [29, 58], applying higher order spatial fluxes [25] and combining Front Tracking and Level Set techniques [64]. Furthermore, the first order reinitialization of the level set function (section 3.3) can effectively destroy the accuracy of the computed curvature, as was also mentioned by Sussman et al. [85]. This is verified in figure 3.6, where the curvature error after reinitialization does not converge in the sup-norm upon grid refinement.

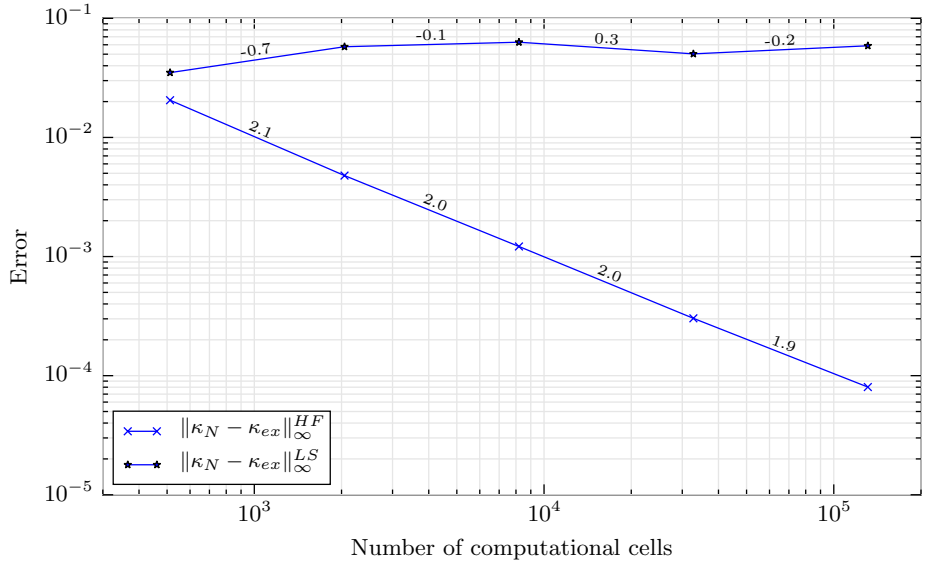


Figure 3.6: Convergence of the interface curvature with grid refinement of a stationary sphere of unit radius. The exact initial level set function is subsequently reinitialized to demonstrate the detrimental effect of the implemented reinitialization procedure on curvature accuracy: no convergence for the level set curvature κ^{LS}, whereas the Height Function curvature κ^{HF} shows second order accuracy upon refinement.

The reason for this is that first and second order differences are calculated from a quantity (the level set values) that itself is only second order accurate. These derivatives are second order only if the truncation error of the level set is sufficiently smooth, which is generally not the case in an arbitrary flow field. In du Chéné et al. [25] for example, it is shown that a fourth order reinitialization algorithm is generally required to maintain second order accuracy of the curvature calculations.

The Height Function (HF) method [12, 18, 40] uses the VoF values to locally reconstruct the interface. Originally developed for Cartesian domains, the implementation of the HF method in cylindrical coordinates is slightly more involved, but it relies on the same underlying principle. Depending on the orientation of the interface normal, a local interface height function h in a column of computational cells is sought that approximates the true interface through a mass balance in the column. If the orientation in the (r, z)-plane is such that the interface height only depends on the radius, i.e. $h = h(r)$, then the relation between the VoF values and the interface height in the i-th column is:

$$\int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} h(r) r \, \mathrm{d}r \sim \sum_j \psi_{i,j} r_i \Delta r_i \Delta z_j. \tag{3.61}$$

In the other case, when the height function only depends on the z-coordinate, i.e. $h = h(z)$, one finds for the k-th column:

$$\frac{1}{2} \int_{z_{k-\frac{1}{2}}}^{z_{k+\frac{1}{2}}} h(z)^2 \, \mathrm{d}z \sim \sum_j \psi_{j,k} r_j \Delta r_j \Delta z_k. \tag{3.62}$$

A schematic example of both cases is given in figures 3.7 and 3.8, respectively.

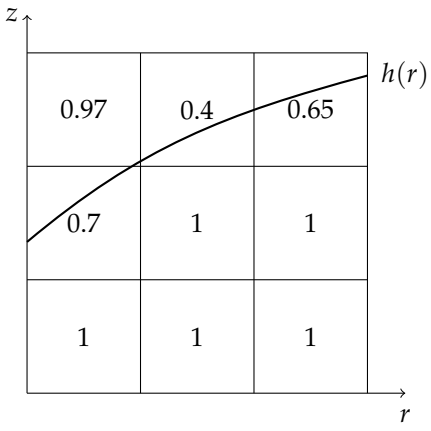


Figure 3.7: The case when $h = h(r)$.

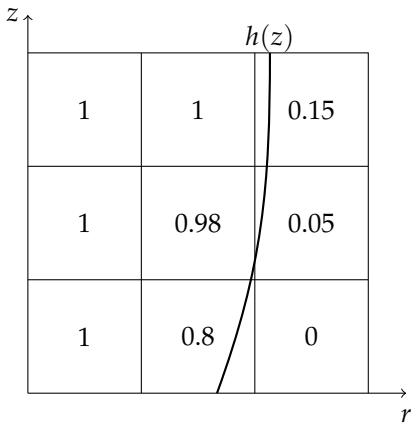


Figure 3.8: The case when $h = h(z)$.

The numbers in the cells represent the VoF values. For all of the three columns, the area under the polynomial curve $h(r)$ in figure 3.7 equals the cumulative amount of volume given by the VoF values. For $h(z)$ in figure 3.8, the same condition holds but then for all of the three rows. The extension to 3D domains requires a local

stencil of nine cells (3×3) to construct an exact quadratic interface.

For $h = h(r)$, the height function h is approximated by a quadratic polynomial, and for $h = h(z)$ a quadratic polynomial is used to approximate the function $h(z)^2$. A set of three columns then suffices for finding the polynomial coefficients, and the derivatives of the polynomial are subsequently used to obtain the respective curvatures. Near $r = 0$ a symmetry condition is used for the coefficients that requires only two columns. Even though this method also yields cell centred curvature values, by construction it generally leads to a significantly more uniform interface curvature field compared to the results based on the Level Set method, but at the cost of slightly higher computational effort. Some additional tuning can be done to improve the accuracy [56], and higher order curvatures are possible [82]. Furthermore, the method is second order accurate, which is verified in figure 3.6 by considering the curvature of a stationary sphere.

For flows with a significant amount of surface tension, we require for consistency that the magnitude of the parasitic currents vanishes when the grid is refined. In our implementation of the Ghost Fluid method, this implies that we need a curvature calculation that converges upon grid refinement. The Level Set method requires a high order reinitialization technique to guarantee convergence of the curvature. Although approaches exist to achieve this (like in du Chéné et al. [25]), the large stencils that are required may cause resolution problems when multiple interfaces approach each other and the level set function becomes poorly shaped. This could of course be remedied by locally switching to a lower order stencil, but clearly this approach is not optimal. Furthermore, the higher order reinitialization procedure becomes significantly more expensive, thereby increasing the overall computational complexity of the interface algorithm. For that reason, the primary curvature calculation in our cylindrical MCLS algorithm is based on the Height Function method. If an invalid solution is detected (for example due to a lack of resolution), the algorithm switches to a curvature calculation based on the level set for robustness.

3.3.6 Computational performance

Performance profiling for different grid sizes shows that the interface representation algorithm is very light (less than 5% of CPU time in a typical run) when compared to the cost of the flow solver. This can be expected when considering the complexity of the fully implicit flow solver versus the explicit interface advection. Although

likely to be slightly more efficient than standard VoF/PLIC because of the explicit interface present, our method is more expensive than compressive VoF methods. However, these algorithms in turn are often tied by a very stringent time step constraint to remain stable and conservative. Our method on the other hand is able to handle CFL numbers up to 0.5, which allows the use of significantly larger time steps. The efficiency of the method is further improved through the use of interface subcycling, where multiple interface advection steps can be performed within a single flow time step, thereby fully utilizing the unconditional stability of the flow solver in case of CFL numbers larger than 0.5.

3.4 Numerical validation of the algorithm

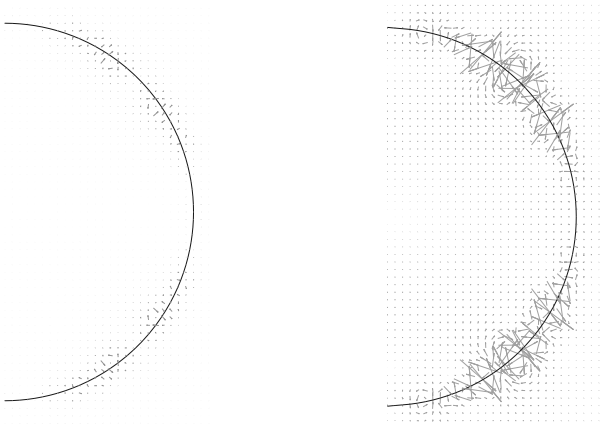
To validate the algorithm and establish its accuracy, four commonly used test cases are performed. These cases are chosen such that they can be performed on an axisymmetric domain.

3.4.1 Stationary bubble with high surface tension

As the first test case, we consider a spherical bubble inside a cylindrical domain. No gravity is present, and a large surface tension is balanced by a jump in pressure. It is well known that many numerical models that attempt to simulate similar cases are subject to parasitic or spurious velocities due to errors in the curvature calculation [39, 69, 72]. Figure 3.9 shows the scaled currents, which for a curvature calculation based on the Height Function method are usually maximal at the location where the inclination of the interface with respect to the coordinate directions is largest.
We will consider the test case of Denner and van Wachem [21], but instead of a closed Cartesian domain the case is performed in a closed cylinder. It is assumed that, for a sufficiently large domain, the results are comparable to the results in the literature that were done in a cube. The variables of interest are the maximum value of the spurious velocity $|\mathbf{u}|_{\max}$ around the interface, and the maximum relative errors in the pressure jump $E(\Delta p)$ and in the interface curvature $E(\kappa)$:

$$E(\Delta p) = \frac{\max|\Delta p - \Delta p_{exact}|}{\Delta p_{exact}}, \qquad E(\kappa) = \frac{\max|\kappa - \kappa_{exact}|}{\kappa_{exact}}. \tag{3.63}$$

Both Δp_{exact} and κ_{exact} are known from the Young-Laplace equation and the curvature of a sphere, respectively. The flow velocity, the pressure difference and the interface curvature are calculated after a single time step as well as after fifty time



(a) After 1 time step. (b) After 50 time steps.

Figure 3.9: Spurious velocities around the interface after 1 and 50 time steps.

steps to analyze the growth of the parasitic currents. The results are shown in table 3.1, together with the results of the VOF CELESTE method (a compressive VoF method that uses least-squares to approximate the curvature) and the VOF-PLIC HF method (a VoF method with PLIC interface reconstruction that uses the Height Function method for the curvature computation).

Table 3.1: Results of the stationary bubble test case after 1 and 50 time steps. The values of the VOF CELESTE and VOF-PLIC algorithms are from Denner and van Wachem [21].

After 1 step	$\lvert\mathbf{u}_{\max}\rvert$ (m/s)	E(Δp)	E(κ)
VOF CELESTE	$1.01 \cdot 10^{-3}$	$1.37 \cdot 10^{-2}$	$2.20 \cdot 10^{-2}$
VOF-PLIC HF	$7.92 \cdot 10^{-3}$	$3.69 \cdot 10^{-3}$	$4.35 \cdot 10^{-3}$
MCLS	$2.45 \cdot 10^{-3}$	$8.62 \cdot 10^{-3}$	$1.13 \cdot 10^{-2}$
After 50 steps	$\lvert\mathbf{u}_{\max}\rvert$ (m/s)	E(Δp)	E(κ)
VOF CELESTE	$6.75 \cdot 10^{-2}$	$1.26 \cdot 10^{-2}$	$1.85 \cdot 10^{-2}$
VOF-PLIC HF	$4.26 \cdot 10^{-2}$	$6.58 \cdot 10^{-3}$	$2.02 \cdot 10^{-2}$
MCLS	$9.33 \cdot 10^{-2}$	$1.03 \cdot 10^{-2}$	$1.09 \cdot 10^{-2}$

It is expected that, especially in the inviscid case, the pressure jump over the interface has second order accuracy upon grid refinement. To test this, the error $E(\Delta p)$ is calculated after 1 time step for several grids; as shown in figure 3.10, second order convergence is indeed achieved. To measure the growth of the parasitic currents, also the total kinetic energy after a single time step is shown in figure 3.10. The order of the kinetic energy seems to be much less regular than the order of the

pressure jump, but still good convergence is observed.

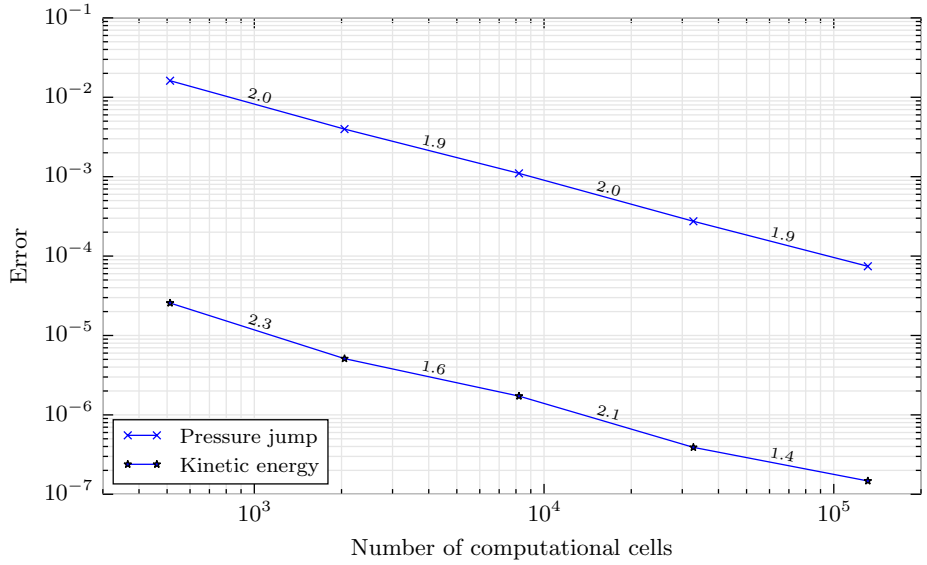


Figure 3.10: Error in the pressure jump and in the magnitude of the kinetic energy for the stationary bubble test case.

Table 3.1 shows that our algorithm produces results of the same order of magnitude when compared with the other two algorithms. Even though surface tension-dominated flows are not of our primary interest, this test case shows that the current implementation of the Height Function method for the curvature together with the Ghost Fluid method for the treatment of the surface tension is quite adequate.

3.4.2 Rising spherical gas bubble in a stagnant liquid column

The second test case is the rising bubble in a column filled with a viscous liquid. Starting at rest, the bubble rises due to boyancy, and eventually settles, usually with a terminal velocity and shape, but not always (in the case of wobbling bubbles). For this test case, we replicate cases A and B of the spherical cap bubble experiments from Hnat and Buckmaster [42]. The only difference between these cases is a larger initial bubble diameter in case B than in case A. The parameters of interest are the Reynolds number Re and the Morton number Mo, defined as:

$$Re = \frac{\rho_l U_t r}{\mu_l}, \qquad Mo = \frac{g\mu_l^4}{\rho_l \sigma^3}, \tag{3.64}$$

where ρ_l and μ_l are the density and dynamic viscosity of the liquid, U_t is the

terminal rise velocity, r the initial bubble radius, g the gravitational acceleration and σ the surface tension coefficient. Table 3.2 shows the parameters used for the simulation of both cases.

Table 3.2: Parameters of the rising bubble simulations ($Mo = 0.065$).

Parameter	Case A	Case B
cylinder radius R [m]	$4.2 \cdot 10^{-2}$	$6 \cdot 10^{-2}$
cylinder length L [m]	$1.68 \cdot 10^{-1}$	$2.4 \cdot 10^{-1}$
liquid density ρ_l [kg/m^3]	875.5	875.5
gas density ρ_g [kg/m^3]	1.225	1.225
liquid viscosity μ_l [kg/m/s]	0.118	0.118
gas viscosity μ_g [kg/m/s]	$1.77 \cdot 10^{-5}$	$1.77 \cdot 10^{-5}$
initial bubble radius r [m]	$6.08 \cdot 10^{-3}$	$8.3 \cdot 10^{-3}$
initial bubble height h [m]	$3.36 \cdot 10^{-2}$	$4.8 \cdot 10^{-2}$

The time step Δt was fixed at 10^{-4} s, and the simulations were performed until $t = 0.45$ s on a grid of 128×512 cells in the radial and axial direction, respectively. The value of the surface tension σ is set at 0.0322 kg/s^2, and gravity acts in the negative z-direction with magnitude 9.8 m/s^2. The initial state of the simulation is shown in figure 3.11.

The dimensions of the cylindrical domain are chosen according to Mukundakrishnan et al. [62], to make sure that wall effects are not significant. We are interested in the terminal rise velocity U_t expressed as a terminal Reynolds number Re_t. Note that Mo is an input parameter, whereas Re is an output parameter that results from the simulation. The rise velocity $U^{(n)}$ at time $t^{(n)}$ is defined as the velocity of the bubble centroid using the VoF values:

$$U^{(n)} = \frac{\sum_{i,k} \frac{1}{2} \left(u^{(n)}_{z_{i,k-1}} + u^{(n)}_{z_{i,k}} \right) \psi^{(n)}_{i,k} \Omega_i}{\sum_{i,k} \psi^{(n)}_{i,k} \Omega_i}, \tag{3.65}$$

where $\Omega_i = r_i \Delta r_i \Delta z$ is the volume of cell (i,k). Table 3.3 shows the terminal Reynolds number, as obtained with the respective experimental results for both cases, and also as obtained with our MCLS algorithm. For a better comparison, the two cases are also performed with two commercial CFD packages (Fluent v15.0 and Star-CCM+ v10.02) and with the open-source software OpenFOAM v2.3. These additional results are also included in table 3.3. The four simulation tools used about the same number of spatial grid cells. In Fluent, the convective terms of the momen-

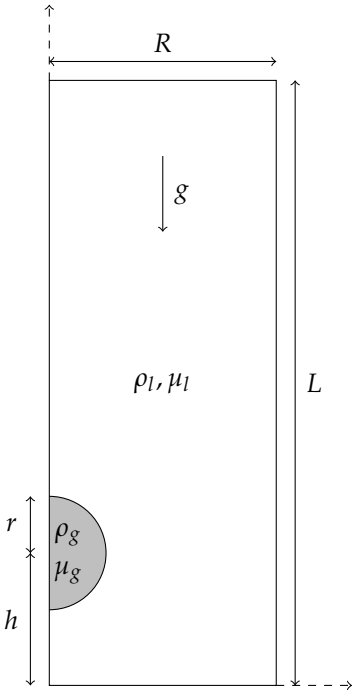


Figure 3.11: Geometry and initial state of the rising bubble simulations.

tum equation are discretized by using a second order upwind method with a limiter, and the PRESTO algorithm is used in the pressure calculation. A first order time integration is used. The interface representation is based on the VoF method with a geo-reconstructed (PLIC) interface. For Star-CCM+, the Navier Stokes equations are discretized by using a third order MUSCL scheme for the convective terms and a first order (Euler Backwards) time integration. A second order High-Resolution Interface Capturing (HRIC) scheme with a sharpening factor of 0.1 is used for the advection of the volume fraction. Finally, for OpenFOAM, the interFoam solver is used with a second order upwind scheme (with a limiter) for the discretization of the convective operator terms in the momentum equation and the explicit version of the MULES scheme in the phase fraction equation. The CFL number is set to a constant value of 0.05 to achieve sufficient interface compression.

Table 3.3: Results for the terminal Reynolds number in the rising bubble test case ($Mo = 0.065$).

Case	Re_{MCLS}	$Re_{OpenFOAM}$	$Re_{Star-CCM+}$	Re_{Fluent}	Re_{exp}
A	9.4	9.3	9.1	9.3	9.8
B	15.3	14.7	15.1	15.3	17.1

Table 3.3 shows that the four simulation models give very close results for the terminal Reynolds number in cases A and B, albeit slightly lower than the experiments. All four results are somewhat lower than the experimental values; the difference between the simulations and experiments was not further analyzed as our main purpose is the comparison of numerical methods for the given mathematical model. A possible reason is the presence of wall effects in the experiment, which are not included in the simulations. Figure 3.12 shows that qualitatively, our algorithm is well capable of simulating the shape of the bubble as soon as it is stabilized.

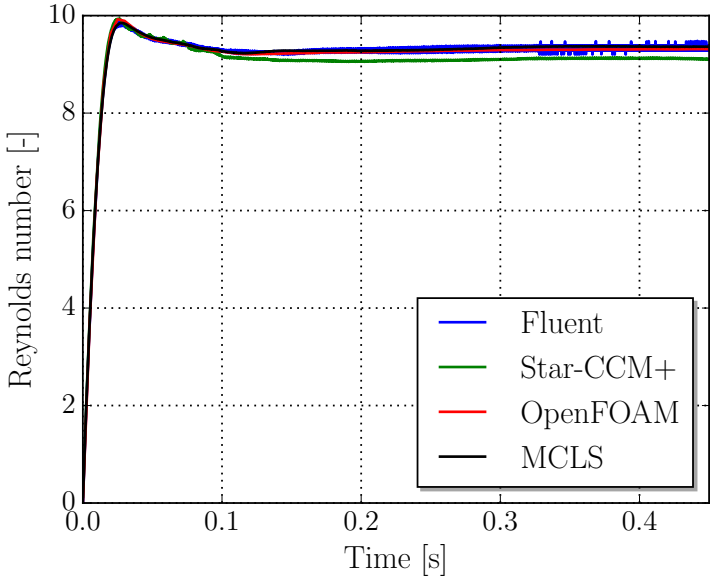
(a) Case A (b) Case B

Figure 3.12: Steady state configuration of a rising air bubble in viscous liquid for cases A and B: experiment (left side of graphs) versus simulation (right side of graphs). (Experiment photo copied from Hnat and Buckmaster [42])

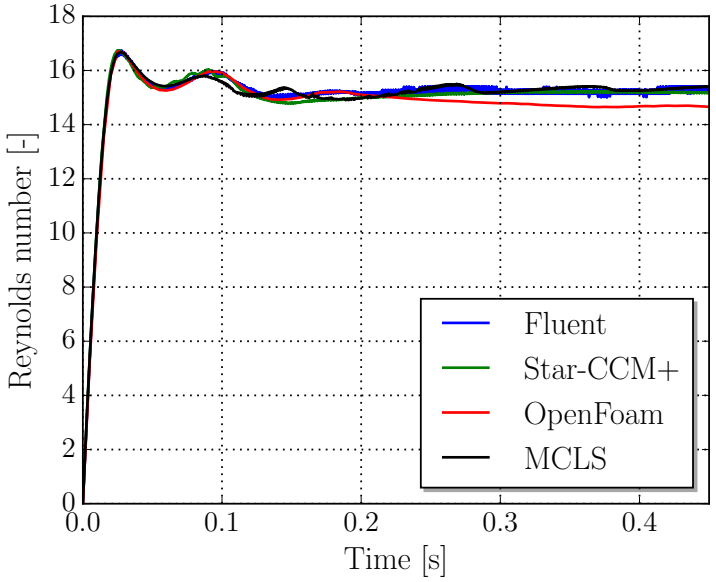
Figure 3.13 shows the terminal Reynolds number over time for both cases A and B. For a qualitative comparison between the four different simulation algorithms, figures 3.14 and 3.15 show the contour of the bubbles at times $0, 0.05, 0.1, 0.25$ and 0.45 seconds.

Case A involves a relatively stable bubble, as can be seen in figure 3.13a. The bubble accelerates and without much wobbling approaches its terminal shape and velocity. For case B, however, the bubble is less stable, and figure 3.13b shows that after the initial acceleration some wobbling occurs that slowly damps out. Looking at figure 3.15, the skirt of the bubble in case B is much sharper than in case A, and this may cause large errors in the curvature calculations and in the subsequent surface tension approximations.

The MCLS method maintains a sharp edge, but when the bubble rises, the edge slowly droops and is subsequently 'pulled' back up again to form a relatively flat edge, which is repeated over time (this oscillation can be seen in figure 3.13b). The drooping of the bubble edge is most clearly visible in the Fluent result, while OpenFOAM and Star-CCM+ do not seem to suffer from this. However, a visualization of the VoF field around the bubble in figure 3.16 shows that in fact these latter two

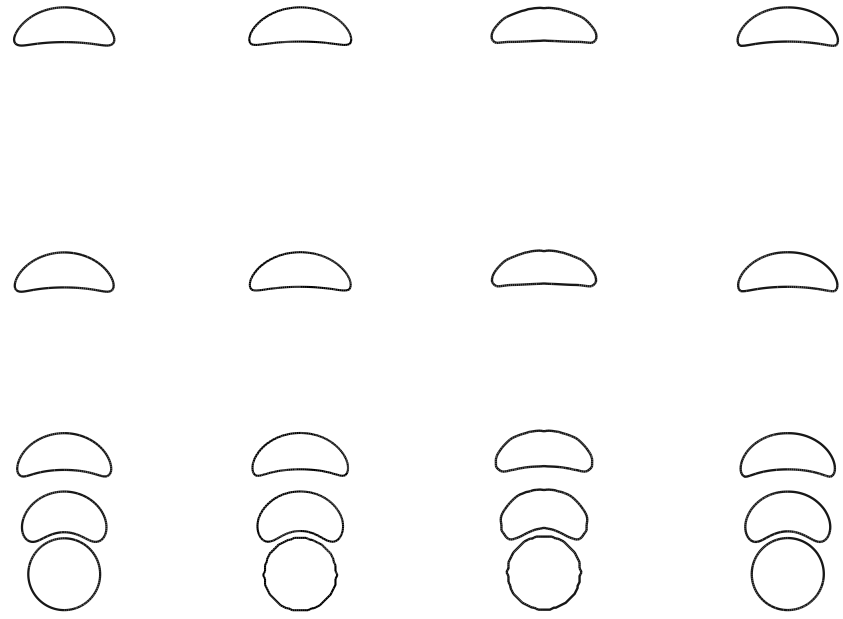


(a) Case A: experimental Reynolds number is 9.8.



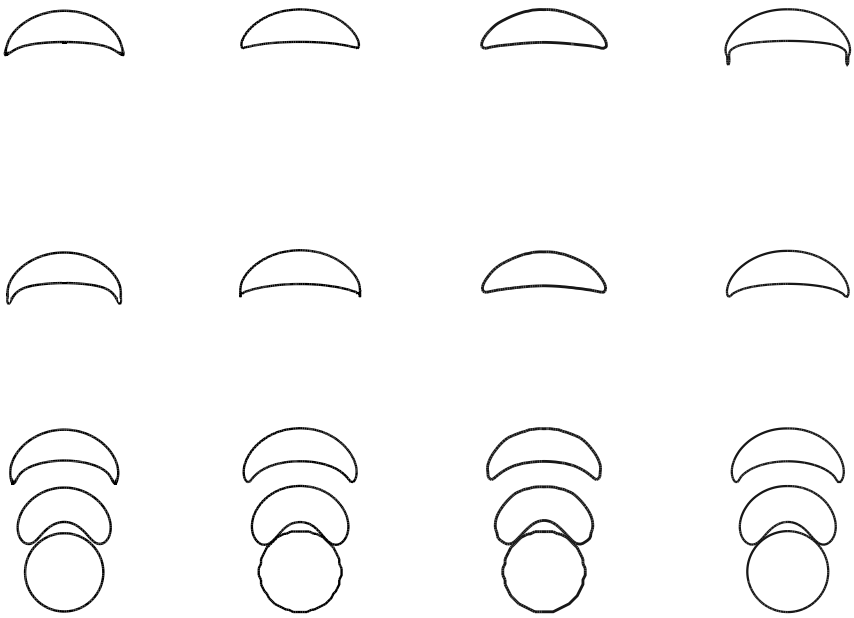
(b) Case B: experimental Reynolds number is 17.1.

Figure 3.13: Rise velocities expressed as Reynolds numbers for both cases A and B.



(a) MCLS (b) OpenFOAM (c) Star-CCM+ (d) Fluent

Figure 3.14: Snapshots of the rising bubble case A at times $t = 0, 0.05, 0.1, 0.25$ and $t = 0.45$ s.



(a) MCLS (b) OpenFOAM (c) Star-CCM+ (d) Fluent

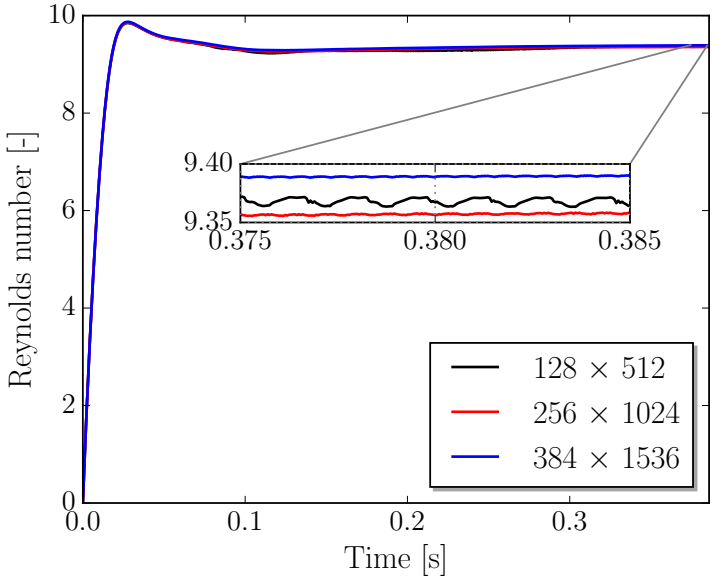
Figure 3.15: Snapshots of the rising bubble case B at times $t = 0, 0.05, 0.1, 0.25$ and $t = 0.45$ s.

packages suffer from volume leakage from the bubble. Subsequent investigation has shown that in turn the MCLS method and Fluent do not show any signs of leakage during the test case. From figure 3.16 it is clear that the volume loss of the bubble in OpenFOAM is much more significant than in Star-CCM+. Since the method used to calculate the rising velocity (expression 3.65) does not distinguish between contributions inside or outside the bubble (it is therefore only useful for bubbles without break-up), the relatively high volume leakage may contribute to the lower terminal velocity of OpenFOAM in figure 3.13b.

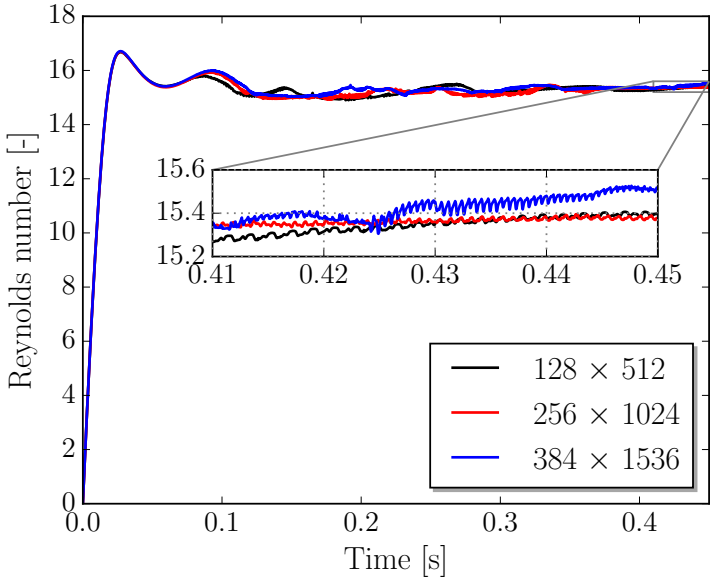
(a) OpenFOAM (b) Star-CCM+

Figure 3.16: VoF contours during the rising bubble test case. Both OpenFOAM and Star-CCM+ show a degree of leakage that originates from the corner of the flattened bubble.

A refinement study has also been performed for both cases A and B using the original grid with 128×512 cells, and two refined grids with 256×1024 cells and 384×1536 cells, respectively. The time step was decreased accordingly to keep the Courant number constant. The results are shown in figure 3.17 for both cases. The oscillatory behaviour of the rise velocity is magnified for both case A and B, and it is clear that in case A both the period and the amplitude diminish upon refinement. It can be seen in figures 3.13a and 3.13b that the other methods also suffer from these oscillations (Fluent in particular). For case B, however, there appears an initial decrease in fluctuation, but at the finest grid the oscillations are growing again. The exact nature of the velocity fluctuation needs to be investigated more thoroughly, but they are likely to stem from an imbalance between pressure and surface tension forces. Furthermore, it can be seen that the convergence is not monotonic in both of the two cases, and a truly steady state has not yet been reached on the current grids. Further refinement studies therefore need to be performed to study the convergence of the rise velocities.



(a) Grid refinement study for case A.



(b) Grid refinement for case B.

Figure 3.17: Successive grid refinement for both cases A and B using our interface algorithm.

3.4.3 Taylor bubble

The third test case comprises the rise of a long gas bubble through a vertical cylindrical pipe, also known as a Taylor bubble. In practice this configuration is often found in wells and vertical risers used for hydrocarbon recovery from reservoirs and transport to offshore platforms. It serves as an important test case for the simulation of slug flow. As for the rising bubble case, the main interest lies in the rise velocity of the bubble [2, 50, 86], as well as in the thickness of the falling liquid film between the bubble and the pipe wall [55] and the wake of the bubble [15, 50]. We will try to reproduce some of the numerical results of Lu and Prosperetti [57], which in turn were compared with the experiments of White and Beardmore [95] and Bugg and Saad [15]. The commonly used dimensionless parameters for this test case are the Eötvös number, the Morton number and the Froude number, defined as:

$$Eo = \frac{\rho_l g D^2}{\sigma}, \qquad Mo = \frac{g\mu_l^4}{\rho_l \sigma^3}, \qquad Fr = \frac{U_t}{\sqrt{gD}}, \tag{3.66}$$

where ρ_l and μ_l are the fluid density and viscosity, respectively, g the gravitational acceleration, $D = 2R$ the cylinder diameter, σ the surface tension coefficient and U_t the terminal rise velocity of the bubble defined as in equation 3.65. Note that Eo and Mo are input parameters, whereas Fr is an output quantity that results from the simulation. A total of five cases are considered, with two low-viscosity cases (A and B) and three moderately viscous cases (C, D and E). The parameters of each case are shown in table 3.4.

Table 3.4: Parameters for the different test cases of the Taylor bubble.

Case	Eo	Mo	Fr (exp)	Fr (MCLS) 35×350	Fr (MCLS) 70×700
A	15	$1.8 \cdot 10^{-8}$	0.23	0.233	0.234
B	60	$1.8 \cdot 10^{-8}$	0.33	–	–
C	18.7	$1.6 \cdot 10^{-2}$	0.10	0.0975	0.0984
D	74.6	$1.6 \cdot 10^{-2}$	0.27	0.273	0.275
E	100	$1.5 \cdot 10^{-2}$	0.30	0.297	0.297

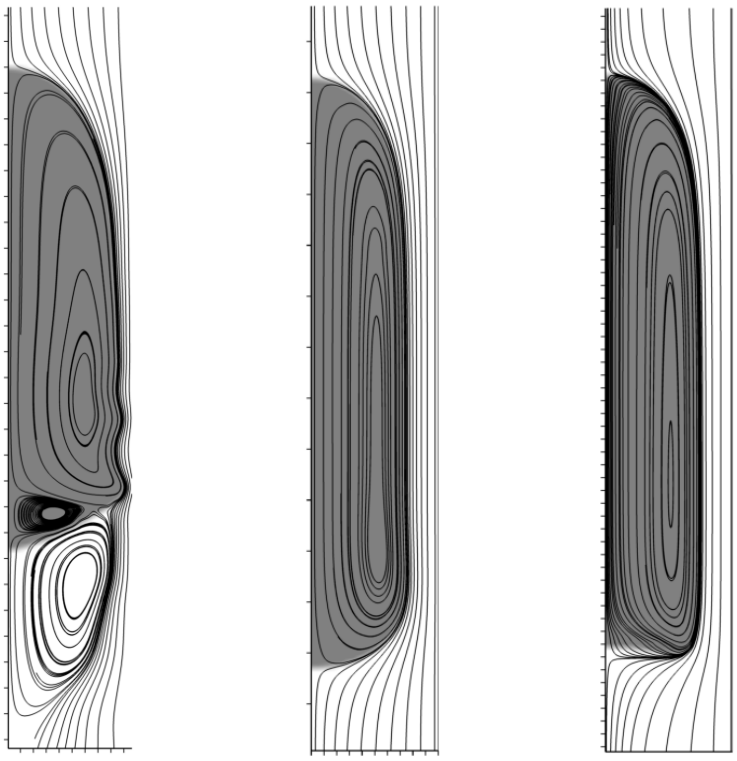
The values in the column headed by 'Fr (exp)' are the numerical values found by Lu and Prosperetti [57], which are claimed to be within 1% of the experimentally measured values of White and Beardmore [95]. The column headed by 'Fr (MCLS)' are the numerical values found with our algorithm. A non-uniform grid is employed,

with a radial node distribution according to:

$$r_i = \frac{\tanh(\alpha R i / N_r)}{\tanh(\alpha R)}, \quad i = 1, \ldots, N_r, \tag{3.67}$$

with R the cylinder radius and $\alpha = \alpha(R)$ a scaling constant. The grid is refined near the cylinder wall at $r = R$ to better capture the thin film of liquid between the wall and the rising bubble. For all cases, the number of cells in radial direction N_r was set at 35 to be able to compare with Lu and Prosperetti [57], who used 35×420 cells. To test the efficiency of the non-uniform grid, we performed additional simulations of case A for a radial distribution equal to equation 3.67, but with only 20 cells and a higher level of non-uniformity due to a larger value of α. This gave a resulting Froude number of 0.228, compared to the 35-cell value of 0.232. A refinement case was done as well, where 70 cells were used in radial direction and the Courant numbers were kept constant. The results are shown in the last column of table 3.4.

The tail of the bubble in case A is oscillating and therefore no true steady state configuration is reached, although the rise velocity has stabilized. This corresponds very well with the results from Lu and Prosperetti [57], where a similar oscillatory behaviour is found for case A.
For case B, the low viscosity in combination with the relatively small surface tension cause the bubble to break up after some time. The initially convex rear end of the bubble rapidly becomes concave in the absence of sufficient surface tension, and subsequently starts to protrude the bubble from below until it breaks up completely. Consequently, no valid value of the terminal rise velocity could be derived. This break up does not seem to appear in the simulations by Lu and Prosperetti [57].
Cases C, D and E have nearly the same fluid viscosity but there is a decreasing influence of the surface tension as indicated by the increasing values of the Eötvös number. For high surface tension, the initial bubble tends to become spherical as this is the optimum shape from an energetic point of view. An Eötvös number below about 4 will completely prevent the bubble from moving [95]. Consequently, the terminal rise velocity increases with decreasing Eötvös number, which is also seen in the simulation results.
The difference in liquid viscosity yields two distinctive flow patterns below the rear end of the bubble. In cases A and B, large vortices with reverse flow are present, while for cases C, D and E, the higher viscosity prevents any recirculation from



(a) Case A (b) Case C (c) Case D

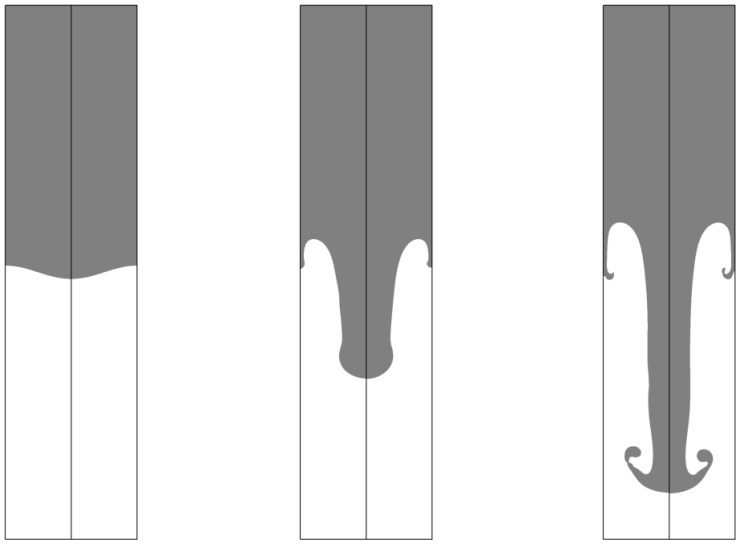
Figure 3.18: Flow patterns of the Taylor bubbles of cases A (low viscosity), and C and D (moderate viscosity).

forming and the falling liquid is smoothly passing the bubble. This is shown in figure 3.18 for the cases A, C and D.

3.4.4 Taylor-Rayleigh instability

To test the ability of our algorithm to capture instabilities, a Taylor-Rayleigh instability, initiated by a heavier fluid on top of a lighter fluid in a closed vertical cylinder, is considered as a fourth test case. The fluid parameters are taken from Gopala and van Wachem [35], with fluid densities $\rho_h = 1.225$ kg/m^3 and $\rho_l = 0.1694$ kg/m^3, and both viscosities μ_h and μ_l equal to $3.13 \cdot 10^{-3}$ kg/m/s. Here the subscript h refers to the heavier fluid and the subscript l refers to the lighter fluid. The cylindrical domain has a radius $R = 0.5$ m and length $L = 4$ m, and the initial interface height is given by $z(r) = 0.05\cos(2\pi r)$. A no-slip boundary condition is imposed on all the walls of the cylinder. The simulation is performed on two grids of 32×256 and 64×512 cells with constant time steps $2 \cdot 10^{-4}$ and $1 \cdot 10^{-4}$ seconds, respec-

tively, and without surface tension. The results from both grids are shown in figure 3.19. Both simulations are performed with an allowable mass loss of 10^{-10}.



(a) At $t = 0$ seconds. (b) At $t = 0.6$ seconds. (c) At $t = 0.9$ seconds.

Figure 3.19: A Taylor-Rayleigh instability at various time instances. The left halves are the results from the 32×256 grid, the right halves are the results from the 64×512 grid.

In the case of incompressible inviscid fluids, it can be shown that small perturbations η of an initially axisymmetric interface at $z = 0$ in a long cylinder can be described by:

$$\eta(r,t) = h_0 J_0\left(\frac{kr}{R}\right)\cosh(st), \tag{3.68}$$

where h_0 is some constant and J_0 is the Bessel function of the first kind of order zero [20]. The growth rate s for small initial perturbations of the interface can be determined through:

$$s^2 = \frac{g(\rho_h - \rho_l)k - \sigma k^3}{\rho_l + \rho_h}, \tag{3.69}$$

where g is the gravitational acceleration, σ is the surface tension coefficient and k satisfies:

$$J_0'(kR) = 0, \tag{3.70}$$

in which R is the cylinder radius. This can be rewritten as:

$$s^2 = kgA\left(1 - \Phi\right), \tag{3.71}$$

where A is the Atwood number, defined as:

$$A = \frac{\rho_h - \rho_l}{\rho_l + \rho_h}, \tag{3.72}$$

and Φ is a stability parameter:

$$\Phi = \frac{\sigma k^2}{g(\rho_h - \rho_l)}. \tag{3.73}$$

The inherently unstable configuration of a heavier fluid on top of a lighter fluid can thus be stabilized by a surface tension force between the two fluids. The value Φ determines whether any initial perturbation is accelerated due to gravity ($\Phi < 1$), or whether is it is neutralized due to surface tension ($\Phi > 1$). The ability of our algorithm to follow the linearized theory is tested by determining the growth rate of the interface height for a set of values for Φ. The initial perturbation of the interface is given by $z(r) = h_0 J_0(kr)$ with $h_0 = 5 \cdot 10^{-3}$ m. The cylindrical domain has radius $R = 1$ m and length $L = 6$ m, and the fluid densities are $\rho_h = 1$ kg/m^3 and $\rho_l = 0.25$ kg/m^3 for the heavier and lighter fluid, respectively, while both viscosities are zero. The value of k follows from $J_0'(kR) = 0$, which gives $k = 3.8317$ m^{-1}. The acceleration due to gravity is taken as $g = 1$ m/s^2. As the fluids are considered to be inviscid, free-slip boundary conditions are used on all cylinder walls. The computation is done on two grids with 30×180 and 60×360 computational cells in radial and axial direction, respectively. The interface at a fixed radial location is tracked during a time span of 3 seconds, after which the growth rate s is determined while assuming that the perturbation develops according to expression 3.68. Figure 3.20 shows the results for various values for Φ.

The relative errors (i.e. the scaled difference between the numerical prediction and the analytical solution) for the considered values of Φ are between $0.3 - 1.1\%$. The deviation is largest for vanishing surface tension. Because the measurements are done at a fixed time, the interface has moved much more than it would have done with larger surface tension, and therefore it may have left the regime in which the analytical assumption holds, leading to larger errors. Overall, however, there is a very satisfactory agreement between the numerical predictions and the analytical results.

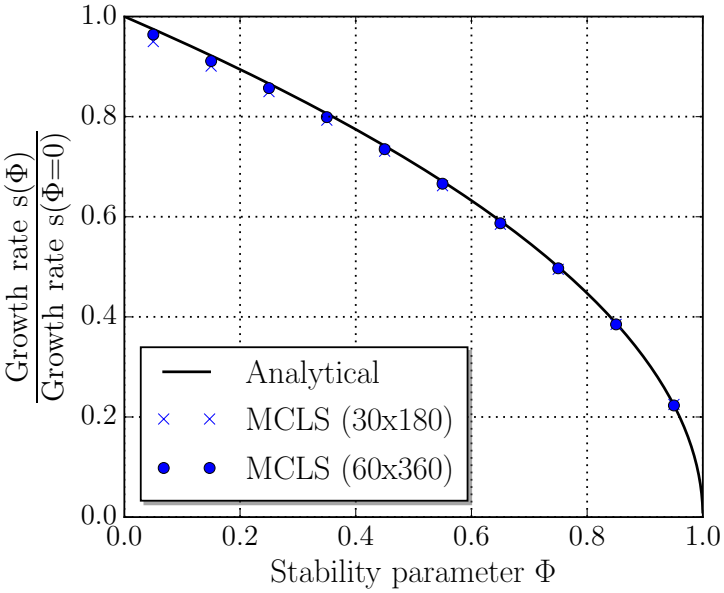


Figure 3.20: Scaled growth factor s for different surface tension values: MCLS results versus the analytical result of expression 3.71.

3.5 Concluding remarks

This chapter described the development of a dual interface capturing method for the simulation of incompressible immiscible two-phase pipe flows based on the interface model of the MCLS framework. The focus lies on computational efficiency (having the simulation of turbulent flows in mind) at the cost of a reduced flexibility in choosing the grid. The cylindrical Navier-Stokes equations are discretized using second order finite differences on a staggered cylindrical grid, and the flow is evolved in time using a second order implicit midpoint scheme in combination with a standard projection scheme to decouple the pressure and the velocity. The interface is modelled by a coupled level set/VoF method that is optimized for efficiency. An interface advection algorithm that is inherently conservative is introduced, and the VoF-based Height Function method is extended to cylindrical geometries to obtain converging curvature computations upon grid refinement. The Ghost Fluid method is used to model the interface jump conditions.

To validate our algorithm, four well known test cases for multiphase flows are considered. The surface tension model (Ghost Fluid with Height Function curvature)

is tested and gives comparable results with the literature in terms of the pressure jump and spurious velocities. The rising bubble test case shows that our method consistently follows the majority of the results obtained with three other CFD packages, even though all four give a slightly lower terminal bubble velocity than the experimental result. This is clearly an improvement on the performance of the original MCLS algorithm on a Cartesian grid as considered by Denner et al. [22], where the rising bubble was not well captured. The simulation of a Taylor bubble shows very good results when compared to other numerical results in the literature and to experiments, both for the shape and for the terminal velocity. Finally, it is shown that our algorithm is well capable of capturing a Rayleigh-Taylor instability as demonstrated through a comparison with the associated analytical theory.

Acknowledgments

We would like to sincerely thank dr. M. Pourquie and dr. J. Kreeft for their effort in obtaining the numerical results for the rising bubble test case of section 3.4.2 in both Fluent and Star-CCM+, respectively.

Chapter 4

Numerical prediction of two-phase flow instabilities in cylindrical pipes

In this chapter we study the application of a combined Level Set/Volume of Fluid method to simulate two-phase Kelvin-Helmholtz instabilities in cylindrical geometries. The choice of a structured cylindrical grid allows for a fast and efficient algorithm. The simulation results of the algorithm are compared with linear modal theory in the case of a 2D core-annular flow, giving a very good agreement for the growth factors. The 3D capabilities of the method are tested through simulations of Kelvin-Helmholtz waves in inclined pipe sections under similar conditions as used in the well known experiments of Thorpe [88].

4.1 Introduction

The numerical modeling of multiphase flow phenomena has been a topic of much research over the past few decades. Examples of multiphase flows in industry are ubiquitous: boilers, condensers, reactors, mixers, separators and (cavitating) propellers are among some of the applications. In the oil and gas industry the effort for hydrocarbon recovery from more remote and deeper production fields has led to more complex pipeline transport. It is quite common to find that both

The content of this chapter is based on the article:

G.T. Oud, D.R. van der Heul, C. Vuik, and R.A.W.M. Henkes. *Numerical prediction of two-phase flow instabilities in cylindrical pipes*, to be submitted.

phases are transported through a single pipeline to an offshore platform or to an onshore plant. The liquid-gas mixture flowing through a pipeline can be in a specific flow regime, such as stratified flow or slug flow, depending on, among others, the liquid and gas flow rates and the pipe diameter. Unstable flows with slugs cause significant mechanical stresses on the pipe wall and its supports, and can also cause difficulties for the facilities at the end of the pipeline, such as flooding of the separator or trips of the compressor. It is therefore desirable to be able to predict what kind of conditions lead to slug flow, in both horizontal and inclined pipes.

4.1.1 3D cylindrical coordinates for accuracy and efficiency

To get a better understanding of the physical mechanisms underlying this transition process to slug flow, a highly accurate and strictly mass conserving computational model has been developed specifically for the simulation of two-phase flow in a straight segment of a cylindrical pipe with a circular cross section. A key element is the choice to use 3D cylindrical coordinates in the discretization process. Although this choice leads to certain numerical challenges near the cylinder origin, discretizations on structured orthogonal grids generally show at least one order of improved performance in both accuracy and computational efficiency compared to unstructured grids, which are currently often used in cylindrical geometries. As the field of CFD for turbulent multiphase flows is currently emerging, we believe that this substantial gain in performance justifies an investigation into the use of cylindrical coordinates in two-phase flow modelling. By limiting the applicability of our algorithm to a straight pipe section, we can subsequently optimize the accuracy which enables us to make very high resolution (eventually DNS and LES) simulations using moderate computational resources.

The flow is considered to be incompressible and isothermal, and the two immiscible phases are separated by a sharp interface. It can therefore be accurately described by a two-phase flow model consisting of the variable density/viscosity Navier-Stokes equations and a separate model for the evolution of the interface. Controlling numerical dissipation is very important in this application to be able to distinguish the actual physical waves from the spurious numerical amplification of interface disturbances. Furthermore, mass should be accurately conserved, as loss of (moving) mass generally implies loss of momentum (i.e. artificial damping). We believe that an optimal accuracy and (parallel) efficiency can be achieved by using a sim-

ple second order finite difference discretization for the Navier-Stokes equations in cylindrical coordinates on an orthogonal structured grid, together with a dedicated mass conserving, finite volume based interface model that exploits all symmetries and regularity of the control volumes on that same grid.

4.1.2 Dual interface capturing models

Nearly all interface models that do not impose any restrictions on changes in the topology of the interface are based on either the Level Set (LS) method [66] or the Volume of Fluid (VoF) method [41]. Individually, however, each suffers from drawbacks: the LS method is inherently incapable of conserving mass over time, while the VoF method, which lacks an explicit interface, often requires complicated and computationally expensive interface reconstructions. Although the impact of these drawbacks can be reduced to a certain extent, yet another approach combines both methods with the aim of benefiting from their advantages (the explicit interface of the LS method and the conservation property of the VoF method) while eliminating their disadvantages.

As the interface is described by congruent LS and VoF fields, we refer to these methods as *hybrid-* or *dual* interface capturing methods. Because no such algorithms in 3D cylindrical coordinates appear to exist in the literature, we have devised a dedicated cylindrical dual interface method for our purposes. Because of the similarities between generic dual methods, we provide only a brief explanation of our algorithm in section 4.2.2. More details are given on three sub-algorithms specifically designed for cylindrical coordinates: (1) an efficient function relating the LS and VoF values, (2) a conservative interface advection scheme and (3) an algorithm to improve the accuracy of the VoF advection near the origin.

4.1.3 Stratified multiphase pipe flow

To test the full 3D capabilities of our interface algorithm, a test case based on the well-known stratified flow experiments of Thorpe [88] is performed. These experiments are well established as the basis for a number of 2D computational multiphase studies, such as by Atmakidis and Kenig [3], Bartosiewicz et al. [9], Fleau et al. [31], Štrubelj and Tiselj [80]. The channel configuration allows simulations to be done in 2D rectangular domains due to the application of Squire's theorem to stratified layers of fluids [7, 97]. Furthermore, classical (inviscid) Kelvin-Helmholtz

theory in a confined domain can be applied to make a priori estimates of the magnitude of the expected wave lengths, of the onset time of the perturbation and of the velocity profiles. For 3D domains with a circular cross section, however, theory similar as for the 2D channel appears to be rather limited. The stability analysis for stratified flows in pipes is generally based on modal stability theory of the linearized one-dimensional two-fluid model, as studied by, among others, Barnea and Taitel [8], Guo et al. [37], Salhi et al. [75], Taitel and Dukler [87]. These closure correlations cover different physical flow features, such as the wall friction and the interfacial stress between the phases [79, 89]. The accuracy of the outcome of the stability analysis is ultimately limited by the 1D character of the two-fluid model (and the applied closure relations), while it is plausible that the curved walls may introduce non-negligible 2D effects that are consequently not captured. However, no literature has been found that numerically investigates the accuracy of the two-fluid model-based stability theory in cylindrical pipes using a 3D interface representation.

4.1.4 Outline

The objective of this chapter is to analyze to what extent our cylindrical algorithm is capable of simulating interface instabilities in two-phase pipe flows. In particular, we focus on the formation of Kelvin-Helmholtz instabilities that arise due to small perturbations of the interface in shear flows. We apply our algorithm to two test cases that resemble common physical phenomena: shear flow in a core-annular flow, and stratified flow in an inclined pipe section. Due to the shape of the interface, the first case, which is in 2D, allows a comparison with well-established stability theory. For any interface that does not lie along a coordinate direction, however, the supporting theory is lacking. Therefore, the second test case, which is in 3D, is of a more qualitative nature.

Section 4.2 gives a description of the numerical algorithm used. The discretization of the Navier-Stokes equations in space and time is explained for a cylindrical geometry, and some information on the parallelization is given. Section 4.2.2 describes the applied dual interface method. The procedure of the efficient coupling between the LS method and the VoF method is explained, and a conservative interface advection algorithm is presented, together with a proposed method to improve the VoF accuracy near the cylinder origin. The first test case in section 4.3 is based on modal stability analysis of an axisymmetric interface in a core-annular flow. The re-

sulting growth factor that stems from the linear stability analysis is compared to the result of our numerical algorithm for various conditions. In section 4.4, a stratified shear flow in a periodic domain is considered as a preparation for the simulation of a closed pipe. Finally, in section 4.5 we study stratified flow in an inclined closed pipe segment, similar to the conditions in the experiments of Thorpe [88]. The shear pattern at the interface in the stratified flow configuration gives Kelvin-Helmholtz waves in a certain part of the pipe. We conclude with a summary of the results and some concluding remarks in section 4.6.

4.2 Numerical approach

In this section, the numerical algorithm used for the investigation is discussed. The approximation of the flow field (4.2.1) and the cylindrical interface algorithm (4.2.2) are described separately.

4.2.1 Calculation of the flow field

The flow field in the straight cylindrical pipe section is calculated by using a conservative finite difference approximation of the governing flow equations. We consider the three-dimensional cylindrical domain:

$$\left\{ (r,\theta,z) \;\middle|\; 0 \le r \le R, 0 \le \theta < 2\pi, -\frac{L}{2} \le z \le \frac{L}{2} \right\}, \tag{4.1}$$

on which the Navier-Stokes equations in cylindrical coordinates are solved for the unknown velocity vector $\mathbf{u} = (u_r, u_\theta, u_z)$ and for the pressure scalar field p. The flow equations in conservative form are given by:

$$\frac{1}{r}\frac{\partial (r u_r)}{\partial r} + \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} + \frac{\partial u_z}{\partial z} = 0, \tag{4.2}$$

and:

$$\begin{aligned} \frac{\partial u_r}{\partial t} + \frac{1}{r}\frac{\partial (r u_r^2)}{\partial r} + \frac{1}{r}\frac{\partial (u_r u_\theta)}{\partial \theta} + \frac{\partial (u_r u_z)}{\partial z} - \frac{u_\theta^2}{r} &= -\frac{1}{\rho}\frac{\partial p}{\partial r} \\ &+ \frac{1}{\rho r}\frac{\partial (r\tau_{rr})}{\partial r} + \frac{1}{\rho r}\frac{\partial \tau_{r\theta}}{\partial \theta} + \frac{1}{\rho}\frac{\partial \tau_{rz}}{\partial z} - \frac{\tau_{\theta\theta}}{\rho r} + g_r, \end{aligned} \tag{4.3}$$

$$\frac{\partial u_\theta}{\partial t} + \frac{1}{r}\frac{\partial (r u_r u_\theta)}{\partial r} + \frac{1}{r}\frac{\partial (u_\theta^2)}{\partial \theta} + \frac{\partial (u_\theta u_z)}{\partial z} + \frac{u_r u_\theta}{r} = -\frac{1}{\rho r}\frac{\partial p}{\partial \theta} + \frac{1}{\rho}\frac{\partial (r\tau_{r\theta})}{r\partial r} + \frac{1}{\rho r}\frac{\partial \tau_{\theta\theta}}{\partial \theta} + \frac{1}{\rho}\frac{\partial \tau_{\theta z}}{\partial z} + \frac{\tau_{r\theta}}{\rho r} + g_\theta, \tag{4.4}$$

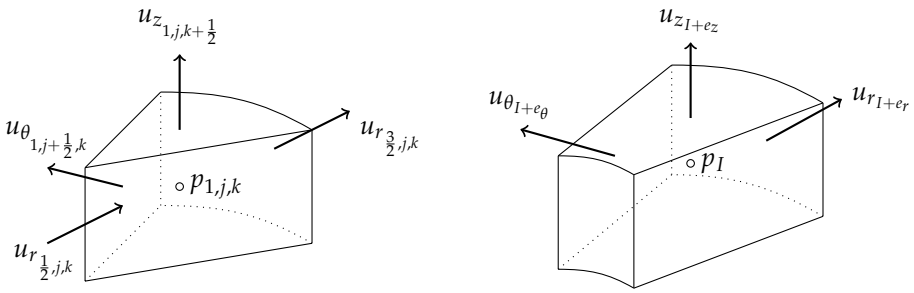
$$\frac{\partial u_z}{\partial t} + \frac{1}{r}\frac{\partial (r u_r u_z)}{\partial r} + \frac{1}{r}\frac{\partial (u_\theta u_z)}{\partial \theta} + \frac{\partial (u_z^2)}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial z} + \frac{1}{\rho}\frac{\partial (r\tau_{rz})}{r\partial r} + \frac{1}{\rho r}\frac{\partial \tau_{\theta z}}{\partial \theta} + \frac{1}{\rho}\frac{\partial \tau_{zz}}{\partial z} + g_z, \tag{4.5}$$

together with a set of boundary- and initial conditions. The body forces are contained in the vector $\mathbf{g} = (g_r, g_\theta, g_z)$, and the viscous stresses are given by:

$$\tau_{rr} = 2\mu\frac{\partial u_r}{\partial r}, \quad \tau_{r\theta} = \mu\left(\frac{\partial u_\theta}{\partial r} - \frac{u_\theta}{r} + \frac{1}{r}\frac{\partial u_r}{\partial \theta}\right), \quad \tau_{rz} = \mu\left(\frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r}\right), \tag{4.6}$$

$$\tau_{\theta\theta} = 2\mu\left(\frac{u_r}{r} + \frac{1}{r}\frac{\partial u_\theta}{\partial \theta}\right), \quad \tau_{\theta z} = \mu\left(\frac{\partial u_\theta}{\partial z} + \frac{1}{r}\frac{\partial u_z}{\partial \theta}\right), \quad \tau_{zz} = 2\mu\frac{\partial u_z}{\partial z} \tag{4.7}$$

A Marker and Cell (MAC) arrangement of the flow unknowns according to Harlow and Welch [38] is used, in which the velocity components are located on the cell faces and the pressure is located in the cell centres (see figure 4.1).



(a) Cells near the axis.

(b) Cells away from the axis.

Figure 4.1: Staggered location of the flow variables $\mathbf{u}$ and p in the computational domain.

The Navier-Stokes equations in cylindrical coordinates 4.2-4.5 are discretized in space using the second order accurate conservative finite difference approach of Morinishi et al. [61]. In this approach, the staggered arrangement of the velocity components implies that there is a radial velocity component located at $r = 0$ in every cell. Its value is found by solving an adapted radial momentum equation at $r = 0$ and by applying a subsequent Cartesian reconstruction that leads to a well-defined, second order accurate treatment of the velocity at the coordinate origin. During test simulations, however, we have found that this approach, although ele-

gant and conservative, can lead to nonphysical flow behaviour near the axis. The same observation was made by Desjardins et al. [23], who traced the result back to an inconsistent discretization. For that reason, we use the method of Fukagata and Kasagi [33], where the radial velocity components at $r = 0$ are obtained from averaging of the surrounding values. Oud et al. [67] present a conservative finite difference discretization of the Navier-Stokes equations in cylindrical coordinates, but only in the case of constant viscosity flows. At this moment, the latter discretization cannot yet be applied due to the fact that, although the viscosity is formally constant within each phase, some smearing is introduced near the interface in our algorithm to simplify the implementation of the numerical jump conditions.

The choice of a cylindrical computational grid implies that angular velocities in the small grid cells close to $r = 0$ can cause a severe time step restriction when using explicit methods. Therefore the second order Implicit Midpoint integration method is used for the temporal advancement of the momentum equations as it provides unconditional stability without any artificial damping, even in the case of vanishing viscosity. The latter is a valuable property when applying the method in future simulations to turbulent flows. The flow field is advanced by using a common projection scheme that consists of a predictor step in which an updated velocity field is calculated that may not be divergence-free. This newly obtained velocity field is subsequently made divergence-free by using an updated pressure field that results from a Poisson equation. The density ρ required in the predictor equation is calculated as the volume-weighted average of the two (constant) densities ρ_0 and ρ_1 using the VoF value ψ:

$$\rho = \psi\rho_1 + (1 - \psi)\rho_0. \tag{4.8}$$

The viscosity μ follows from a step function $H_s = H_s(\phi(\mathbf{x}))$ that is smoothed using the Level Set function ϕ:

$$\mu = \mu_0 + H_s(\phi)(\mu_1 - \mu_0). \tag{4.9}$$

The assumption of a sharp interface that separates two incompressible, immiscible fluids leads to jump conditions for the flow variables at the interface. Both the density and the viscosity are constant in each fluid phase, and the jump value of their difference normal to the interface is denoted by $[\rho]$ and $[\mu]$, respectively. Because viscous flows are considered, the velocity is continuous across the interface

and hence $[\mathbf{u}] = 0$. The pressure is not continuous but satisfies:

$$[p] = \sigma\kappa + 2[\mu]\mathbf{n}^T \cdot \nabla\mathbf{u} \cdot \mathbf{n}, \tag{4.10}$$

with σ the surface tension coefficient, κ the interface curvature and $\mathbf{n}$ the interface normal vector. The Ghost Fluid method [51] is used to implement the pressure jump as an additional term on the right hand side of the Poisson equation.

4.2.2 Representation of the interface

The sharp interface between the fluids is reconstructed by using an algorithm based on commonly known dual interface methods that combine both the Level Set method and the Volume of Fluid method in an efficient way. Successful examples of this approach can be found in the work of Bourlioux [13], Sussman and Puckett [83] (CLSVOF), Van der Pijl et al. [90] (MCLS), Yang et al. [96] (ACLSVOF) and Sun and Tao [81] (VOSET). In essence, all hybrid methods aim at benefitting from the computationally cheap and straightforward interface representation that the Level Set method offers, while enforcing non-trivial (phase-wise) volume conservation through the VoF method. Both the Level Set function ϕ and the VoF function ψ are advected with the flow field, and a key element is the coupling between the two variables when the interface is constructed during each time step. The minimization problem of finding an interface to a given volume requires the formulation of an abstract function f, such that $f(\phi) = \psi$. The inverse of f is generally much more difficult to construct, and the main differences between the various hybrid methods usually lie in the construction of the function f and its inverse, and the subsequent solution strategy of the minimization problem. Other differences arise due to the availability of both the Level Set function and the VoF function for various sub-algorithms (either LS or VoF based), like the density/viscosity determination and the interface curvature calculation, sometimes without having an obvious best choice.

In our algorithm, both the Level Set function ϕ and the VoF function ψ are cell-centred values in the cylindrical grid. The time-dependent interface Γ in a computational cell Ω is represented by the zero iso-contour of the linearized Level Set function ϕ, i.e:

$$\Gamma(t) = \{\mathbf{x} \in \Omega \mid \phi(\mathbf{x}_0, t) + \nabla\phi(\mathbf{x}_0, t) \cdot (\mathbf{x} - \mathbf{x}_0) = 0\}. \tag{4.11}$$

To maintain accurate finite differences, the Level Set function is kept a signed distance function through a PDE-based reinitialization procedure due to Sussman et al. [84]. First order ENO spatial fluxes [77] and Euler forward integration in time are used, and the sub-cell fix of Russo and Smereka [73] is applied to make the fluxes truly upwind, thereby reducing the movement of the interface during the reinitialization process. For the primary interface curvature calculation, the VoF-based second order Height Function method [12, 18, 40] is extended to cylindrical coordinates. In the case of insufficient grid resolution for the Height Function method to work properly, the curvature is calculated based on the level set field to maintain robustness of the algorithm.

The next sections describe three elements of our interface algorithm that are specifically designed for flows in cylindrical coordinates. Because we have not found any similar 3D cylindrical interface algorithms in the literature, these elements are presented in more detail. First, in section 4.2.3, we propose an efficient numerical evaluation of the function f that provides a VoF value for a given Level Set interface. Then, in section 4.2.4, a conservative advection scheme for both the Level Set function and the VoF function is presented. Finally, in section 4.2.5, the accuracy of time-explicit VoF advection near the polar origin is analyzed, as well as a proposed method to mitigate the reduction of the accuracy.

4.2.3 Calculation of the VoF function f

For a given interface that is described by the zero level set of the function ϕ, the associated VoF value $\psi_\Omega \in [0,1]$ belonging to some cell Ω can be calculated as:

$$\psi_\Omega = \frac{1}{|\Omega|} \int_\Omega \mathcal{H}(\phi(\mathbf{x}))\, \mathrm{d}\mathbf{x}, \tag{4.12}$$

where $\mathcal{H}$ is the Heaviside function. For arbitrary ϕ this integral can generally not be evaluated in a straightforward way, but if ϕ is restricted to the subset of linear functions, then exact solutions can be readily found. More specifically, the VoF value ψ_Ω associated to a linear interface can be calculated as:

$$\psi_\Omega = \frac{1}{|\Omega|} \int_\Omega \mathcal{H}\left[\phi(\mathbf{x}_0) + \nabla\phi(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0)\right] \mathrm{d}\mathbf{x} = f\left(\phi(\mathbf{x}_0), \nabla\phi(\mathbf{x}_0)\right). \tag{4.13}$$

In the MCLS approach of Van der Pijl et al. [90], this integral is evaluated analytically, leading to an elegant, closed-form formulation of f. In our experience,

however, we found that limit cases (i.e. when one or more elements of the interface normal vector become zero) occasionally affect the robustness of the algorithm, and we have implemented a stable alternative based on exact numerical integration. The availability of the Level Set function gives a tremendous simplification, as will be demonstrated. Consider an arbitrary computational cell as in figure 4.2 containing the dashed linear interface that encloses the associated gray volume (notice that a linear interface in cylindrical coordinates is curved in Cartesian space).

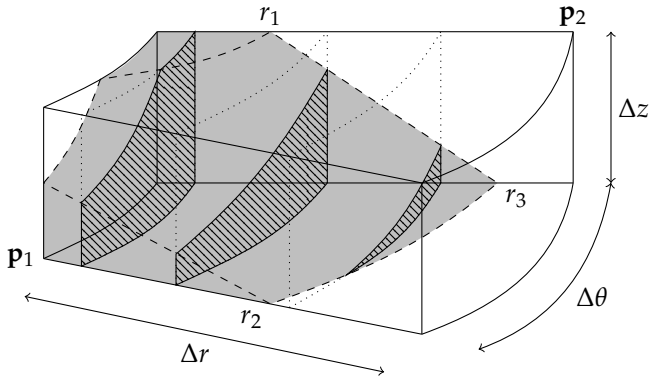


Figure 4.2: Subdivision of the volume into domains based on cross section similarity.

1. For the given cell, we start by calculating the level set values in the eight vertices of the cell based on the linearized level set function around the cell centre $\mathbf{x}_0 = (r_i, \theta_j, z_k)$, i.e:

$$\phi(r,\theta,z) = \phi(\mathbf{x}_0) + D_r(\mathbf{x}_0)(r - r_i) + D_\theta(\mathbf{x}_0)(\theta - \theta_j) + D_z(\mathbf{x}_0)(z - z_k). \quad (4.14)$$

The partial derivatives D_r, D_θ and D_z are determined by central differences of the level set function:

$$D_r(\mathbf{x}_0) = \frac{\phi_{i+1,j,k} - \phi_{i-1,j,k}}{2\Delta r}, \quad (4.15)$$

$$D_\theta(\mathbf{x}_0) = \frac{\phi_{i,j+1,k} - \phi_{i,j-1,k}}{2\Delta\theta}, \quad (4.16)$$

$$D_z(\mathbf{x}_0) = \frac{\phi_{i,j,k+1} - \phi_{i,j,k-1}}{2\Delta z}. \quad (4.17)$$

As an example, evaluation of expression 4.14 in the point $\mathbf{p}_1$ in figure 4.2 gives:

$$\phi(\mathbf{p}_1) = \phi(\mathbf{x}_0) - D_r(\mathbf{x}_0)\frac{\Delta r}{2} - D_\theta(\mathbf{x}_0)\frac{\Delta\theta}{2} - D_z(\mathbf{x}_0)\frac{\Delta z}{2}, \quad (4.18)$$

while evaluation in point $\mathbf{p}_2$ gives:

$$\phi(\mathbf{p}_2) = \phi(\mathbf{x}_0) + D_r(\mathbf{x}_0)\frac{\Delta r}{2} + D_\theta(\mathbf{x}_0)\frac{\Delta \theta}{2} + D_z(\mathbf{x}_0)\frac{\Delta z}{2}. \tag{4.19}$$

Whether or not the cell contains an interface is now easily checked by considering the maximum and minimum of the set of vertex values.

2. If an interface is found to be present, a sweep is done along the four ribs in radial direction using the level set values in the vertices to detect any intersections with the interface. Detection is as trivial as checking the sign of the product of the two level set values on both ends of the rib. Determining the distance r_{int} from the radial face of the cell is done through (exact) linear interpolation, conveniently expressed in terms of the vertex values:

$$r_{int} = \frac{|\phi_i|\Delta r}{|\phi_i| + |\phi_o|}, \tag{4.20}$$

where ϕ_i is the level set value of the radially inner most vertex and ϕ_o the value of the outer vertex. The collection of radial coordinates of the intersections is then stored as a sorted array. In the example in figure 4.2, the locations are denoted as r_1, r_2 and r_3.

3. The sorted list of radial locations is now used to identify a number of domains with similar cross sectional polygons in the $(\theta - z)$-plane. The three-dimensional integral in equation 4.13 is split up into a two-dimensional integral over a cross section in the $(\theta - z)$-plane and a resulting one-dimensional integral in the radial direction over the cross sections. In the example of figure 4.2, a total of four regions can be identified. The part of the cell with $[r_{min}, r_1]$ has a rectangular cross section with one corner missing (the left hatched plane). The region $[r_1, r_2]$ has a trapezoidal cross section (the middle hatched plane) and region $[r_2, r_3]$ has a triangular cross section (the right hatched plane). Region $[r_3, r_{max}]$ contains no interface and hence no cross section.

4. Consider the first domain $[r_{min}, r_1]$ of figure 4.2. For any radial location in this domain, the cross section in the $(\theta - z)$-plane is similar to figure 4.3.

   For a fixed radial location r, we can once again define the level set values in the four corners using the cell level set value $\phi_{i,j,k}$ and the partial derivatives from equations 4.15-4.17. The intersections of the interface polygon with the ribs

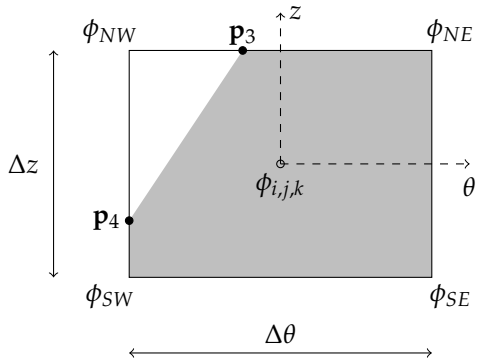


Figure 4.3: Integration of a 2D polygon in the $(\theta - z)$-plane.

are found through linear interpolation, completely analogous to the process of equation 4.20:

$$\theta_{\mathbf{p}_3} \sim \frac{|\phi_{NW}|\Delta\theta}{|\phi_{NW}| + |\phi_{NE}|}, \qquad z_{\mathbf{p}_4} \sim \frac{|\phi_{SW}|\Delta z}{|\phi_{SW}| + |\phi_{NW}|}, \tag{4.21}$$

for the points $\mathbf{p}_3$ and $\mathbf{p}_4$ in figure 4.3. With the ordered set of the N coordinates $\{(\theta_i, z_i)\}_{i=0}^{N-1}$ of the polygon then known, the area A is calculated as the convex hull of the coordinate set using the expression:

$$A = \frac{1}{2}\sum_{i=0}^{N-1}(\theta_i z_{i+1} - \theta_{i+1} z_i). \tag{4.22}$$

This calculation is particularly robust, even when the interface lies on a cell vertex, or worse, when it coincides with a cell rib. Clearly, the resulting polygon area $A = A(r)$ is a function of the radial location within the domain $[r_{min}, r_1]$.

5. A key observation is that the cross sectional area A varies quadratically in r for linear interfaces. The remaining one-dimensional integral in the radial direction $\int_r A(r) r \, dr$ is therefore solved exactly using a two-point Gauss-Legendre quadrature method. This method is particularly suitable as its nodes are internal points, and therefore they do not lie on the domain boundaries where the function evaluation may be relatively inaccurate. Every domain thus requires the determination of two cross sectional areas at pre-defined radial Gauss-Legendre points to calculate the volume of the interface polyhedron in that domain.

Steps 4 and 5 are repeated for all domains determined in step 1, and the total volume follows by adding all sub-volumes. Note that, with only minimal changes,

the algorithm can be applied to cells adjacent to the cylinder origin as well. This relatively simple sequence of calculations, solely based on a small number of vertex level set values, results in an efficient function f such that $\psi = f(\phi, \nabla\phi)$.

4.2.4 Advection of the interface

The transport of the interface variables is discussed in this section. Special attention is required for the VoF fluxes because the angular velocity becomes very high in the vicinity of the origin at $r = 0$. This issue is further addressed in section 4.2.5.

Both the level set function ϕ and the VoF function ψ are advected with the underlying flow field $\mathbf{u}$:

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0, \qquad \frac{\partial \psi}{\partial t} + \mathbf{u} \cdot \nabla \psi = 0. \tag{4.23}$$

For both processes, the second order directional split algorithm of Weymouth and Yue [94] is used. For the level set ϕ centred in cell (i, j, k), it is given by:

$$\frac{\phi^{(*)} - \phi^{(n)}}{\Delta t} = \frac{(r\widetilde{F}^{(n)})_{i-\frac{1}{2}} - (r\widetilde{F}^{(n)})_{i+\frac{1}{2}}}{r_i \Delta r} + \phi^{(n)} \frac{(ru_r^{(n)})_{i+\frac{1}{2}} - (ru_r^{(n)})_{i-\frac{1}{2}}}{r_i \Delta r}, \tag{4.24}$$

$$\frac{\phi^{(**)} - \phi^{(*)}}{\Delta t} = \frac{\widetilde{G}^{(*)}_{j-\frac{1}{2}} - \widetilde{G}^{(*)}_{j+\frac{1}{2}}}{r_i \Delta \theta} + \phi^{(n)} \frac{u^{(n)}_{\theta_{j+\frac{1}{2}}} - u^{(n)}_{\theta_{j-\frac{1}{2}}}}{\Delta \theta}, \tag{4.25}$$

$$\frac{\phi^{(n+1)} - \phi^{(**)}}{\Delta t} = \frac{\widetilde{H}^{(**)}_{k-\frac{1}{2}} - \widetilde{H}^{(**)}_{k+\frac{1}{2}}}{\Delta z} + \phi^{(n)} \frac{u^{(n)}_{z_{k+\frac{1}{2}}} - u^{(n)}_{z_{k-\frac{1}{2}}}}{\Delta z}, \tag{4.26}$$

with $\widetilde{F}_{i\pm\frac{1}{2}}$, $\widetilde{G}_{j\pm\frac{1}{2}}$ and $\widetilde{H}_{k\pm\frac{1}{2}}$ being the scalar level set fluxes in the radial, angular and axial directions, respectively, and the time levels $t^{(*)}$ and $t^{(**)}$ being two intermediary time levels between $t^{(n)}$ and $t^{(n+1)}$. The fluxes are determined in a similar way as the upwind approach of Sussman and Puckett [83]. For cell (i, j, k), the flux $\widetilde{F}_{i+\frac{1}{2}}$ at the positive face at $r_{i+\frac{1}{2}}$ is given by:

$$\widetilde{F}_{i+\frac{1}{2}} = \begin{cases} u_{r_{i+\frac{1}{2}}} \left(\phi_{i,j,k} + \dfrac{1}{2} \left(\Delta r - u_{r_{i+\frac{1}{2}}} \Delta t \right) \right) \dfrac{\phi_{i+1,j,k} - \phi_{i-1,j,k}}{2\Delta r}, & u_{r_{i+\frac{1}{2}}} \geq 0 \\ u_{r_{i+\frac{1}{2}}} \left(\phi_{i+1,j,k} - \dfrac{1}{2} \left(\Delta r + u_{r_{i+\frac{1}{2}}} \Delta t \right) \right) \dfrac{\phi_{i+2,j,k} - \phi_{i,j,k}}{2\Delta r}, & u_{r_{i+\frac{1}{2}}} < 0. \end{cases} \tag{4.27}$$

The angular flux $\widetilde{G}_{j+\frac{1}{2}}$ is given by:

$$\widetilde{G}_{j+\frac{1}{2}} = \begin{cases} u_{\theta_{j+\frac{1}{2}}} \left(\phi_{i,j,k} + \frac{1}{2} \left(\Delta\theta - \frac{u_{\theta_{j+\frac{1}{2}}} \Delta t}{r_i} \right) \right) \frac{\phi_{i,j+1,k} - \phi_{i,j-1,k}}{2\Delta\theta}, & u_{\theta_{j+\frac{1}{2}}} \geq 0 \\ u_{\theta_{j+\frac{1}{2}}} \left(\phi_{i,j+1,k} - \frac{1}{2} \left(\Delta\theta + \frac{u_{\theta_{j+\frac{1}{2}}} \Delta t}{r_i} \right) \right) \frac{\phi_{i,j+2,k} - \phi_{i,j,k}}{2\Delta\theta}, & u_{\theta_{j+\frac{1}{2}}} < 0, \end{cases} \tag{4.28}$$

and for the axial direction the flux $\widetilde{H}_{k+\frac{1}{2}}$ is given by:

$$\widetilde{H}_{k+\frac{1}{2}} = \begin{cases} u_{z_{k+\frac{1}{2}}} \left(\phi_{i,j,k} + \frac{1}{2} \left(\Delta z - u_{z_{k+\frac{1}{2}}} \Delta t \right) \right) \frac{\phi_{i,j,k+1} - \phi_{i,j,k-1}}{2\Delta z}, & u_{z_{k+\frac{1}{2}}} \geq 0 \\ u_{z_{k+\frac{1}{2}}} \left(\phi_{i,j,k+1} - \frac{1}{2} \left(\Delta z + u_{z_{k+\frac{1}{2}}} \Delta t \right) \right) \frac{\phi_{i,j,k+2} - \phi_{i,j,k}}{2\Delta z}, & u_{z_{k+\frac{1}{2}}} < 0 \end{cases} \tag{4.29}$$

For the VoF function, we denote the radial, angular and axial fluxes by $F_{i\pm\frac{1}{2}}$, $G_{j\pm\frac{1}{2}}$ and $H_{k\pm\frac{1}{2}}$, respectively, and the advection scheme is given by:

$$\psi^{(*)} - \psi^{(n)} = \frac{1}{|\Omega|} \left(F^{(n)}_{i-\frac{1}{2}} - F^{(n)}_{i+\frac{1}{2}} \right) + \mathcal{H}(\phi^{(n)}) \Delta t \frac{(r u_r^{(n)})_{i+\frac{1}{2}} - (r u_r^{(n)})_{i-\frac{1}{2}}}{r_i \Delta r}, \tag{4.30}$$

$$\psi^{(**)} - \psi^{(*)} = \frac{1}{|\Omega|} \left(G^{(*)}_{j-\frac{1}{2}} - G^{(*)}_{j+\frac{1}{2}} \right) + \mathcal{H}(\phi^{(n)}) \Delta t \frac{u^{(n)}_{\theta_{j+\frac{1}{2}}} - u^{(n)}_{\theta_{j-\frac{1}{2}}}}{r_i \Delta\theta}, \tag{4.31}$$

$$\psi^{(n+1)} - \psi^{(**)} = \frac{1}{|\Omega|} \left(H^{(**)}_{k-\frac{1}{2}} - H^{(**)}_{k+\frac{1}{2}} \right) + \mathcal{H}(\phi^{(n)}) \Delta t \frac{u^{(n)}_{z_{k+\frac{1}{2}}} - u^{(n)}_{z_{k-\frac{1}{2}}}}{\Delta z}, \tag{4.32}$$

with $\mathcal{H}$ the Heaviside function and $|\Omega| = r_i \Delta r \Delta\theta \Delta z$ the volume of the cell. The time-explicit second term on the right hand side guarantees that the volume is conserved up to machine precision as long as the Courant number of the flow field, calculated as $\Delta t(u_r/\Delta r + u_\theta/(r_i\Delta\theta) + u_z/\Delta z)$, remains less than 0.5. After application of the divergence theorem, the radial VoF flux $F^{(n)}_{i+\frac{1}{2}}$ becomes:

$$F^{(n)}_{i+\frac{1}{2}} = r_{i+\frac{1}{2}} \int_{t_n}^{t_n+\Delta t} \int_{z_{k-\frac{1}{2}}}^{z_{k+\frac{1}{2}}} \int_{\theta_{j-\frac{1}{2}}}^{\theta_{j+\frac{1}{2}}} \chi(r_{i+\frac{1}{2}}, \theta, z, t) u_r(r_{i+\frac{1}{2}}, \theta, z, t) \, \mathrm{d}\theta \, \mathrm{d}z \, \mathrm{d}t, \tag{4.33}$$

where χ is the colour function. To eliminate the time dependency from this expression, the motion of the colour function in the radial direction only is considered. It

is observed that the continuity equation restricted to the radial direction:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(ru_r\right) = 0, \quad r > 0, \tag{4.34}$$

implies that, at some fixed time $t \in [t_n, t_n + \Delta t]$, the characteristics are curved and satisfy:

$$r(t)u_r(r,\theta,z,t) = r_{i+\frac{1}{2}}u_r(r_{i+\frac{1}{2}},\theta,z,t). \tag{4.35}$$

Integration then yields:

$$\int_{r(t)}^{r_{i+\frac{1}{2}}} r\,\mathrm{d}r = \int_t^{t_n+\Delta t} r_{i+\frac{1}{2}}u_r(r_{i+\frac{1}{2}},\theta,z,\tau)\,\mathrm{d}\tau, \tag{4.36}$$

so that:

$$r^2(t) = r^2_{i+\frac{1}{2}} - 2\int_t^{t_n+\Delta t} r_{i+\frac{1}{2}}u_r(r_{i+\frac{1}{2}},\theta,z,\tau)\,\mathrm{d}\tau, \tag{4.37}$$

for some $t \in [t_n, t_n + \Delta t]$. If we now assume $u_r(r_{i+\frac{1}{2}},\theta,z,t)$ to be constant in time during the interval $[t_n, t_n + \Delta t]$ and equal to $u_r(r_{i+\frac{1}{2}},\theta,z,t_n)$, then the integral reduces to:

$$r^2(t) = r^2_{i+\frac{1}{2}} - 2r_{i+\frac{1}{2}}u_r(r_{i+\frac{1}{2}},\theta,z,t_n)(t_n + \Delta t - t). \tag{4.38}$$

A change of variable finally allows us to rewrite equation 4.33 as:

$$F^{(n)}_{i+\frac{1}{2}} = \int_{z_{k-\frac{1}{2}}}^{z_{k+\frac{1}{2}}}\int_{\theta_{j-\frac{1}{2}}}^{\theta_{j+\frac{1}{2}}}\int_{r^*}^{r_{i+\frac{1}{2}}} \chi(r,\theta,z,t_n)r\,\mathrm{d}r\,\mathrm{d}\theta\,\mathrm{d}z, \tag{4.39}$$

with:

$$r^* = \sqrt{r_{i+\frac{1}{2}}\left(r_{i+\frac{1}{2}} - 2u_r(r_{i+\frac{1}{2}},\theta,z,t_n)\Delta t\right)}. \tag{4.40}$$

The flux $F^{(n)}_{i+\frac{1}{2}}$ is subsequently calculated by substitution of $\mathcal{H}(\phi)$ for the colour function χ and the techniques described in section 4.2.3. Notice how the radial dimension of the donating region $[r^*, r_{i+\frac{1}{2}}]$ differs from the Cartesian case where generally $r^* = r_{i+\frac{1}{2}} - u_r\Delta t$ so that the cell geometry is taken into account. In particular, at the polar axis $r = 0$, the inner cell face collapses and no flux exists as there is no cell face for a flux to move through. This is consistently modelled when using the flux region given by expression 4.40.

The angular and axial VoF fluxes are given by:

$$G^{(*)}_{j+\frac{1}{2}} = \int_{t_n}^{t_n+\Delta t} \int_{z_{k-\frac{1}{2}}}^{z_{k+\frac{1}{2}}} \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \chi(r, \theta_{j+\frac{1}{2}}, z, t) u_\theta(r, \theta_{j+\frac{1}{2}}, z, t) r \, \mathrm{d}r \, \mathrm{d}z \, \mathrm{d}t, \tag{4.41}$$

$$H^{(*)}_{k+\frac{1}{2}} = \int_{t_n}^{t_n+\Delta t} \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \int_{\theta_{j-\frac{1}{2}}}^{\theta_{j+\frac{1}{2}}} \chi(r, \theta, z_{k+\frac{1}{2}}, t) u_z(r, \theta, z_{k+\frac{1}{2}}, t) r \, \mathrm{d}r \, \mathrm{d}\theta \, \mathrm{d}t, \tag{4.42}$$

and since the continuity equation restricted to the respective directions is:

$$\frac{1}{r}\frac{\partial u_\theta}{\partial \theta} = 0, \qquad \frac{\partial u_z}{\partial z} = 0, \tag{4.43}$$

the resulting VoF fluxes based on a divergence-free velocity field become:

$$G^{(*)}_{j+\frac{1}{2}} = \int_{z_{k-\frac{1}{2}}}^{z_{k+\frac{1}{2}}} \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \int_{\theta^*}^{\theta_{j+\frac{1}{2}}} \chi(r, \theta, z, t_n) r \, \mathrm{d}\theta \, \mathrm{d}r \, \mathrm{d}z, \tag{4.44}$$

$$H^{(*)}_{k+\frac{1}{2}} = \int_{\theta_{j-\frac{1}{2}}}^{\theta_{j+\frac{1}{2}}} \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \int_{z^*}^{z_{k+\frac{1}{2}}} \chi(r, \theta, z, t_n) r \, \mathrm{d}z \, \mathrm{d}r \, \mathrm{d}\theta, \tag{4.45}$$

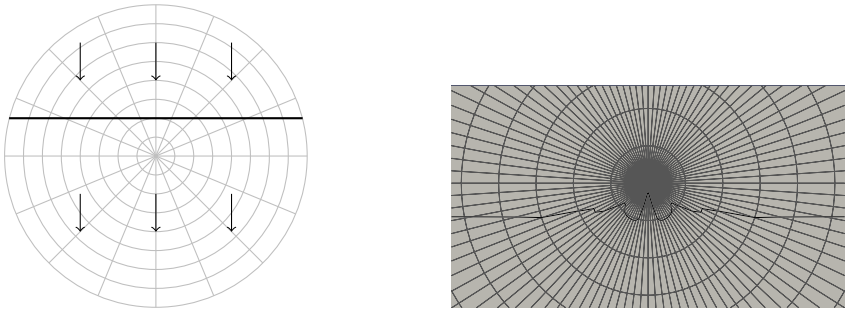
where:

$$\theta^* = \theta_{j+\frac{1}{2}} - \frac{u_\theta(r, \theta_{j+\frac{1}{2}}, z, t_n)\Delta t}{r_i}, \qquad z^* = z_{k+\frac{1}{2}} - u_z(r, z_{k+\frac{1}{2}}, t_n)\Delta t. \tag{4.46}$$

Conservation of mass (equivalent to conservation of volume in the incompressible case) in a simulation model is of paramount importance for the results to have a clear physical meaning. Furthermore, loss of mass introduces loss of momentum in the numerical scheme. Generic VoF methods frequently suffer from volume loss during advection; even if the advection scheme is conservative, there is no intrinsic mechanism that prevents computational cells from overfilling (VoF larger than one) or from over-emptying (VoF less than zero). These nonphysical occurrences need to be subsequently corrected to obtain a physical solution through processes that usually either change the total volume (by adding or removing volume) or redistribute the total volume to eliminate invalid VoF values. The advection scheme used in our algorithm appears to be the only split flux scheme that is conservative and which simultaneously prevents nonphysical VoF values up to rounding errors.

4.2.5 VoF advection accuracy near the polar origin

Besides placing a severe time step constraint on explicit time integration methods, the geometry of the cylindrical cells near the polar axis at $r = 0$ also significantly affects the accuracy of the VoF advection when a planar interface crosses the origin. To demonstrate this phenomenon, consider the case of downward moving planar interface in the polar plane, initially positioned as shown in figure 4.4a.



(a) Initial interface position.

(b) Interface passes through the origin.

Figure 4.4: Simulation of planar interface in the polar plane. Grid size $N_r \times N_\theta$ is 20×100.

The Cartesian velocity components (u_x, u_y) are taken as $(0, -1)$, and the interface remains exactly horizontal as it moves through the domain. The numerical result, using the explicit VoF advection scheme of section 4.2.4, shows a significantly different interface shape as it passes the origin, which is shown in figure 4.4b: a cusp is formed around the origin which takes a significant amount of time to disappear as the interface continues to move downwards. The VoF advection is responsible for this behaviour, since the level set function and its iso-contours are not affected[†] by the presence of the origin (besides the time step constraint on its advection). For an analysis of the VoF advection near the origin, the contravariant components $(\dot{r}, \dot{\theta})$ of the velocity field are found through the coordinate transformation:

$$x = r\cos\theta, \qquad y = r\sin\theta, \tag{4.47}$$

and for the given Cartesian velocity vector, these contravariant components have the values:

$$\dot{r} = -\sin\theta, \qquad \dot{\theta} = -\frac{\cos\theta}{r}. \tag{4.48}$$

The dot refers to time differentiation here, and for the angular movement in partic-

[†]This is easily verified because the algorithm implementation also allows a level set-only interface by simply disabling the coupling to the VoF after advection of both quantities.

ular, the path of a particle is thus determined by the differential equation:

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = -\frac{\cos\theta}{r}. \tag{4.49}$$

It is assumed that, just before crossing the origin, the initial location of the interface is $(r, \theta) = (r, 0)$ (only the right half of the domain is considered for simplicity). Furthermore, only the advection in angular direction is considered while the radial transport is momentarily neglected. Integrating equation 4.49 over a temporal interval $[0, \Delta t]$ with $\theta(0) = 0$ then yields the expression of the traversed angle of the interface:

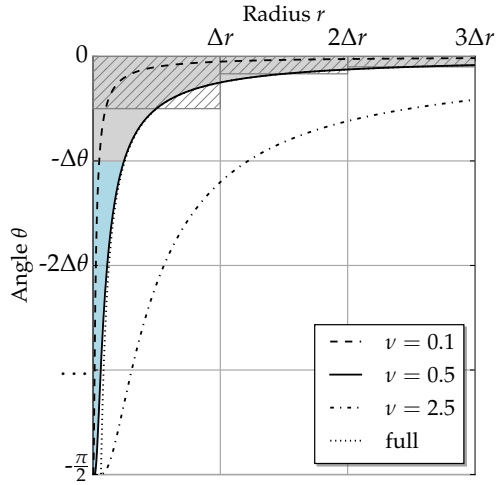
$$\theta(r, \Delta t) = -\arctan\left(\sinh\left(\frac{\Delta t}{r}\right)\right). \tag{4.50}$$

This expression relates the angular location of the interface after a time step Δt for all radial coordinates. In particular, a particle on the interface close to $r = 0$ moves to $\theta = -\pi/2$ instantaneously when r goes to zero, irrespective of the time step; this follows from the limit:

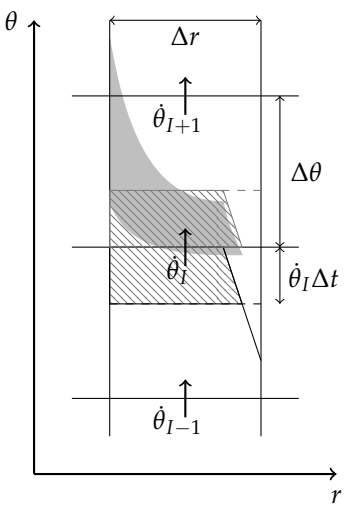
$$\lim_{r\to 0^+} \arctan\left(\sinh\left(\frac{\Delta t}{r}\right)\right) = \frac{\pi}{2}, \qquad \forall \Delta t > 0. \tag{4.51}$$

For the described configuration, figure 4.5a shows the exact interface location based on expression 4.50 on a grid in computational space after a single time step of varying magnitude (made dimensionless as a Courant number ν). The asymptotic movement near $r = 0$ is clearly visible, and it can be concluded that the interface always traverses all inner cells with $-\pi/2 < \theta < 0$ for every positive time step. Furthermore, figure 4.5a shows that angular advection of the interface causes subsequent radial movement of the interface in all inner cells with $-\pi/2 \leq \theta \leq -\Delta\theta$. This comes as no surprise, however, as a radial flux through the origin is absent due to expression 4.40. For completeness, the interface location after a full (both radial and angular) advection step is also shown for the case when $\nu = 0.5$. For angles close to zero, there is no distinction between the full and the angular result because the radial velocity is nearly zero at these locations. When $\theta \to -\pi/2$, where the radial velocity is maximum, the discrepancy increases and can be attributed to (the absence of) the radial advection of the interface.

The result of the exact angular movement of the interface near the origin now allows a comparison with the numerical VoF flux approximation as presented in section 4.2.4. The finite volume fluxes for the angular direction, as shown in expressions 4.44 and 4.46, assume an angular velocity $\dot{\theta}$ that is constant over a cell face Δr (or



(a) Exact angular interface advection in the inner most cells for different Courant numbers.



(b) Exact (solid fill) vs approximated (hatched) image of a donating region in cells near the origin.

Figure 4.5: Schematic overview of the VoF advection near the origin at $r = 0$.

$\Delta r \Delta z$ in 3D). Based on this velocity, the corresponding donating region is obtained by backtracing, and it therefore always has a rectangular shape in computational space. The resulting angular VoF fluxes based on expressions 4.44 and 4.46 for the case when $\nu = 0.5$ are shown as the hatched areas in figure 4.5a. It can be seen that the numerical fluxes approximate the exact gray surfaces reasonably well, even near the origin. The problem, however, is the time-explicit nature of the VoF fluxes that prevents instantaneous filling of multiple cells during a single time step. The cyan coloured surface is to be divided over all cells with $-\pi/2 < \theta < -\Delta\theta$, but it can not be captured by explicit VoF fluxes due to their CFL restriction. Decreasing the time step reduces the inaccuracy by reducing the amount of fluid that is instantaneously moved through the inner layer of cells (compare with the loss of $\nu = 0.1$), but for negligible loss this may lead to unrealistically small time steps that will make simulations impractical.

For all but the inner layer of cells, the situation in the case of a generic interface is shown in figure 4.5b. The explicit VoF flux approximation assumes a translation of the donating region of length $\dot{\theta}_I \Delta t$ in computational space, whereas the true image of the donating region becomes skewed due to the exact non-constant angular velocity that grows as $1/r$ when r decreases. For these cells, however, a sufficiently small CFL number exists to guarantee that not more than a single cell is traversed

by the interface during a single time step. For a given radial location of the cell, its value can be obtained by using expression 4.50.

The analysis above shows that explicit VoF advection is not capable of correctly capturing the dynamics of the interface close to the origin, and explains the numerical result of figure 4.4b. The cusp originates due to the delay in the angular VoF advection in the inner layer of cells that can only fill up to a single cell per time step. This simple example demonstrates that explicit VoF advection is likely to be insufficiently accurate when used in the study of stratified interface instabilities, and therefore improvements are required. An implicit treatment of the VoF fluxes appears an attractive alternative, because it allows the interface to cross multiple cells in a single time step without CFL constraint. The main disadvantage, unfortunately, is a highly complex and computationally expensive implementation, with barely any supporting references from the literature. Despite this, it is strongly recommended to evaluate the implicit VoF advection in further research, but it is outside the scope of the present work. Instead, we have devised a relatively simple alternative method that provides satisfactory results for the test cases in this chapter. In essence, it is based on an interface approximation provided by the level set function at the new time level, and therefore it can be loosely regarded as an approximation of implicit VoF advection. More specifically, a VoF redistribution algorithm is employed based on the observation that near the origin, the level set advection, although not volume-conserving, is more physically accurate than the VoF field. Consider a subset of cells $\{V_i\}$ of the complete discrete domain, and let V_f and V_ψ be the amount of volume in this set according to the linearized level set and the VoF, respectively:

$$V_f := \sum_i f(\phi_i, \nabla\phi_i)|V_i|, \qquad V_\psi := \sum_i \psi_i |V_i|, \tag{4.52}$$

where $|V_i| = r_i \Delta r \Delta\theta \Delta z$ is the volume of cell i and f is the VoF function described in section 4.2.3. The complementary volumes V_f^c and V_ψ^c are defined analogously:

$$V_f^c := \sum_i \left(1 - f(\phi_i, \nabla\phi_i)\right)|V_i|, \qquad V_\psi^c := \sum_i \left(1 - \psi_i\right)|V_i|, \tag{4.53}$$

and note that both $V_f + V_f^c$ and $V_\psi + V_\psi^c$ sum up to the volume of the subset. Depending on the ratio V_ψ / V_f, the volume V_ψ based on the VoF field $\{\psi_i\}$ is then redistributed over $\{V_i\}$ according to a weight factor based on the level set volume

prediction $f(\phi, \nabla\phi)$ into an updated VoF field $\{\widetilde{\psi}_i\}$ through:

$$\widetilde{\psi}_i = \begin{cases} f(\phi_i, \nabla\phi_i)\dfrac{V_\psi}{V_f}, & V_\psi \leq V_f, \\ 1 - (1 - f(\phi_i, \nabla\phi_i))\dfrac{V_\psi^c}{V_f^c}, & V_\psi > V_f \end{cases} \tag{4.54}$$

Note that we must distinguish between $V_\psi / V_f \leq 1$ and $V_\psi / V_f > 1$ in order to prevent overfilling cells. Multiplying with the cell volume and subsequent summation shows that the redistribution is conservative, i.e. $\sum_i \psi_i |V_i| = \sum_i \widetilde{\psi}_i |V_i|$. Furthermore, the algorithm can be applied at any given subset, although without mixed cells it leaves the original VoF field unchanged.

To demonstrate the effect of the redistribution algorithm, it is applied to the initial test case of a horizontal interface in a uniform velocity field $u_y = -1$. Figure 4.6 shows the resulting interface when it traverses the origin. The improvement due to the redistribution is clearly noticeable, and further experiments on finer grids show similar results. For the pure explicit VoF advection, the magnitude of the error does not decrease upon refinement. This corresponds with the explanation given above.

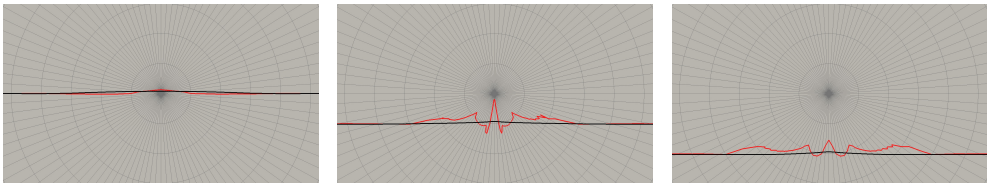


Figure 4.6: Interface evolution (from left to right) around the origin in a downwards moving flow field. The red line represents the interface from explicit VoF advection without redistribution, and the black line is the interface that results from explicit VoF advection with subsequent redistribution according to equation 4.54.

Since the exact interface location is known, the accuracy of the proposed VoF redistribution method can be quantified by looking at the L_1-error of the VoF values, defined as:

$$\|\psi - \psi_{ex}\|_1 := \sum_I |\psi_i - \psi_i^{ex}| r_I \Delta r \Delta\theta \Delta z, \tag{4.55}$$

with I some multi-index, ψ and ψ^{ex} the calculated and exact VoF values at a predefined time level, respectively, and $r_I \Delta r \Delta\theta \Delta z$ the volume of cell I. For various grids, table 4.1 shows the resulting values.

For the current interface configuration, it is observed that the error reduces with grid refinement, although without a consistent rate. Performing the same simulations without using the redistribution algorithm leads to an error that does not

Table 4.1: L_1-error of the VoF advection with redistribution.

Grid size $(N_r \times N_\theta \times N_z)$	VoF error $\|\psi - \psi_{ex}\|_1$
$5 \times 10 \times 2$	$1.652 \cdot 10^{-8}$
$10 \times 50 \times 2$	$8.973 \cdot 10^{-9}$
$20 \times 100 \times 2$	$2.111 \cdot 10^{-9}$
$40 \times 200 \times 2$	$1.195 \cdot 10^{-9}$

reduce at all. Therefore, this example clearly shows the added benefit of the level set function and how it can contribute to a significantly improved interface accuracy. Nonetheless, we recognize that the proposed VoF redistribution algorithm is rather elementary, and further research into improving the accuracy of the VoF advection near the origin is required.

4.3 Modal stability of a cylindrical interface

In this section, we test the capability of our algorithm to capture an infinitesimal perturbation of a cylindrical interface in a shear flow, and to predict the associated growth factor. Modal stability theory is used for the analysis, and an approach similar to the work in Cartesian domains by Bagué et al. [4] is followed. The cylindrical Orr-Sommerfeld equation is derived based on a streamfunction formulation of the perturbed velocities. The interface boundary conditions are then found, and the resulting eigenvalue problem is solved using a Chebyshev collocation method. We find that there exists a region of small real wave numbers for which the flow is unstable; here an initial perturbation diverges according to a theoretical growth factor as given by the solution of the eigenvalue problem. The aim is to test whether for a given perturbed flow field and interface, the simulated growth factor in our algorithm matches the value from the modal stability theory.

The computational domain consists of a cylinder with radius R, and the two fluids are transported as core-annular flow. As such, they are initially separated at the interface at $r = \frac{R}{2}$. The density and viscosity of the fluid in the core and the fluid in the annulus are denoted by (ρ_c, μ_c) and (ρ_a, μ_a), respectively. The amplitude of the interface centred around $r = \frac{R}{2}$ is denoted by $\eta(t, z)$, and the boundary layer thickness in both phases is δ. Figure 4.7 gives a schematic view of the setup of the case.

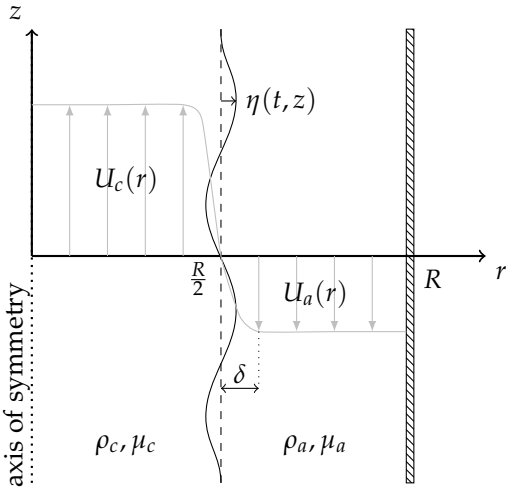


Figure 4.7: Schematic of axisymmetric parallel shear flow with a wavy interface.

4.3.1 The base flow

For the base flow, we choose a shear flow profile based on the solution of an impulsively accelerated annular flow between two positive radii. With the convective terms and the pressure gradient being equal to zero, the resulting axial momentum equation reduces to a radial diffusion equation with a Bessel series solution. Because this solution contains Bessel functions of the second kind, it becomes unbounded as $r \to 0$, and a distinction is made between the core flow (occupying the region $0 \leq r < \frac{R}{2}$) and the annular flow (occupying the region $\frac{R}{2} \leq r \leq R$). The profile of the annular flow U_a is described by:

$$U_a(r,t) = c_a \sum_{i=1}^{\infty} \frac{J_0(k_i R/2) f(k_i, r)}{J_0(k_i R/2) + J_0(k_i 3R/2)} e^{-k_i^2 \nu_a t}, \quad c_a \in \mathbb{R}, \tag{4.56}$$

where J_0 and Y_0 are Bessel functions of the first and second kind, respectively, and:

$$f(k,r) = J_0(kr) Y_0(k3R/2) - J_0(k3R/2) Y_0(kr), \tag{4.57}$$

with $\{k_i\}$ the infinite set of solutions that satisfy $f(k, R/2) = 0$. After taking limits, the profile for the core flow U_c becomes:

$$U_c(r,t) = c_c \sum_{i=1}^{\infty} \frac{J_0(l_i r)}{l_i J_1(l_i)} e^{-l_i^2 \nu_c t}, \quad c_c \in \mathbb{R}, \tag{4.58}$$

where $\{l_i\}$ is now the set of solutions of $J_0(lR/2) = 0$. In both expressions ν_a and ν_c represent the constant kinematic viscosities of the annular flow and the core flow, respectively. The profiles are such that the base flow has zero velocity at the interface, i.e. $U_c(R/2, t) = 0 = U_a(R/2, t)$. The time t effectively determines the size of the boundary layer δ (see figure 4.7) around the interface, defined as the location where the velocity equals 99% of the respective free stream velocity. Throughout our analysis, the boundary layers in both phases have the same thickness δ for simplicity. For a given value of δ, the associated times t_a and t_c have to be found by using a numerical root finding algorithm. For practical purposes, the infinite sums in expressions 4.56 and 4.58 are truncated after 150 terms; this number of terms is found to yield a sufficiently accurate representation with negligible oscillations for most cases, but the quality of the approximation rapidly deteriorates when the boundary layer becomes smaller. The velocity profile then approaches a step function, and the combination of the Gibbs phenomenon and round-off errors cause oscillations that impose a practical lower bound for the boundary layer thickness δ. The constants c_c and c_a are subsequently used to scale the resulting profiles. First, the value of c_c is defined such that $U_c(0, t_c) = 1$ m/s irrespective of the core fluid properties. The continuity of tangential shear stress, denoted by:

$$\mu_c \frac{\partial U_c}{\partial r}\left(\frac{R}{2}, t_c\right) = \mu_a \frac{\partial U_a}{\partial r}\left(\frac{R}{2}, t_a\right), \tag{4.59}$$

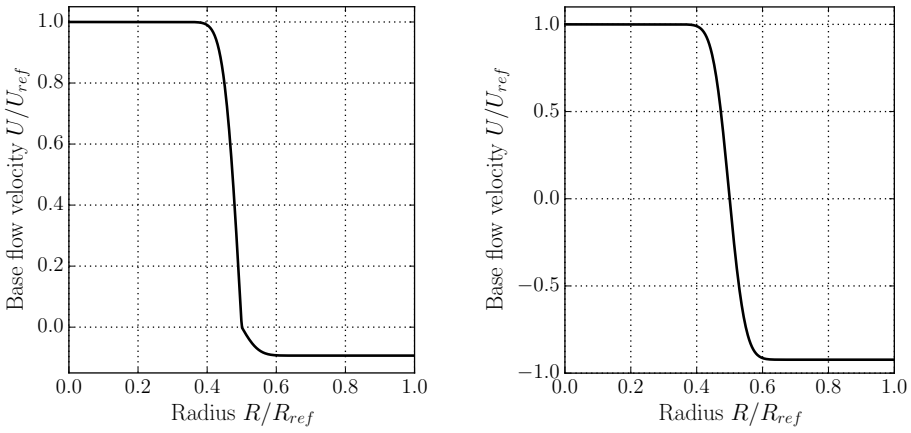
then dictates the value of the constant c_a. Finally, the base flow is defined as:

$$\begin{cases} U_c(r) = U_c(r, t_c), & 0 \leq r < \dfrac{R}{2}, \\ U_a(r) = U_a(r, t_a), & \dfrac{R}{2} \leq r < R. \end{cases} \tag{4.60}$$

Figure 4.8 shows two typical base flow profiles that are applied to the test cases that are considered.

We define the following dimensionless numbers. The viscosity ratio is given by $\gamma = \mu_c/\mu_a$ and the density ratio is given by $\zeta = \rho_c/\rho_a$. The Reynolds numbers for the core and the annulus are based on the core free stream velocity U_c:

$$\mathrm{Re}_c = \frac{\rho_c U_c(0)\delta}{\mu_c}, \qquad \mathrm{Re}_a = \frac{\rho_a U_c(0)\delta}{\mu_a}. \tag{4.61}$$



(a) Base flow profile with $Re_c = 10^4$.

(b) Base flow profile with $Re_c = 10^2$.

Figure 4.8: Axisymmetric base flow in the case of a cylinder of radius $R_{ref} = 1$ m and boundary layer thickness $\delta/R = 0.1$. The velocity U_{ref} is 1 m/s, and the interface is located at $R/R_{ref} = \frac{1}{2}$ where the flow velocity is zero.

The Weber numbers are defined as:

$$\mathrm{We}_c = \frac{\rho_c U_c(0)^2 \delta}{\sigma}, \qquad \mathrm{We}_a = \frac{\rho_a U_c(0)^2 \delta}{\sigma}. \tag{4.62}$$

4.3.2 Derivation of the perturbation equation

The base flow in the axisymmetric coordinate system is perturbed by the quantities $(\tilde{u}_r, \tilde{u}_z, \tilde{p})$ such that:

$$u_r = \tilde{u}_r, \quad u_z = U + \tilde{u}_z, \quad p = P + \tilde{p}, \tag{4.63}$$

where U is the axial base flow and P is the associated pressure field. Linearization of the Navier-Stokes equations around the base flow yields the following momentum equations:

$$\frac{\partial \tilde{u}_r}{\partial t} + U\frac{\partial \tilde{u}_r}{\partial z} + \frac{1}{\rho}\frac{\partial \tilde{p}}{\partial r} - \nu\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \tilde{u}_r}{\partial r}\right) - \frac{\tilde{u}_r}{r^2} + \frac{\partial^2 \tilde{u}_r}{\partial z^2}\right] = 0, \tag{4.64}$$

$$\frac{\partial \tilde{u}_z}{\partial t} + \tilde{u}_r\frac{\partial U}{\partial r} + U\frac{\partial \tilde{u}_z}{\partial z} + \frac{1}{\rho}\frac{\partial \tilde{p}}{\partial z} - \nu\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \tilde{u}_z}{\partial r}\right) + \frac{\partial^2 \tilde{u}_z}{\partial z^2}\right] = 0, \tag{4.65}$$

and the continuity equation:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \tilde{u}_r}{\partial r}\right) + \frac{\partial^2 \tilde{u}_z}{\partial z^2} = 0. \tag{4.66}$$

Assuming small perturbations, the perturbed velocity components can be expressed in terms of a Stokes streamfunction Ψ such that:

$$\tilde{u}_r = -\frac{1}{r}\frac{\partial \Psi}{\partial z}, \qquad \tilde{u}_z = \frac{1}{r}\frac{\partial \Psi}{\partial r}. \tag{4.67}$$

Two streamfunctions, Ψ_c and Ψ_a, are required for the core flow and the annular flow, respectively. It is postulated that these streamfunctions can be written as:

$$\Psi_c = \phi_c(r)e^{i\alpha(z-ct)}, \qquad 0 \leq r < \frac{R}{2}, \tag{4.68}$$

$$\Psi_a = \phi_a(r)e^{i\alpha(z-ct)}, \qquad \frac{R}{2} \leq r \leq R. \tag{4.69}$$

Here $\alpha \in \mathbb{R}$ is the wave number (we consider temporal instabilities only), and $c \in \mathbb{C}$. The split up in real and complex parts, i.e. $c = c_r + ic_i$, determines the phase velocity c_r and the growth rate αc_i of the unstable modes. Substitution of both streamfunctions into expressions 4.64 and 4.65 gives two equations with an unknown pressure perturbation $\tilde{p}$, which can be eliminated by differentiating the axial momentum equation 4.65 to r. Subtracting the two expressions then yields the axisymmetric equivalent of the Orr-Sommerfeld equation in terms of ϕ:

$$\begin{aligned}\frac{\nu}{i\alpha}\left(\phi'''' + \frac{2\phi'''}{r}\right) + \left[U + \frac{\nu}{i\alpha}\left(2\alpha^2 - \frac{3}{r^2}\right)\right]\left(\phi'' - \frac{\phi'}{r}\right) \\ - \left(U\alpha^2 + U'' - \frac{U'}{r} - \nu i\alpha^3\right)\phi = c\left(\phi'' - \frac{\phi'}{r} - \alpha^2\phi\right),\end{aligned} \tag{4.70}$$

where the apostrophe denotes differentiation with respect to the radial coordinate. Just like its Cartesian counterpart, the equation is of fourth order when viscosity is present and second order in the inviscid case.

4.3.3 Boundary conditions

The two Orr-Sommerfeld equations for each phase require a total of eight boundary conditions. These are imposed at the cylinder origin (two), at the interface (four) and at the cylinder wall (two). At the cylinder wall, no-slip conditions are imposed so that both $\tilde{u}_r$ and $\tilde{u}_z$ are zero, which implies:

$$\phi_a(R) = 0, \qquad \phi_a'(R) = 0. \tag{4.71}$$

At the cylinder axis of symmetry, the boundary conditions are more complicated.

Axisymmetry dictates that the radial velocity $\tilde{u}_r$ at $r = 0$ has to be zero; this can be demonstrated by considering a decomposition into Cartesian components:

$$\tilde{u}_r = u_x \cos\theta + u_y \sin\theta. \tag{4.72}$$

For constant non-zero values of the Cartesian components u_x and u_y at $r = 0$, the radial velocity will vary with the angular coordinate θ. This dependence can only be removed if $u_x = 0 = u_y$, which is when $\tilde{u}_r = 0$ identically. There appears to be no condition for the axial velocity $\tilde{u}_z$, however, other then than that its value must be finite. A Dirichlet condition for $\tilde{u}_z$ could be imposed, and for simplicity it is assumed to be homogeneous. In that case, a solution of the form $\phi(r) = rf(r)$ could be constructed. From the homogeneous Dirichlet condition for $\tilde{u}_r$ it follows that $\phi/r \to 0$ as $r \to 0$, which implies that $f(0) = 0$. For $\tilde{u}_z$, the condition is $\phi'/r \to 0$, which becomes:

$$\lim_{r\to 0} \frac{\phi'}{r} = \lim_{r\to 0} \frac{(rf)'}{r} = \lim_{r\to 0} \left(f' + \frac{f}{r} \right). \tag{4.73}$$

Using l'Hopitâl's rule and the condition $f(0) = 0$, it follows that $f'(0) = 0$. It must be noted that during our computations, this approach was not required as it turned out that setting $\phi_c(0) = 0$ and $\phi_c'(0) = 0$ gave solutions with sufficient decay near $r = 0$.

At the interface at $r = \frac{R}{2} + \eta$, continuity of both velocity and stress is required in the viscous case (we do not consider inviscid flows). The interface conditions are linearized around $r = \frac{R}{2}$ assuming small deviations. Then continuity of normal and tangential velocity implies, respectively:

$$\phi_c = \phi_a, \tag{4.74}$$

$$U_c'\phi_c + c\phi_c' = U_a'\phi_a + c\phi_a', \tag{4.75}$$

at $r = \frac{R}{2}$. Continuity of tangential and normal stress implies, respectively:

$$\gamma \left[c \left(\phi_c'' - \frac{\phi_c'}{r} \right) + (U_c'' + c\alpha^2)\phi_c \right] = c \left(\phi_a'' - \frac{\phi_a'}{r} \right) + (U_a'' + c\alpha^2)\phi_a, \tag{4.76}$$

$$\frac{\gamma}{i\alpha \mathrm{Re}_a} \left[\phi_c''' - \frac{\phi_c''}{r} + \left(\frac{1}{r^2} - 3\alpha^2 \right) \phi_c' - \frac{2\alpha^2 \phi_c}{r} \right] + \gamma \left(c\phi_c' + U_c'\phi_c \right) = \tag{4.77}$$

$$\frac{1}{i\alpha \mathrm{Re}_a} \left[\phi_a''' - \frac{\phi_a''}{r} + \left(\frac{1}{r^2} - 3\alpha^2 \right) \phi_a' - \frac{2\alpha^2 \phi_a}{r} \right] + c\phi_a' + U_a'\phi_a + \frac{\alpha^2 \phi_a}{c\mathrm{We}_a},$$

at $r = \frac{R}{2}$. Notice that equation 4.77, which represents the continuity of the normal stress, is non-linear in the parameter c due to the presence of surface tension.

4.3.4 Solving the generalized eigenvalue problem

Equation 4.70 with its given boundary conditions is solved with a Chebyshev collocation method. First, the core and annular domains $[0, R/2]$ and $[R/2, R]$ are mapped to the interval $[-1, 1]$ using the following coordinate transformations:

$$x : [0, R/2] \to [-1, 1], \quad x = \frac{4r}{R} - 1 \quad \text{for the core,} \tag{4.78}$$

$$x : [R/2, R] \to [-1, 1], \quad x = -\frac{4r}{R} + 3 \quad \text{for the annulus.} \tag{4.79}$$

For both domains $x = 1$ now represents the interface and $x = -1$ represents the cylinder core and outer wall. Both functions ϕ_c and ϕ_a are discretely approximated by a truncated Chebyshev series, i.e:

$$\phi_c = \sum_{i=0}^{N} \alpha_i T_i, \qquad \phi_a = \sum_{i=0}^{M} \beta_i T_i, \tag{4.80}$$

where T_i is the Chebyshev polynomial of the first kind of order i. In our case, we choose $N = M$. The computational domain $[-1, 1]$ is discretized using $N - 3$ collocation nodes $\{x_i\}$ which are defined as the location of the internal extrema of the Chebyshev function T_{N-2}:

$$x_i = \cos\left(\frac{i\pi}{N-2}\right), \qquad i = 1, \ldots, N-3, \tag{4.81}$$

with the aim of obtaining spectral convergence. This gives $2N - 6$ equations on the internal nodes, together with 4 boundary conditions at $x = -1$ and 4 boundary conditions at $x = 1$ for a total of $2N + 2$ equations for the set of $2N + 2$ unknowns $\{\alpha_i, \beta_i\}$. The non-linear boundary condition 4.77 is linearized using equations 4.75 and 4.76 following Boomkamp et al. [11]. This results in:

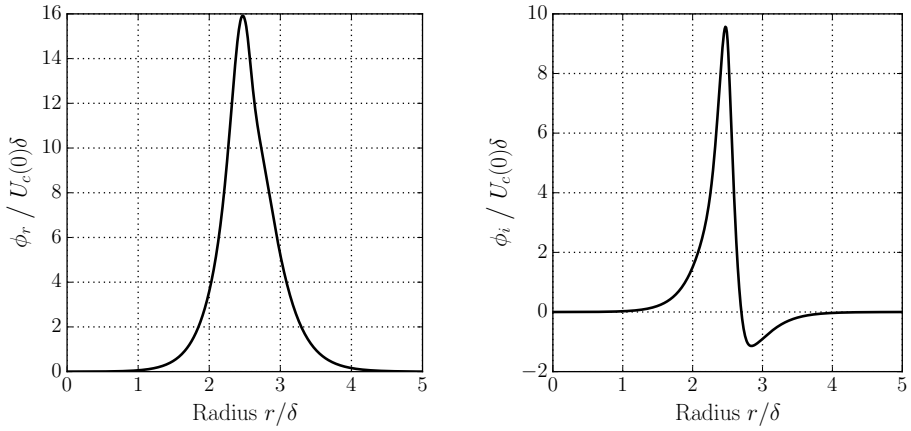
$$\begin{aligned} &\frac{\gamma}{i\alpha \mathrm{Re}_a}\left[\phi_c''' - \frac{\phi_c''}{r} + \left(\frac{1}{r^2} - 3\alpha^2\right)\phi_c' - \frac{2\alpha^2\phi_c}{r}\right] + \gamma\left(c\phi_c' + U_c'\phi_c\right) = \\ &\frac{1}{i\alpha \mathrm{Re}_a}\left[\phi_a''' - \frac{\phi_a''}{r} + \left(\frac{1}{r^2} - 3\alpha^2\right)\phi_a' - \frac{2\alpha^2\phi_a}{r}\right] + c\phi_a' + U_a'\phi_a + \frac{i\alpha^3}{\mathrm{We}_a}\frac{\phi_c' - \phi_a'}{U_c' - U_a'}, \end{aligned} \tag{4.82}$$

The approximations in expression 4.80 are substituted in equation 4.70 which results

in a complex generalized eigenvalue problem for the growth factor c:

$$\mathbf{A}\mathbf{x} = c\mathbf{B}\mathbf{x}, \qquad \mathbf{x} = \Big(\alpha_0, \quad \ldots, \quad \alpha_N, \quad \beta_0, \quad \ldots, \quad \beta_N\Big)^T \tag{4.83}$$

Due to the rows of zeros that arise from the homogeneous boundary conditions at $x = -1$, the matrix $\mathbf{B}$ is singular. The implementation of the QZ algorithm with balancing in Matlab is used to solve the system in equation 4.83. With the coefficients $\{\alpha_i, \beta_i\}$ now known, the two series of expression 4.80 are concatenated to form the complete eigenfunction ϕ over the whole domain $[0, R]$. Figure 4.9 shows the non-dimensional real and imaginary parts of a typical resulting eigenfunction for the case where $\delta/R = 0.2$.



(a) Real part of the eigenfunction ϕ_r.

(b) Imaginary part of the eigenfunction ϕ_i.

Figure 4.9: Eigenfunction solution for case C with $\delta/R = 0.2$.

The non-trivial solutions then turn equation 4.70 into a dispersion relation that returns a vector of growth factors αc_i and an associated set of eigenvectors for a given real wave number α.

4.3.5 Comparison of growth rates

For our simulation purposes, we assume a small perturbation ϵ (with fixed value 10^{-4}) to the streamfunction Ψ. Associating the eigenfunction ϕ to the largest eigenvalue known, the initial perturbed velocity field $(\tilde{u}_r, \tilde{u}_z)$ can be determined using

equation 4.67 through:

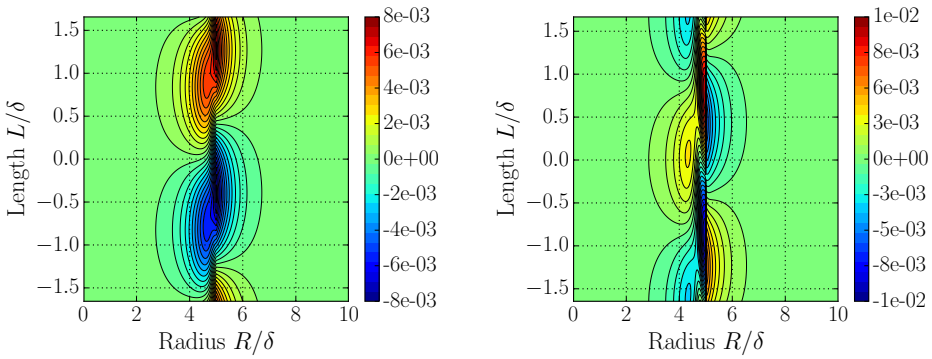
$$\tilde{u}_r(r,z) = \Re\left(-\frac{1}{r}\frac{\partial \Psi}{\partial z}\right), \qquad \tilde{u}_z(r,z) = \Re\left(\frac{1}{r}\frac{\partial \Psi}{\partial r}\right), \tag{4.84}$$

where $\Re$ denotes the real parts of the expression. Writing the complex variables c and ϕ in complex form as $c_r + ic_i$ and $\phi_r + i\phi_i$, these expressions can be expanded to:

$$\tilde{u}_r(r,z) = \frac{\epsilon\alpha}{r}\left[\phi_r(r)\sin\alpha z + \phi_i(r)\cos\alpha z\right], \tag{4.85}$$

$$\tilde{u}_z(r,z) = \frac{\epsilon}{r}\left[\phi_r'(r)\cos\alpha z - \phi_i'(r)\sin\alpha z\right]. \tag{4.86}$$

The complete initial flow field then becomes $(\tilde{u}_r, U + \tilde{u}_z)$. Figure 4.10 shows a typical initial flow field based on a perturbed streamfunction.



(a) Perturbed radial velocity $\tilde{u}_r$.

(b) Perturbed axial velocity $\tilde{u}_z$.

Figure 4.10: Initial velocity field due to a streamfunction perturbation.

In a similar way, the initial interface perturbation $\eta(0,z)$ is determined as:

$$\eta(0,z) = \frac{2\epsilon}{R(c_r^2 + c_i^2)}\left[(c_i\phi_i + c_r\phi_r)\cos\alpha z + (c_i\phi_r - c_r\phi_i)\sin\alpha z\right], \tag{4.87}$$

where the components of the eigenfunction ϕ are evaluated at the interface at $r = R/2$. The wave length of the sinusoidal interface around $r = R/2$ is exactly $2\pi/\alpha$. Both the perturbed flow field and interface are used as initial conditions for our algorithm. Since the flow field is not exactly divergence-free in the discretization, a projection is performed by the Poisson solver. This will cause small changes to the initial analytical velocities, but the effect appears to be negligible.

We consider four cases with two different base flows and parameters based on the simulations of Bagué et al. [4]. For cases A and B, the core and annular densities are equal, while the viscosity ratio is 0.1. For cases C and D, the viscosities are almost equal ($\gamma = 0.99$), but the density ratio is 0.1. The cases B and D are equal to A and C respectively but with a small amount of surface tension added. Table 4.2 summarizes all relevant parameters for all cases.

Table 4.2: Parameters for the different cases.

Case	γ	ζ	Re_c	Re_a	We_c	We_a
A	1	0.1	10000	1000	∞	∞
B	1	0.1	10000	1000	5	5
C	0.1	0.99	100	990	∞	∞
D	0.1	0.99	100	990	1.96	19.6

The axisymmetric domain $R \times L$ used for the simulation has dimension 1 m $\times$3 m, and it is discretized with 160×480 computational cells. The time step varies between 10^{-5} and 10^{-4} s to keep the CFL number more or less constant in all runs. To be able to simulate different wave numbers with a constant cell resolution, the boundary layer thickness δ is varied to give a range of dimensionless wave numbers $\alpha\delta$. The resulting dimensionless growth factor $\alpha c_i \delta / U_c(0)$ is then monitored during the simulation by a least squares fit of a sinusoidal wave $r = R/2 + a\sin(2\pi z/L + b)$. The growth factor is determined by fitting a linear function to the log-plot of the amplitude a versus time.

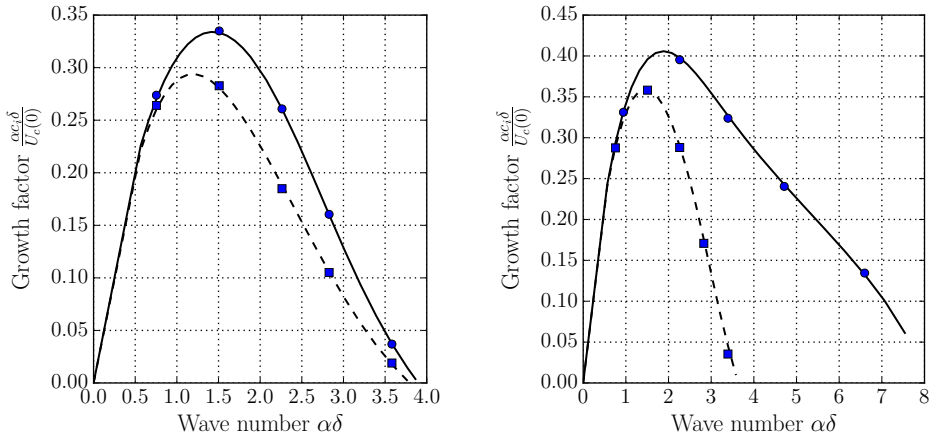
It is generally difficult to determine the end of the exponential regime; the transition is often gradual, and a firm criterion is thus likely to be subjective. To create consistency in our results, we monitor the quality of the least squares fits at every time step. First, the R^2 value of the sinusoidal fit of the interface in space is determined as:

$$R^2 = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \overline{y})^2}, \tag{4.88}$$

where $\{y_i\}$ are the interface locations from the simulation and $\{f_i\}$ are the associated approximations, and the bar denotes the average over the sample set. Furthermore, we monitor the R^2 value of the linear approximation to the logarithm of the amplitude in time using an expression as suggested by Owkes and Desjardins [68]:

$$R^2 = \frac{\left(\sum_{i=1}^{T}\left[(t_i - \bar{t})(\ln(a_i) - \ln(\bar{a})\right]\right)^2}{\sum_{i=1}^{T}\left(t_i - \bar{t}\right)^2 \sum_{i=1}^{T}\left(\ln(a_i) - \ln(\bar{a})\right)^2}, \tag{4.89}$$

with $\{t_i\}$ the time instances until $t = T$ and $\{a_i\}$ the amplitudes from the sinusoidal fit. For all test cases, we fix two (different) values close to unity, and as soon as one of the two R^2 values drops below this value, the evaluations are stopped as it is assumed that the flow is no longer in the exponential flow regime.



(a) Case A and B.

(b) Case C and D.

Figure 4.11: Growth rate according to the Orr-Sommerfeld theory (lines) versus numerical simulations (symbols) with surface tension (dashed line) and without surface tension (solid line).

The results for the cases in table 4.2 are shown in figure 4.11. The curves represent the growth rate of the most unstable mode derived from the Orr-Sommerfeld theory with and without surface tension. The stabilizing effect of the surface tension is apparent from the consistently lower growth factors for a given wave number. Every case has been performed five times for varying wave numbers throughout the unstable spectrum. An excellent agreement between the simulations and the Orr-Sommerfeld theory is found, with relative errors that are less than 1% on average. The relative deviation tends to grow slightly when the wave numbers approach the critical value α_c where the growth factor c_i becomes zero, but the overall agreement throughout the unstable regime of wave numbers is satisfactory. We can conclude that modal stability analysis can be successfully applied to axisymmetric shear flows, and the accurate simulation results gives confidence in the quality of the interface algorithm and its ability to capture Kelvin-Helmholtz instabilities.

4.4 Stratified shear flow in a periodic pipe section

This section serves as a prelude to section 4.5, where a large closed pipe experiment will be simulated. To prevent any influence of the end walls, we consider a small section of the pipe under periodic conditions to be able to monitor the wave development for a predefined shear flow and initial perturbation of the interface.

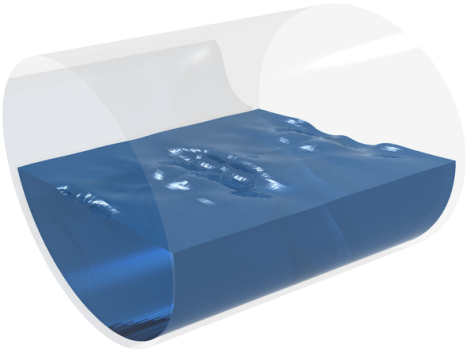
To study the effects of a stratified shear flow in a controlled environment, we simulate the behaviour of two layers of fluid in a periodic pipe section of radius $R = 0.025$ m and length $L = 2R$. The lower heavier phase consists of water and the upper lighter phase consists of oil. The velocity jump at the interface is initially discontinuous, i.e.:

$$U_z(r,\theta) = \begin{cases} U, & y \geq h, \\ -U, & y < h, \end{cases} \tag{4.90}$$

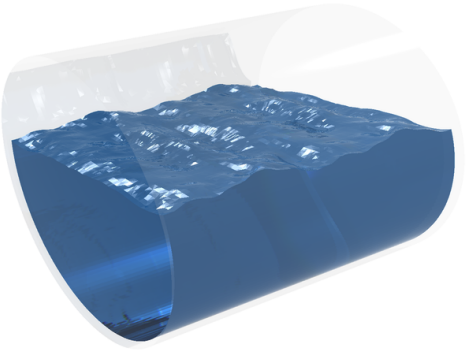
where $y = r\sin\theta$ and $-R \leq h \leq R$. h is a predefined height that depends on the water volume fraction. Two volume fractions are considered, $V = \frac{1}{2}$ and $V = \frac{2}{3}$. The former is computationally challenging as it involves the movement of the interface through and around the origin. The latter in turn is used to study the influence of the curved walls on the flow. An initial perturbation of the form $y = h + \epsilon R\cos(4\pi z/L)$ is applied so that the domain covers two full wave lengths and the amplitude of the perturbation is ϵR. The perturbation factor ϵ is fixed at 0.01. The inviscid Kelvin-Helmholtz theory in a channel of height R provides a condition for the onset of the instability, namely when the velocity difference $2U$ satisfies [24]:

$$(2U)^2 > \frac{\rho_w + \rho_o}{\rho_w \rho_o}\left(\frac{g(\rho_w - \rho_o)}{k} + k\sigma\right)\tanh(kR). \tag{4.91}$$

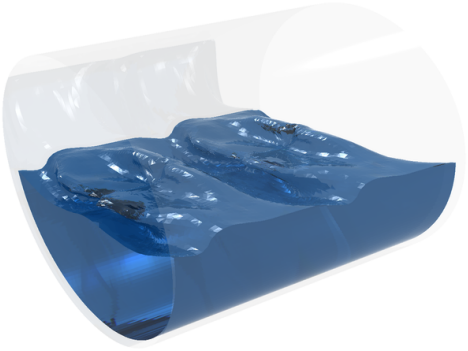
When applied to the parameters given above, this yields $2U > 0.171$ m/s. In our simulation we have fixed the velocity U at 0.25 m/s to guarantee unstable flow conditions. A grid of $30 \times 150 \times 60$ cells is used with a time step of $\Delta t = 10^{-5}$ s to satisfy the time step constraint due to surface tension. No-slip boundary conditions are applied at the cylinder walls and periodicity is assumed in axial direction.
Figure 4.12 shows the water fraction and the interface at different times for both $V = \frac{1}{2}$ (left column) and $V = \frac{2}{3}$ (right column). Profound discrepancies can be seen between the development of the waves for both cases. The initial states (upper graphs in figure 4.12) show a much smoother interface for $V = \frac{1}{2}$ than for $V = \frac{2}{3}$. The in-plane analysis shows that this is due to cross flow induced by the curved



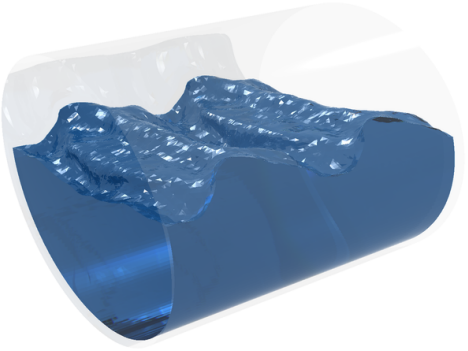
(a) Initial waves at $t = 3 \cdot 10^{-4}$ s for $V = \frac{1}{2}$.



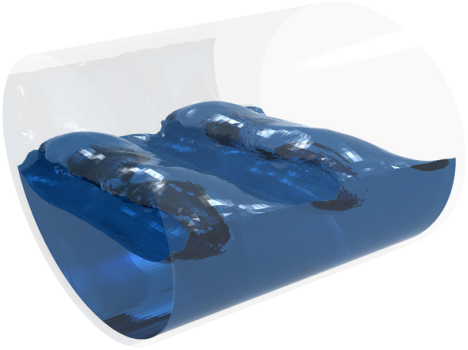
(b) Initial waves at $t = 4 \cdot 10^{-4}$ s for $V = \frac{2}{3}$.



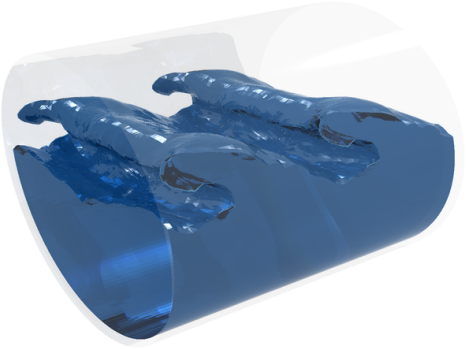
(c) Wave build up at $t = 6 \cdot 10^{-4}$ s for $V = \frac{1}{2}$.



(d) Wave build up at $t = 6 \cdot 10^{-4}$ s for $V = \frac{2}{3}$.



(e) Roll waves at $t = 1.05 \cdot 10^{-3}$ s for $V = \frac{1}{2}$.



(f) Roll waves at $t = 8.5 \cdot 10^{-4}$ s for $V = \frac{2}{3}$.

Figure 4.12: Evolution of an initially perturbed interface in a shear flow. The left column has water fraction $1/2$ and the right column has $2/3$. The grid size is $30 \times 150 \times 60$.

walls in the latter case, while nearly no cross flow exists in the former. Also, the figure shows that the instability first occurs in the centre for $V = \frac{1}{2}$, while slightly

later the wave is initiated nearly uniformly across the pipe for $V = \frac{2}{3}$. At a slightly later instant, as shown in the middle graphs of figure 4.12, waves have grown, and a clear difference in the amplitudes for the waves for $V = \frac{1}{2}$ and $V = \frac{2}{3}$ is observed. For $V = \frac{1}{2}$ there is a tendency for the wave to start rolling. The vast movement of the interface around the origin is a challenge for the interface algorithm, which becomes clear in the left middle graph of figure 4.12, where a part of the interface has an nonphysical shape in the front wave (although the rear wave looks unaffected). Finally, for a later time as shown in the lower graphs of figure 4.12, the waves start rolling and the initial wave length is regained.

4.5 Stratified shear flow in a closed inclined pipe

To validate the simulation results, a small experimental study was recently carried out in the fluid flow lab of our university. Some dedicated experiments have been performed based on the setup of Thorpe [88], including both a rectangular channel and a cylindrical pipe. The Kelvin-Helmholtz instability arising from the shear flow within the pipe was observed. A pipe length of 1 m with closed end walls was used. At the start of the experiment, the pipe is tilted from horizontal to a pre-defined inclination, which varied between 2°, 5°, 10° and 15°. The fluids used were water and oil with the following material properties:

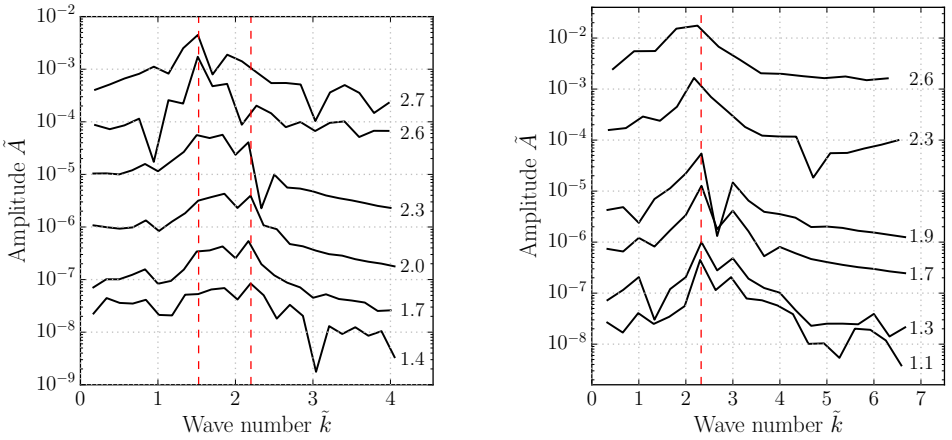
$$\begin{aligned} \rho_w &= 9.97 \cdot 10^2 \ \text{kg/m}^3, & \rho_o &= 8.74 \cdot 10^2 \ \text{kg/m}^3 \\ \mu_w &= 8.90 \cdot 10^{-4} \ \text{kg/m/s}, & \mu_o &= 4.37 \cdot 10^{-3} \ \text{kg/m/s} \end{aligned}$$

The surface tension is about $\sigma = 0.035$ kg/s^2. The experiments were done with a water fraction V of 1/3, 1/2 and 2/3. In the simulations, we only considered an inclination angle of 5°, and the simulations were performed both with and without surface tension.

4.5.1 Results without surface tension

We start by performing a simulation without surface tension to observe how the interface perturbation grows under the influence of gravity and viscosity alone. From equation 4.91 it follows that in this case $k \sim 1/\Delta U$, and hence inviscid theory predicts the presence of increasing wave lengths as the velocity difference ΔU increases. Viscosity, however, is known to enforce a lower bound on the wave lengths due to its damping effect. Simulations are performed on two grids, and a Fast Fourier

Transform (FFT) analysis is done on a fixed segment of the interface at different time instances. The segment is chosen such that sufficient resolution for the FFT exists. Figure 4.13 shows the truncated results for both grids on a log scale. Using the cylinder radius R, the wave number k and the interface amplitude A are made dimensionless as $\tilde{k}$ and $\tilde{A}$, respectively.



(a) Results from the $20 \times 100 \times 500$ grid. The red lines are located at $\tilde{k} = 1.53$ and $\tilde{k} = 2.2$.

(b) Results from the $26 \times 120 \times 1000$ grid. The red line is located at $\tilde{k} = 2.3$.

Figure 4.13: FFT of the interface at different times (in seconds) in the centre region of the pipe without surface tension.

At early times, both grids seem to contain a dominating wave number $\tilde{k}$ close to 2.3. This value can therefore be considered as the cut-off wave number due to viscosity. When the velocity difference grows with increasing time, there is a shift of the leading wave number which is most clearly noticeable in figure 4.13a. The amplitude around $\tilde{k} = 1.53$ is growing slowly and becomes dominant at $t = 2.3$ s. A similar, but less pronounced shift is observed as well in figure 4.13b. The dominant part of the spectrum at $t = 2.6$ s becomes wider without strong peaks, implying the presence of multiple smaller wave lengths. The shift in the dominant wave length for increasing velocity difference is also observed by Bartosiewicz et al. [9] in their 2D simulation results of the Thorpe experiment.

To demonstrate the damping effect of the viscosity, the simulation on the finest grid with $26 \times 120 \times 1000$ cells is repeated with inviscid fluids. In this case, $k \sim 1/\Delta U$ should hold exactly without viscous damping, and larger wave numbers are expected compared to the viscous case. The resulting FFT analysis at a fixed time, together with the viscous result, is shown in figure 4.14. The inviscid spectrum is clearly much wider than the viscous spectrum, with peaks up to $\tilde{k} = 9$. Note that

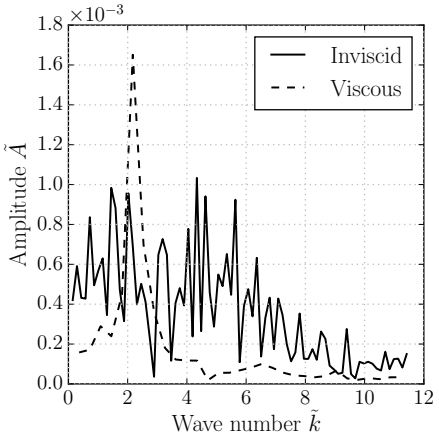


Figure 4.14: Truncated FFT results from the $26 \times 120 \times 1000$ grid at $t = 1.4$ s.

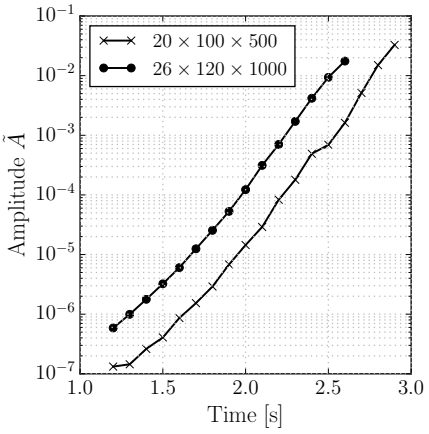


Figure 4.15: Amplitude growth of the initial perturbation resulting from the FFT analysis.

near this wave number, the computational grid size used gives only about three cells per wave length, and hence the observed upper limit is likely to result from the choice of the grid size.

Finally, the growth of the perturbation amplitude is considered. The inviscid theory predicts that any unstable wave number initially grows exponentially in time until non-linear effects become dominant. To that end, the wave amplitudes that result from the viscous FFT analyses are plotted over time on a log scale for both grids in figure 4.15. In particular for the results on the finer grid, a region of exponential growth is indeed found.

4.5.2 Results with surface tension

The simulations of the previous section are repeated, but now with a non-zero surface tension of $\sigma = 0.035$ kg/s^2. According to the inviscid theory, the presence of surface tension suppresses small wave lengths. The analysis of the right hand side of equation 4.91 shows at least two observations. Firstly, a critical velocity difference exists, below which all perturbations are damped. Secondly, for a sufficiently large velocity difference, there exists a bounded spectrum of unstable wave numbers. A wave length of about 3.4 cm was measured in the lab for a pipe at 5° inclination. Furthermore, it was noted during the lab experiments that the wave front was not necessary uniform throughout the cross section of the pipe. For the simulated

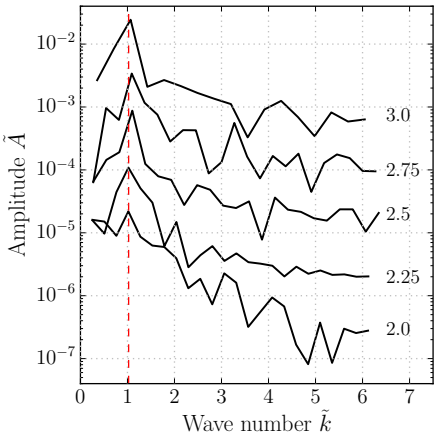


Figure 4.16: Truncated FFT result. The red line is at $\tilde{k} = 1$.

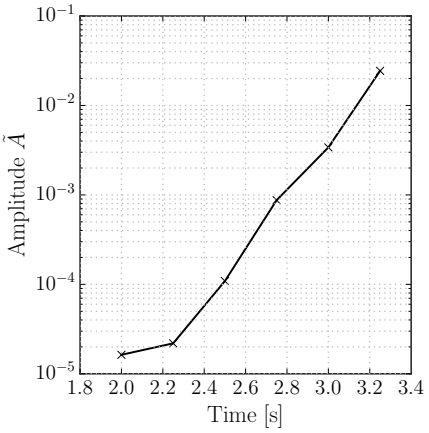


Figure 4.17: Amplitude growth of the initial perturbation resulting from the FFT analysis.

results, an FFT analysis similar to the one described in the previous section is performed. Figure 4.16 shows a clear peak at around $\tilde{k} = 1$, corresponding to a wave length of around 2.5 cm, which is somewhat shorter than the experimental value of 3.4 cm. Similar to the case without surface tension, the perturbation grows nearly exponentially, as shown in figure 4.17. Finally, figure 4.18 shows the interface at time $t = 3.25$ s, which is just before roll waves appear.

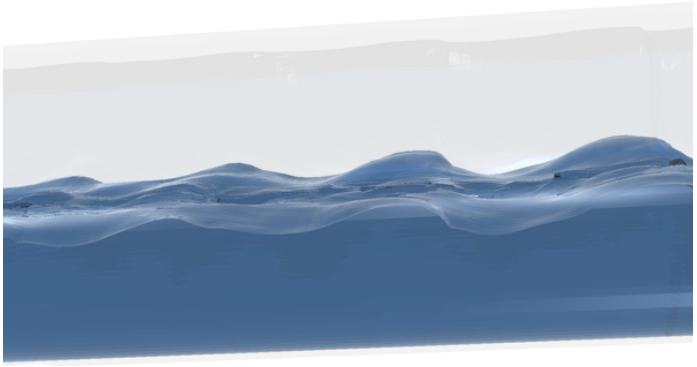


Figure 4.18: Simulated interface near the midsection of the pipe.

A notable issue that we encountered is related to the boundary condition of the level set function. In Cartesian domains often a homogeneous Neumann condition for the level set function is used at the walls of the domain. In the case of 2D stratified two-phase flows, this seems to be a reasonable approximation. In cylindrical coordinates, however, the interface becomes excessively curved near the walls due

to the surface tension force. An example is shown in figure 4.19, where the solid line represents the interface in a cross section of the pipe.

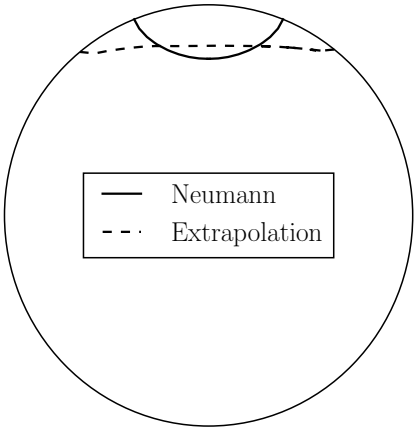


Figure 4.19: Cross-section of the pipe at $z = -0.45$ m with the interface for both homogeneous Neumann and extrapolation boundary conditions.

As the upper area becomes even smaller, the condition leads to numerical stability problems as regions with extreme curvature are present. To remedy this, wettability should be included in the boundary conditions, which will likely result in a more realistic behaviour of the interface near the walls. A simple alternative is to apply linear extrapolation of the level set function at the boundary (notice that the homogeneous Neumann condition is in fact a zero order extrapolation). Figure 4.19 shows the interface of the same simulation with linear extrapolation of the boundary conditions, and the result looks much more physical compared to applying the Neumann condition. A potential drawback that we encountered with the extrapolation method is the slightly less stable level set reinitialization when this is performed close to the wall. Since wettability is not yet included in our algorithm, we have chosen to apply the extrapolation boundary condition for the level set function.

4.5.3 Computational performance

The algorithm uses domain decomposition in the axial direction and the MPI protocol for parallel computations. Preconditioning is applied to the solving of both the momentum equations and the Poisson equation. A simple but effective Jacobi preconditioner is used for the linearized momentum equations, and the Conjugate Gradient method uses block Incomplete Cholesky preconditioning with Subdomain

Deflation. A strong scaling test is performed using the closed cylinder setup of section 4.5, with a grid size of $20 \times 100 \times 500$ computational cells. A total of 1000 iterations are performed to eliminate initialization influences, and the resulting measured wall clock time is shown in figure 4.20.

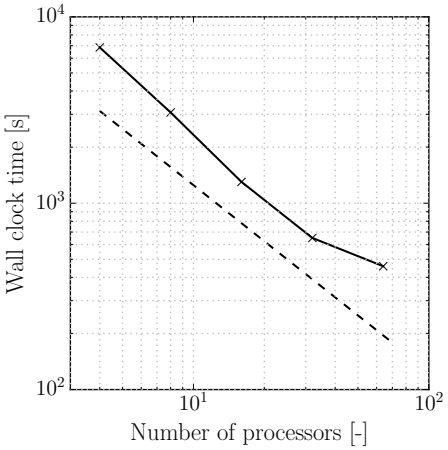


Figure 4.20: Strong scaling results of the algorithm implementation.

Performance profiling for different grid sizes shows that the interface representation algorithm is very light (less than 5% of CPU time in a typical run) when compared to the cost of the flow solver. This can be expected when considering the complexity of the fully implicit flow solver versus the explicit interface advection. Our algorithm is able to handle CFL numbers up to 0.5, which allows the use of significantly larger time steps than in commonly used compressive VoF methods. The efficiency of the method is further improved through the use of interface subcycling, where multiple interface advection steps can be performed within a single flow time step, thereby fully utilizing the unconditional stability of the flow solver in case of CFL numbers larger than 0.5.

4.6 Conclusions

A sharp interface model for cylindrical coordinates was designed, tested, and validated. By using structured orthogonal coordinates, the aim is to obtain better accuracy and efficiency compared to commonly used unstructured methods to allow LES or DNS for turbulent multiphase flows at reasonable computational costs. The method is based on an efficient combination of the LS method and the VoF

method, and it was validated using two test cases. The simulations show an excellent behaviour of the interface algorithm for a controlled perturbation in core-annular flow. The agreement of the simulations with experiments for the onset of Kelvin-Helmholtz waves in an inclined pipe section is reasonable.

In the absence of similar methods to compare with, we acknowledge that the algorithm presented is a first step towards the efficient use of cylindrical coordinates in multiphase pipe flows. Their use can be particularly troublesome in the discretization of both the flow field and the interface. In Oud et al. [67], the former problem was dealt with for single phase flows, and the difficulty in the extension to multiphase flows lies in the interface algorithm. A complete implementation of the Ghost Fluid method [51], including both the jump in pressure and its gradient, would allow the use of the flow discretization of Oud et al. [67] in our algorithm. The implementation of the pressure gradient jump is challenging, but it is recommended to explore this in future work.

The behaviour of the VoF function near the polar origin needs further investigation. A detailed analysis was performed, and it was concluded that the cause of the encountered numerical oscillations lies in the time-explicit advection of the VoF field. An improvement was derived in the form of a VoF redistribution algorithm, but further research into alternative discretization techniques might be considered to improve the quality of the interface representation when it crosses the origin. An implicit treatment of the VoF fluxes appears an attractive alternative, because it allows the interface to cross multiple cells in a single time step without CFL constraint. It is strongly recommended to explore this implicit approach in future work.

Acknowledgments

We would like to thank SURFsara for the support in using the Lisa Computing Cluster. Roel Noordman is acknowledged for carrying out the lab experiments as referred to in this chapter.

Chapter 5

Recommended further developments

The objective of the study described in the first chapter of this thesis was to obtain an improved numerical tool that is able to carry out relatively fast, robust, and accurate simulations of physical, two-phase flow instabilities in straight pipe sections. Thereto the Navier-Stokes equations and the Laplace-Young equation for the interface were discretized on cylindrical coordinates for a straight pipe section. A second-order accuracy of the discretization could be obtained by using a mimetic approach. The Mass-Conserving Level Set method, as was originally developed for interfaces on a Cartesian grid, was extended to cylindrical coordinates. In the numerical analysis, much attention was given to handling the singularity in the grid at the centreline. The new numerical tool was verified and validated against a number of two-phase flow test cases, with increasing complexity.

The most complex test case considered so far with our method was the onset and growth of Kelvin-Helmholtz waves in a finite pipe section (with closed end walls) that was tilted to a small angle, starting from a full horizontal position. The pipe was filled with two liquids with a different density and viscosity. Experiments for this case, as obtained at Delft University, exist, and they include the onset and growth of the waves, but also the damping, until a new steady state with the two layers of liquid is obtained for the new tilted position. The wave damping was not yet simulated.

Up to now, all considered transient effects occur in laminar two-phase base flows. The real check of whether the new tool is indeed fast, robust, and accurate for problems of industrial interest requires considering cases with a higher Reynolds

number, which will give a turbulent base flow.

Although we obtained significant new insights in the numerics of two-phase flow, there is plenty of room for further investigation. The recommended areas of research are described in the subsequent subsections of this chapter.

Feasibility of high Reynolds number computations

The derivation of the kinetic energy conservation in section 2.4.3 appears to hinge on the use of the Implicit Midpoint time integration method. As mentioned, the choice of the time integration method is as important as the choice of the spatial discretization method if exact conservation is desired. However, the implicit nature of the method used and the fact that conservation requires the solution of the non-linear equations lead to a computationally expensive discretization. Of course, it is completely up to the objectives of the user whether this effort merits the computational costs, but it warrants an investigation into the possibilities of cheaper (perhaps explicit) time integration methods with similar or nearly similar conservation properties, such as in the work of Sanderse [76] and Verstappen and Veldman [91].

For LES or DNS, it is likely that the overall second order accuracy of the proposed Navier-Stokes discretization is too restrictive. Further research should therefore entail the extension to higher order discretizations, while maintaining the conservation properties of the scheme. The extension needs to concern at least two important aspects. First, higher order approximations of the discrete vector operators and inner products as defined in section 2.2 should be established. Except for the cells near the origin, straightforward application of higher order differences seems sufficient for this. The difficulty will likely be the second aspect, namely the finding of a consistent averaging procedure for the discrete convective terms, which governs both the momentum and kinetic energy conservation. Alternatively, obtaining higher order of approximations based on Richardson extrapolation can be explored along the lines of Verstappen and Veldman [91], but the implications on the conservation properties (especially on non-uniform grids) should be thoroughly monitored.

Conservation of helicity

Besides mass, momentum and kinetic energy, a scalar quantity known as *helicity* is a conserved quantity in inviscid 3D flows. It is defined as:

$$\mathcal{H} = \int_\Omega \mathbf{u} \cdot (\nabla \times \mathbf{u}) \, \mathrm{d}V = \int_\Omega \mathbf{u} \cdot \boldsymbol{\omega} \, \mathrm{d}V, \tag{5.1}$$

with $\mathbf{u} \in \mathbb{R}^3$ the flow velocity and $\boldsymbol{\omega} \in \mathbb{R}^3$ the flow vorticity. The importance of helicity to the vortical structures of turbulence was recognized by Moffatt and Tsinober [59] a few decades ago. Despite its physical relevance, numerical discretization schemes of the incompressible Navier-Stokes equations designed to conserve (among others) helicity are scarce, but developments do occur; see for example the work of Liu and Wang [54] and Rebholz [70]. It would be interesting to investigate to what extent the proposed Navier-Stokes discretization is capable of conserving helicity as an additional conserved quantity. One of the first tasks will be to find a discrete analogue to the definition of helicity in equation 5.1, since velocity and vorticity are defined at different locations in the staggered grid. This problem is similar to the discretization of the product occurring in the convective terms of the discrete Navier-Stokes equations (see section 2.3.1). The simultaneous conservation of multiple quantities is not easy for numerical schemes: like the conservation of kinetic energy, also the conservation of helicity follows from manipulation of the momentum equations. A successful demonstration of this would make a very positive contribution to the field of turbulent pipe flow simulations.

Extension to variable-viscosity flows

The applicability of the proposed Navier-Stokes discretization in cylindrical coordinates for pipes is currently limited to flows of constant viscosity. This is due to the fact that the development of mimetic discretization techniques was originally focused on first order tensors, i.e. the gradient of a scalar or the curl of a vector. For non-constant viscosity flows, the viscous part of the Navier-Stokes equations consists of a second-order tensor that is more complex to handle in the context of mimetic finite differences. Expanding the field of mimetic approximations to higher order tensors is currently an active field of research (see for example Campbell and Shashkov [16], Campbell et al. [17], da Veiga and Manzini [19]), and it is likely that an extension to variable-viscosity flows lies within the realm of possibilities.

Accuracy of the interface conditions

For a more consistent treatment of the jump conditions at the interface, the current algorithm needs to be modified in such a way that the interface is modeled as truly sharp. As mentioned in section 3.2.2, a continuous viscosity formulation that is smeared out in a narrow region along the interface is currently used to avoid the complex implementation of one of the interface conditions. A full implementation of the Ghost Fluid method similar to Kang et al. [51], in which both the jump in the pressure as well as the jump in the viscosity are modeled, is more consistent with the assumption of a sharp interface. A secondary benefit is that a full implementation would allow the use of the current constant-viscosity discretization for laminar flows as well as for DNS for turbulent flows. The turbulent viscosity ν_t, when used in LES, is not constant in space, which complicates the application of the proposed discretization for the simulation of turbulent flows. Once correctly implemented, the conservation properties of the multiphase scheme should be investigated. Already challenging in single phase flows, the conservation of momentum and kinetic energy in numerical schemes for multiphase flows is even more difficult to obtain. Rigorous analyses of this subject in the literature are rare, and more research into these aspects is required.

Implicit VoF advection

It was shown in section 4.2.5 that explicit finite volume-based VoF advection is highly inaccurate whenever the interface crosses the origin of the polar plane. This is caused by the inability of explicit advection to allow fluid to traverse multiple cells within a single time step. A simple VoF redistribution algorithm was devised to improve the accuracy of the interface near the origin for the intended test cases, but a more generalized solution is required. Implicit treatment of the VoF fluxes does not suffer from the prohibitive CFL constraint of explicit methods, and this is likely to significantly improve the accuracy of the VoF advection near the origin. The complexity of its implementation and the computational cost are strong drawbacks of such an approach, but it is likely that the availability of the level set function may render an implicit treatment feasible. The key observation is that the level set function at the new time level is generally a very good approximation of the exact new interface. When this is used in combination with iterative solvers, it provides an accurate first guess so that the number of iterations can be kept small. In this way, the computational cost can hopefully remain at an acceptable level, but only practical experience will ultimately show the realism of this expectation.

Extension to the Moment of Fluid method

Recent developments in the field of interface capturing methods include the Moment of Fluid (MoF) method [26, 27], which is a generalization of the VoF method by including higher order moments of the colour function in the process of the interface reconstruction. Having available both the fluid quantity and its centroid information in every cell, the reconstruction of a linear interface is formally uniquely defined. An important observation is that this reconstruction can be done cell-wise without any interaction with neighbouring cells, which makes (fine-grained) parallelization of the process relatively simple. Although the MoF method formally renders the level set function redundant in terms of determinacy, the latter can still be used to efficiently simplify various processes in the MoF algorithm. The relatively simple implementation and low computational demands of the level set method then make it a feasible addition nonetheless. Jemison et al. [47] have developed a coupled level set - MoF method for incompressible two-phase flow problems on Cartesian grids with impressive results. The use of the MoF method as an extension to the current algorithm should therefore be considered. The first steps towards such an extension have already been taken by applying the numerical VoF calculation of section 4.2.3 to the calculation of the first moments of the colour function. Additionally, an advection scheme for the fluid moments has been derived.

Parallel performance

The results of the 3D test case in section 4.5 show good scaling properties of the current algorithm. However, so far we have only carried out a few strong scaling tests to analyze the parallelism. For more definite conclusions on the performance, also weak scaling results should be determined in future research. Furthermore, the coarse-grained parallelism can be improved by extending the domain decomposition to radial and angular domains as well (currently only decomposition in axial direction is performed), depending on the configuration considered in the test case. Fine-grained parallelism through the use of OpenMP is currently not employed, but its implementation should be included in the future optimization of the algorithm.

Domain geometry

The initial choice of a straight cylindrical pipe section was made for efficiency purposes, and the genericity of the computational domain was intentionally sacrificed. For an increased applicability of the algorithm, an extension to include smoothly

curved cylinders to simulate bends in pipes will be valuable. Any attempt, however, should strive to maintain the efficiency of the algorithm as much as possible. The orthogonality of the cylindrical coordinate system, which contributes to the overall efficiency, is generally lost when the cylinder axis is described by a continuous curve in space. As a possible solution, however, the coordinate system derived by Germano [34] is close to cylindrical, and can be used to maintain orthogonality of the flow equations. The adaptation of the interface algorithm then remains. Because of the similarity to cylindrical coordinates, it is expected that the conversion can be performed without adding an excessive amount of complexity.

Extension of the test cases

Further validation of the derived interface algorithm requires an extended set of test cases. Configurations close to real-life hydrocarbon applications are of particular interest, like for example the simulation of a Benjamin bubble in a horizontal pipe (for which also analytical solutions exist) or the comparison to stratified flow (laminar and turbulent) in a pipe with capillary waves and with gravity waves at the interface [10]. Furthermore, the comparison with experiments of hydrodynamic slug flow in a horizontal pipe [98] and the comparison to core-annular flow with a laminar oil core with turbulent water in the annulus [65], known as 'bamboo waves' [52, 71], could be investigated in more detail.

Cylindrical coordinates versus Cartesian coordinates

The use of cylindrical coordinates to model pipe flows was motivated by an anticipated performance gain compared to the use of a Cartesian coordinate systems. To precisely quantify the benefits of using the boundary-fitting cylindrical coordinate system over the Cartesian coordinate system in combination with the Immersed Boundary Method (IBM) or similar boundary-capturing techniques, comparitive simulations need to be performed. An appropriate array of tests is required to derive objective conclusions about the difference in robustness, speed and accuracy of both approaches.

Chapter 6

Conclusions

The present thesis has provided a detailed analysis to numerically solve the Navier-Stokes equations for two-phase flows in a pipe using cylindrical coordinates. The work can be split up in finding a proper discretization of the Navier-Stokes equations for each of the multiple phases, and finding a proper representation of the interface between the phases. As the desire was to use a structured, boundary-fitted grid through cylindrical coordinates (instead of an unstructured boundary-fitted grid or a structured, non-boundary-fitted Cartesian grid), resolving the singularity in the grid at the pipe centre line was a main challenge in this study.

The main conclusions of the Navier-Stokes discretization are the following.

- The applied approach leads to a discretization of the flow velocities and the pressure which was numerically demonstrated to be second order accurate in space.
- The applied approach leads to conservation of mass and radial, angular and axial momentum in cylindrical domains with appropriate boundary conditions. Mass and momentum conservation were numerically demonstrated to hold.
- When using the Implicit Midpoint method (which conserves quadratic invariants) for the time integration of the semi-discrete Navier-Stokes equations, the applied approach leads to a discretization that is capable of conserving kinetic energy for inviscid flows with appropriate boundary conditions. The spatial conservation of kinetic energy for such flows has been proven analytically to hold.
- The combination of the mimetic approximation of the differential vector operators and the averaging procedure of the discrete convective terms of the Navier-Stokes

equations leads to a consistent treatment of the coordinate singularity. In the applied approach, the singularity at the pipe axis neither requires any special in situ modifications of the discretization, nor does it affect the second order accuracy or the conservation properties.

- The second order accuracy and the conservation properties apply for both uniform and non-uniform grids (in radial and axial direction).

The main conclusions of the numerical interface representation are the following.

- A combined Level Set method and Volume of Fluid method, which can be seen as a dual interface approach, has been successfully extended from its earlier use in Cartesian coordinates to cylindrical coordinates.
- The Height Function method used to calculate the interface curvature based on a local stencil of VoF values can be extended to 3D cylindrical domains. Numerical tests show that the resulting approximation is second order accurate in space for an exact VoF field.
- It was verified that second order reinitialization of the level set function leads to a non-convergent approximation of the interface curvature when that curvature is approximated by using the level set function.
- Comparison of the simulation results of the developed dual interface method with results from three general-purpose CFD packages (Fluent, STAR-CCM+ and OpenFOAM) for the rising bubble test cases shows a good agreement, both for the bubble shape and for the terminal rise velocity. There is also good agreement with lab experiments.
- The time-explicit VoF advection becomes highly inaccurate when flow in the polar plane through the origin at $r = 0$ is present. This is caused by angular velocities that become infinite, and the subsequent instantaneous movement of the interface cannot be captured by explicit methods.
- Modal stability theory of two-phase pipe flows can successfully be applied to the development of temporal instabilities in the case of an axisymmetric interface in a core-annular flow. The developed interface algorithm is capable of accurately approximating the predicted growth rate for a given perturbation of the interface.
- Performance tests show that the developed interface algorithm generally takes only $5 - 10\%$ of CPU time during each time step. Furthermore, results from large

test cases with more than 10^6 cells show good strong scaling properties of the algorithm.

This study has led to new software for solving two-phase pipeline problems that are of industrial interest. The robustness, speed and accuracy of the CFD algorithm was demonstrated for a number of test cases. Strengths and weaknesses of the applied numerical methods have been identified. It is clear that more work is needed to prove the performance for more challenging test cases, such as turbulent (rather than laminar) two-phase flow, the interaction of a turbulent and wavy interface, hydrodynamic slug flow, entrainment of gas bubbles in a liquid layer, or entrainment of liquid droplets in a gas layer. In the previous chapter, recommendations were given for further research on the numerical aspects that will make the handling of these more difficult test cases possible. Success means the ability to perform 2D and 3D simulations in a robust, fast, and accurate manner, for real-life engineering two-phase pipe flow conditions. This is clearly within reach, and the demonstrated success has and will help to make multiphase CFD more mature, and bring it to a level of reliability comparable to the level at which we nowadays carry out single phase simulations.

Appendix A

Proofs of mimetic inner products

In this appendix, the proofs that are referred to in section 2.2.4 are provided.

Proposition. *For any $p \in HC$ and $\mathbf{u} \in \mathcal{HS}$ on an infinite domain, it holds that* $(\mathbf{Du}, p)_{HC} = -(\mathbf{u}, \overline{\mathbf{G}}p)_{\mathcal{HS}}$.

Proof. Let $p \in HC$ and $\mathbf{u} = (u_r, u_\theta, u_z) \in \mathcal{HS}$. We start with the inner product $(\mathbf{Du}, p)_{HC}$. For cell $\mathbf{I}$, this is given by:

$$
\begin{aligned}
(\mathbf{Du}, p)_{HC_{\mathbf{I}}} = V_{\mathbf{I}} \Bigg(& \frac{r_{i+\frac{1}{2}} u_{r_{\mathbf{I}+\mathbf{e}_r}} - r_{i-\frac{1}{2}} u_{r_{\mathbf{I}-\mathbf{e}_r}}}{r_i \Delta r_i} + \frac{u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}} - u_{\theta_{\mathbf{I}-\mathbf{e}_\theta}}}{r_i \Delta\theta} \\
& + \frac{u_{z_{\mathbf{I}+\mathbf{e}_z}} - u_{z_{\mathbf{I}-\mathbf{e}_z}}}{\Delta z_k} \Bigg) p_{\mathbf{I}} \qquad \text{(A.1)} \\
= & \Big(\Delta\theta \Delta z_k (r_{i+\frac{1}{2}} u_{r_{\mathbf{I}+\mathbf{e}_r}} - r_{i-\frac{1}{2}} u_{r_{\mathbf{I}-\mathbf{e}_r}}) + \Delta r_i \Delta z_k (u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}} - u_{\theta_{\mathbf{I}-\mathbf{e}_\theta}}) \\
& + r_i \Delta r_i \Delta\theta (u_{z_{\mathbf{I}+\mathbf{e}_z}} - u_{z_{\mathbf{I}-\mathbf{e}_z}}) \Big) p_{\mathbf{I}} \qquad \text{(A.2)}
\end{aligned}
$$

where $V_{\mathbf{I}} = r_i \Delta r_i \Delta\theta \Delta z_k$ is the volume of cell $\mathbf{I}$. Then, for the inner product $(\mathbf{u}, \overline{\mathbf{G}}p)_{\mathcal{HS}}$, we collect all contributions to an arbitrary $p_{\mathbf{I}}$. For an arbitrary cell $\mathbf{I}$:

$$
\begin{aligned}
(\mathbf{u}, \overline{\mathbf{G}}p)_{\mathcal{HS}_{\mathbf{I}}} = \frac{\Delta r_i \Delta\theta \Delta z_k}{2} \Bigg[& r_{i-\frac{1}{2}} u_{r_{\mathbf{I}-\mathbf{e}_r}} \frac{2(p_{\mathbf{I}} - p_{\mathbf{I}-2\,\mathbf{e}_r})}{\Delta r_{i-1} + \Delta r_i} + r_{i+\frac{1}{2}} u_{r_{\mathbf{I}+\mathbf{e}_r}} \frac{2(p_{\mathbf{I}+2\,\mathbf{e}_r} - p_{\mathbf{I}})}{\Delta r_i + \Delta r_{i+1}} \\
& + r_i \left(u_{\theta_{\mathbf{I}-\mathbf{e}_\theta}} \frac{p_{\mathbf{I}} - p_{\mathbf{I}-2\,\mathbf{e}_\theta}}{r_i \Delta\theta} + u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}} \frac{p_{\mathbf{I}+2\,\mathbf{e}_\theta} - p_{\mathbf{I}}}{r_i \Delta\theta} \right)
\end{aligned}
$$

$$+ r_i \left(u_{z_{\mathbf{I}-\mathbf{e}_z}} \frac{2(p_{\mathbf{I}} - p_{\mathbf{I}-2\,\mathbf{e}_z})}{\Delta z_{k-1} + \Delta z_k} + u_{z_{\mathbf{I}+\mathbf{e}_z}} \frac{2(p_{\mathbf{I}+2\,\mathbf{e}_z} - p_{\mathbf{I}})}{\Delta z_k + \Delta z_{k+1}} \right) \Bigg] \tag{A.3}$$

Collecting all radial contributions to $p_{\mathbf{I}}$ gives:

$$-\frac{1}{2}\Bigg[\overbrace{-\Delta r_{i-1}\Delta\theta\Delta z_k \left(\frac{2r_{i-\frac{1}{2}} u_{r_{\mathbf{I}-\mathbf{e}_r}}}{\Delta r_{i-1} + \Delta r_i} \right)}^{\text{from cell } \mathbf{I}-2\,\mathbf{e}_r} \overbrace{+\Delta r_{i+1}\Delta\theta\Delta z_k \left(\frac{2r_{i+\frac{1}{2}} u_{r_{\mathbf{I}+\mathbf{e}_r}}}{\Delta r_i + \Delta r_{i+1}} \right)}^{\text{from cell } \mathbf{I}+2\,\mathbf{e}_r} \underbrace{+\Delta r_i\Delta\theta\Delta z_k \left(\frac{2r_{i+\frac{1}{2}} u_{r_{\mathbf{I}+\mathbf{e}_r}}}{\Delta r_i + \Delta r_{i+1}} - \frac{2r_{i-\frac{1}{2}} u_{r_{\mathbf{I}-\mathbf{e}_r}}}{\Delta r_{i-1} + \Delta r_i} \right)}_{\text{from cell } \mathbf{I}} \Bigg], \tag{A.4}$$

which after simplification reduces to $-\Delta\theta\Delta z_k \left(r_{i+\frac{1}{2}} u_{r_{\mathbf{I}+\mathbf{e}_r}} - r_{i-\frac{1}{2}} u_{r_{\mathbf{I}-\mathbf{e}_r}} \right)$. In a similar way, collecting all angular contributions to $p_{\mathbf{I}}$ gives:

$$-\frac{\Delta r_i \Delta z_k}{2}\Bigg[\overbrace{-u_{\theta_{\mathbf{I}-\mathbf{e}_\theta}}}^{\text{from cell } \mathbf{I}-2\,\mathbf{e}_\theta} \underbrace{-u_{\theta_{\mathbf{I}-\mathbf{e}_\theta}} + u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}}_{\text{from cell } \mathbf{I}} \overbrace{+u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}}^{\text{from cell } \mathbf{I}+2\,\mathbf{e}_\theta} \Bigg], \tag{A.5}$$

which reduces to $-\Delta r_i \Delta z_k (u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}} - u_{\theta_{\mathbf{I}-\mathbf{e}_\theta}})$. Finally, for the axial contributions, we get:

$$-\frac{r_i \Delta r_i \Delta\theta}{2}\Bigg[\overbrace{-\frac{2\Delta z_{k-1}}{\Delta z_{k-1} + \Delta z_k} u_{z_{\mathbf{I}-\mathbf{e}_z}}}^{\text{from cell } \mathbf{I}-2\,\mathbf{e}_z} \overbrace{-\frac{2\Delta z_k}{\Delta z_{k-1} + \Delta z_k} u_{z_{\mathbf{I}-\mathbf{e}_z}} + \frac{2\Delta z_k}{\Delta z_k + \Delta z_{k+1}} u_{z_{\mathbf{I}+\mathbf{e}_z}}}^{\text{from cell } \mathbf{I}} \underbrace{+\frac{2\Delta z_{k+1}}{\Delta z_k + \Delta z_{k+1}} u_{z_{\mathbf{I}+\mathbf{e}_z}}}_{\text{from cell } \mathbf{I}+2\,\mathbf{e}_z} \Bigg], \tag{A.6}$$

which reduces to $-r_i \Delta r_i \Delta\theta (u_{z_{\mathbf{I}+\mathbf{e}_z}} - u_{z_{\mathbf{I}-\mathbf{e}_z}})$. Comparing the coefficient for $p_{\mathbf{I}}$ from equation A.2 and the cumulative coefficient from equations A.4, A.5 and A.6 shows that after summation over the entire computational grid it holds that $(\mathbf{Du}, p)_{HC} = -(\mathbf{u}, \overline{\mathbf{G}} p)_{\mathcal{HS}}$. □

Proposition. *For any* $\boldsymbol{\omega} \in \mathcal{HL}$ *and* $\mathbf{u} \in \mathcal{HS}$ *on an infinite domain, it holds that* $(\mathbf{C}\boldsymbol{\omega}, \mathbf{u})_{\mathcal{HS}} = (\boldsymbol{\omega}, \overline{\mathbf{C}}\mathbf{u})_{\mathcal{HL}}$.

Proof. Let $\boldsymbol{\omega} = (\eta, \omega, \zeta) \in \mathcal{HL}$ and $\mathbf{u} = (u_r, u_\theta, u_z) \in \mathcal{HS}$. We collect the components of $\boldsymbol{\omega}$ in both inner products. For $\eta_{\mathbf{I}+\mathbf{e}_\theta+\mathbf{e}_z}$, there are contributions from cells $\mathbf{I}$, $\mathbf{I}+2\,\mathbf{e}_\theta$, $\mathbf{I}+2\,\mathbf{e}_z$ and $\mathbf{I}+2\,\mathbf{e}_\theta+2\,\mathbf{e}_z$. The sum of the contributions to the term $\eta_{\mathbf{I}+\mathbf{e}_\theta+\mathbf{e}_z}$ from the inner product $(\mathbf{C}\boldsymbol{\omega}, \mathbf{u})_{\mathcal{HS}}$ is:

$$\begin{aligned}
&\frac{r_i \Delta r_i \Delta\theta \Delta z_k}{2}\left[\overbrace{\frac{u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}}{\Delta z_k} - \frac{u_{z_{\mathbf{I}+\mathbf{e}_z}}}{r_i\Delta\theta}}^{\text{from cell } \mathbf{I}} + \overbrace{\frac{u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}}{\Delta z_k} + \frac{u_{z_{\mathbf{I}+2\,\mathbf{e}_\theta+\mathbf{e}_z}}}{r_i\Delta\theta}}^{\text{from cell } \mathbf{I}+2\,\mathbf{e}_\theta}\right] \\
&+ \frac{r_i \Delta r_i \Delta\theta \Delta z_{k+1}}{2}\left[\underbrace{-\frac{u_{\theta_{\mathbf{I}+\mathbf{e}_\theta+2\,\mathbf{e}_z}}}{\Delta z_{k+1}} - \frac{u_{z_{\mathbf{I}+\mathbf{e}_z}}}{r_i\Delta\theta}}_{\text{from cell } \mathbf{I}+2\,\mathbf{e}_z} \underbrace{-\frac{u_{\theta_{\mathbf{I}+\mathbf{e}_\theta+2\,\mathbf{e}_z}}}{\Delta z_{k+1}} + \frac{u_{z_{\mathbf{I}+2\,\mathbf{e}_\theta+\mathbf{e}_z}}}{r_i\Delta\theta}}_{\text{from cell } \mathbf{I}+2\,\mathbf{e}_\theta+2\,\mathbf{e}_z}\right],
\end{aligned} \tag{A.7}$$

which reduces to:

$$r_i \Delta r_i \Delta\theta \frac{\Delta z_k + \Delta z_{k+1}}{2}\left(\frac{u_{z_{\mathbf{I}+2\,\mathbf{e}_\theta+ez}} - u_{z_{\mathbf{I}+ez}}}{r_i\Delta\theta} - \frac{2(u_{\theta_{\mathbf{I}+\mathbf{e}_\theta+2\,\mathbf{e}_z}} - u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}})}{\Delta z_k + \Delta z_{k+1}}\right). \tag{A.8}$$

The contributions from the inner product $(\boldsymbol{\omega}, \overline{\mathbf{C}}\mathbf{u})_{\mathcal{HL}}$ consist of:

$$\begin{aligned}
&\left[2 \cdot \overbrace{\frac{r_i \Delta r_i \Delta\theta \Delta z_k}{2}}^{\text{from cells } \mathbf{I},\ \mathbf{I}+2\,\mathbf{e}_\theta} + 2 \cdot \overbrace{\frac{r_i \Delta r_i \Delta\theta \Delta z_{k+1}}{2}}^{\text{from cells } \mathbf{I}+2\,\mathbf{e}_z, \mathbf{I}+2\,\mathbf{e}_\theta+2\,\mathbf{e}_z}\right]\frac{(\overline{\mathbf{C}}\mathbf{u})_{r_{\mathbf{I}+\mathbf{e}_\theta+\mathbf{e}_z}}}{2} \\
&\qquad = r_i \Delta r_i \Delta\theta \frac{\Delta z_k + \Delta z_{k+1}}{2}(\overline{\mathbf{C}}\mathbf{u})_{r_{\mathbf{I}+\mathbf{e}_\theta+\mathbf{e}_z}},
\end{aligned} \tag{A.9}$$

which after substitution of equation 2.44 becomes exactly equation A.8. Hence, for an arbitrary $\eta_{\mathbf{I}+\mathbf{e}_\theta+\mathbf{e}_z}$, both inner products yield the same contributions. Then we collect all contributions to $\omega_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_z}$. These stem from cells $\mathbf{I}$, $\mathbf{I}+2\,\mathbf{e}_r$, $\mathbf{I}+2\,\mathbf{e}_z$ and $\mathbf{I}+2\,\mathbf{e}_r+2\,\mathbf{e}_z$. From the inner product $(\mathbf{C}\boldsymbol{\omega}, \mathbf{u})_{\mathcal{HS}}$, we get:

$$\frac{r_{i+\frac{1}{2}}\Delta\theta}{2}\left[\overbrace{-\Delta r_i u_{r_{\mathbf{I}+\mathbf{e}_r}} + \Delta z_k u_{z_{\mathbf{I}+\mathbf{e}_z}}}^{\text{from cell } \mathbf{I}} \overbrace{-\Delta r_{i+1} u_{r_{\mathbf{I}+\mathbf{e}_r}} - \Delta z_k u_{z_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_z}}}^{\text{from cell } \mathbf{I}+2\,\mathbf{e}_r}\right. \tag{A.10}$$

$$+\underbrace{\Delta r_i u_{r_{\mathbf{I}+\mathbf{e}_r+2\,\mathbf{e}_z}}+\Delta z_{k+1}u_{z_{\mathbf{I}+\mathbf{e}_z}}}_{\text{from cell } \mathbf{I}+2\,\mathbf{e}_z}+\underbrace{\Delta r_{i+1}u_{r_{\mathbf{I}+\mathbf{e}_r+2\,\mathbf{e}_z}}-\Delta z_{k+1}u_{z_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_z}}}_{\text{from cell } \mathbf{I}+2\,\mathbf{e}_r+2\,\mathbf{e}_z}\Bigg],$$

which reduces to:

$$r_{i+\frac{1}{2}}\frac{\Delta r_i+\Delta r_{i+1}}{2}\Delta\theta\frac{\Delta z_k+\Delta z_{k+1}}{2}\cdot \left(\frac{2(u_{r_{\mathbf{I}+\mathbf{e}_r+2\,\mathbf{e}_z}}-u_{r_{\mathbf{I}+\mathbf{e}_r}})}{\Delta z_k+\Delta z_{k+1}}-\frac{2\left(u_{z_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_z}}-u_{z_{\mathbf{I}+\mathbf{e}_z}}\right)}{\Delta r_i+\Delta r_{i+1}}\right). \tag{A.11}$$

From the inner product $(\boldsymbol{\omega},\overline{\mathbf{C}}\mathbf{u})_{\mathcal{HL}}$, we collect:

$$\Bigg[\overbrace{\frac{r_{i+\frac{1}{2}}\Delta r_i\Delta\theta\Delta z_k}{4}}^{\text{from cell } \mathbf{I}}+\overbrace{\frac{r_{i+\frac{1}{2}}\Delta r_i\Delta\theta\Delta z_{k+1}}{4}}^{\text{from cell } \mathbf{I}+2\,\mathbf{e}_z}+ \underbrace{\frac{r_{i+\frac{1}{2}}\Delta r_{i+1}\Delta\theta\Delta z_k}{4}}_{\text{from cell } \mathbf{I}+2\,\mathbf{e}_r}+\underbrace{\frac{r_{i+\frac{1}{2}}\Delta r_{i+1}\Delta\theta\Delta z_{k+1}}{4}}_{\text{from cell } \mathbf{I}+2\,\mathbf{e}_r+2\,\mathbf{e}_z}\Bigg](\overline{\mathbf{C}}\mathbf{u})_{\theta_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_z}} \tag{A.12}$$

$$=r_{i+\frac{1}{2}}\frac{\Delta r_i+\Delta r_{i+1}}{2}\Delta\theta\frac{\Delta z_k+\Delta z_{k+1}}{2}(\overline{\mathbf{C}}\mathbf{u})_{\theta_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_z}}. \tag{A.13}$$

After substitution of $(\overline{\mathbf{C}}\mathbf{u})_{\theta_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_z}}$ from equation 2.46, we get exactly the result of equation A.11. Finally, collecting all contributions to $\zeta_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_\theta}$ from cells $\mathbf{I}$, $\mathbf{I}+2\,\mathbf{e}_r$, $\mathbf{I}+2\,\mathbf{e}_\theta$ and $\mathbf{I}+2\,\mathbf{e}_r+2\,\mathbf{e}_\theta$, we get from the inner product $(\mathbf{C}\boldsymbol{\omega},\mathbf{u})_{\mathcal{HS}}$:

$$\frac{\Delta z_k}{2}\Bigg[\overbrace{\Delta r_i u_{r_{\mathbf{I}+\mathbf{e}_r}}-r_i\Delta\theta u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}}^{\text{from cell } \mathbf{I}}+\overbrace{\Delta r_{i+1}u_{r_{\mathbf{I}+\mathbf{e}_r}}+r_{i+1}\Delta\theta u_{\theta_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_\theta}}}^{\text{from cell } \mathbf{I}+2\,\mathbf{e}_r} \tag{A.14}$$

$$\underbrace{-\Delta r_i u_{r_{\mathbf{I}+\mathbf{e}_r+2\,\mathbf{e}_\theta}}-r_i\Delta\theta u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}}_{\text{from cell } \mathbf{I}+2\,\mathbf{e}_\theta}\underbrace{-\Delta r_{i+1}u_{r_{\mathbf{I}+\mathbf{e}_\theta+2\,\mathbf{e}_\theta}}+r_{i+1}\Delta\theta u_{\theta_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_\theta}}}_{\text{from cell } \mathbf{I}+2\,\mathbf{e}_r+2\,\mathbf{e}_\theta}\Bigg],$$

which reduces to:

$$\frac{\Delta r_i+\Delta r_{i+1}}{2}\Delta\theta\Delta z_k\cdot \left(\frac{2\left(r_{i+1}u_{\theta_{\mathbf{I}+2\,\mathbf{e}_r+\mathbf{e}_\theta}}-r_i u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}\right)}{\Delta r_i+\Delta r_{i+1}}-\frac{u_{r_{\mathbf{I}+\mathbf{e}_r+2\,\mathbf{e}_\theta}}-u_{r_{\mathbf{I}+\mathbf{e}_r}}}{\Delta\theta}\right). \tag{A.15}$$

From the inner product $(\boldsymbol{\omega}, \overline{\mathbf{C}}\mathbf{u})_{\mathcal{HL}}$, we collect:

$$\left[\overbrace{\frac{r_i + r_{i+1}}{2} \frac{\Delta r_i \Delta\theta \Delta z_k}{2}}^{\text{from cells } \mathbf{I} \text{ and } \mathbf{I}+2\,\mathbf{e}_\theta} + \overbrace{\frac{r_i + r_{i+1}}{2} \frac{\Delta r_{i+1} \Delta\theta \Delta z_k}{2}}^{\text{from cells } \mathbf{I}+2\,\mathbf{e}_r \text{ and } \mathbf{I}+2\,\mathbf{e}_r+2\,\mathbf{e}_\theta} \right] (\overline{\mathbf{C}}\mathbf{u})_{z_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_\theta}} \tag{A.16}$$

$$= \frac{r_i + r_{i+1}}{2} \frac{\Delta r_i + \Delta r_{i+1}}{2} \Delta\theta \Delta z_k (\overline{\mathbf{C}}\mathbf{u})_{z_{\mathbf{I}+\mathbf{e}_r+\mathbf{e}_\theta}} \tag{A.17}$$

After substitution of equation 2.48, the resulting expression is exactly equation A.15. For the specific case of $\widetilde{\zeta}_{k_0}$, the inner product $(\mathbf{C}\boldsymbol{\omega}, \mathbf{u})_{\mathcal{HS}}$ get contributions from all cells around the axis:

$$\frac{r_1 \Delta r_i \Delta\theta \Delta z_{k_0}}{2} \sum_{\mathbf{I}, i=1, k=k_0} \frac{u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}} + u_{\theta_{\mathbf{I}-\mathbf{e}_\theta}}}{\Delta r_i} = r_1 \Delta\theta \Delta z_{k_0} \sum_{\mathbf{I}, i=1, k=k_0} u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}, \tag{A.18}$$

while the inner product $(\boldsymbol{\omega}, \overline{\mathbf{C}}\mathbf{u})_{\mathcal{HL}}$ yields:

$$N_\theta \frac{r_1 \Delta r_1 \Delta\theta \Delta z_{k_0}}{4} (\overline{\mathbf{C}}\mathbf{u})_{z_{0,k_0}}, \tag{A.19}$$

and after substitution of equation 2.49, this gives:

$$N_\theta \frac{r_1 \Delta r_1 \Delta\theta \Delta z_{k_0}}{4} \frac{4}{\Delta r_1 N_\theta} \sum_{\mathbf{I}, i=1, k=k_0} u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}} = r_1 \Delta\theta \Delta z_{k_0} \sum_{\mathbf{I}, i=1, k=k_0} u_{\theta_{\mathbf{I}+\mathbf{e}_\theta}}, \tag{A.20}$$

which is precisely equation A.18. □

Bibliography

[1] A. Abba and L. Bonaventura. A mimetic finite difference discretization for the incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 56(8):1101–1106, 2008.

[2] J.D.P. Araújo, J.M. Miranda, A.M.F.R. Pinto, and J.B.L.M. Campos. Wide-ranging survey on the laminar flow of individual Taylor bubbles rising through stagnant Newtonian liquids. *International Journal of Multiphase Flow*, 43:131–148, 2012.

[3] T. Atmakidis and E.Y. Kenig. A study on the Kelvin-Helmholtz instability using two different computational fluid dynamics methods. *Journal of Computational Multiphase Flows*, 2(1):33–45, 2010.

[4] A. Bagué, D. Fuster, S. Popinet, R. Scardovelli, and S. Zaleski. Instability growth rate of two-phase mixing layers from a linear eigenvalue problem and an initial-value problem. *Physics of Fluids*, 22(9):092104, 2010.

[5] J.W. Banks, T. Aslam, and W.J. Rider. On sub-linear convergence for linearly degenerate waves in capturing schemes. *Journal of Computational Physics*, 227 (14):6985–7002, 2008.

[6] E. Barbosa and O. Daube. A finite difference method for 3D incompressible flows in cylindrical coordinates. *Computers & Fluids*, 34(8):950–971, 2005.

[7] I. Barmak, A. Y. Gelfgat, A. Ullmann, and N. Brauner. On the Squire's transformation for stratified two-phase flows in inclined channels. *International Journal of Multiphase Flow*, 88:142–151, 2017.

[8] D. Barnea and Y. Taitel. Kelvin-Helmholtz stability criteria for stratified flow: viscous versus non-viscous (inviscid) approaches. *International Journal of Multiphase Flow*, 19(4):639–649, 1993.

[9] Y. Bartosiewicz, J. Laviéville, and J.-M. Seynhaeve. A first assessment of the NEPTUNE_CFD code: Instabilities in a stratified flow comparison between the VOF method and a two-field approach. *International Journal of Heat and Fluid Flow*, 29(2):460–478, 2008.

[10] M. Birvalski. *Experiments in stratified gas-liquid pipe flow*. PhD thesis, Delft University of Technology, 2015.

[11] P.A.M. Boomkamp, B.J. Boersma, R.H.M. Miesen, and G.V. Beijnon. A Chebyshev collocation method for solving two-phase flow stability problems. *Journal of Computational Physics*, 132(2):191–200, 1997.

[12] G. Bornia, A. Cervone, S. Manservisi, R. Scardovelli, and S. Zaleski. On the properties and limitations of the height function method in two-dimensional Cartesian geometry. *Journal of Computational Physics*, 230(4):851–862, 2011.

[13] A. Bourlioux. A coupled level-set volume-of-fluid algorithm for tracking material interfaces. In *Proceedings of the 6th international symposium on computational fluid dynamics, Lake Tahoe, CA*, volume 15, 1995.

[14] J.U. Brackbill, D.B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *Journal of Computational Physics*, 100(2):335–354, 1992.

[15] J.D. Bugg and G.A. Saad. The velocity field around a Taylor bubble rising in a stagnant viscous fluid: numerical and experimental results. *International Journal of Multiphase Flow*, 28(5):791–803, 2002.

[16] J.C. Campbell and M.J. Shashkov. A tensor artificial viscosity using a mimetic finite difference algorithm. *Journal of Computational Physics*, 172(2):739–765, 2001.

[17] J.C. Campbell, J.M. Hyman, and M.J. Shashkov. Mimetic finite difference operators for second-order tensors on unstructured grids. *Computers & Mathematics with Applications*, 44(1-2):157–173, 2002.

[18] S.J. Cummins, M.M. Francois, and D.B. Kothe. Estimating curvature from volume fractions. *Computers & Structures*, 83(6):425–434, 2005.

[19] L. B. da Veiga and G. Manzini. A higher-order formulation of the mimetic finite difference method. *SIAM Journal on Scientific Computing*, 31(1):732–760, 2008.

[20] B.J. Daly. Numerical study of the effect of surface tension on interface instability. *Physics of Fluids (1958-1988)*, 12(7):1340–1354, 1969.

[21] F. Denner and B.G.M. van Wachem. Fully-coupled balanced-force VOF framework for arbitrary meshes with least-squares curvature evaluation from volume fractions. *Numerical Heat Transfer, Part B: Fundamentals*, 65(3):218–255, 2014.

[22] F. Denner, D.R. van der Heul, G.T. Oud, M.M. Villar, A. da Silveira Neto, and B.G.M. van Wachem. Comparative study of mass-conserving interface capturing frameworks for two-phase flows with surface tension. *International Journal of Multiphase Flow*, 61:37–47, 2014.

[23] O. Desjardins, G. Blanquart, G. Balarac, and H. Pitsch. High order conservative finite difference scheme for variable density low Mach number turbulent flows. *Journal of Computational Physics*, 227(15):7125–7159, 2008.

[24] P.G. Drazin and W.H. Reid. *Hydrodynamic stability*. Cambridge University Press, 2004.

[25] A. du Chéné, C. Min, and F. Gibou. Second-order accurate computation of curvatures in a level set framework using novel high-order reinitialization schemes. *Journal of Scientific Computing*, 35(2-3):114–131, 2008.

[26] V. Dyadechko and M. Shashkov. Moment-of-fluid interface reconstruction. *Los Alamos report LA-UR-05-7571*, 2005.

[27] V. Dyadechko and M. Shashkov. Reconstruction of multi-material interfaces from moment data. *Journal of Computational Physics*, 227(11):5361–5384, 2008.

[28] J.G.M Eggels. *Direct and large eddy simulation of turbulent flow in a cylindrical pipe geometry*. PhD thesis, Delft University of Technology, 1994.

[29] Å. Ervik, K.Y. Lervåg, and S.T. Munkejord. A robust method for calculating interface curvature and normal vectors using an extracted local level set. *Journal of Computational Physics*, 257:259–277, 2014.

[30] E. Fan and M. Bussmann. Piecewise linear volume tracking in spherical coordinates. *Applied Mathematical Modelling*, 37(5):3077–3092, 2013.

[31] S. Fleau, S. Mimouni, N. Mérigoux, and S. Vincent. Simulations of two-phase flows with a multifield approach. In *Proceedings of Computational Heat Transfer conference CHT-15*, May 2015.

[32] M.M. Francois, S.J. Cummins, E.D. Dendy, D.B. Kothe, J.M. Sicilian, and M.W. Williams. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *Journal of Computational Physics*, 213(1):141–173, 2006.

[33] K. Fukagata and N. Kasagi. Highly energy-conservative finite difference method for the cylindrical coordinate system. *Journal of Computational Physics*, 181(2):478–498, 2002.

[34] M. Germano. On the effect of torsion on a helical pipe flow. *Journal of Fluid Mechanics*, 125:1–8, 1982.

[35] V.R. Gopala and B.G.M. van Wachem. Volume of fluid methods for immiscible-fluid and free-surface flows. *Chemical Engineering Journal*, 141(1):204–221, 2008.

[36] M.D. Griffin, E. Jones, and J.D. Anderson. A computational fluid dynamic technique valid at the centerline for non-axisymmetric problems in cylindrical coordinates. *Journal of Computational Physics*, 30(3):352–360, 1979.

[37] L.-J. Guo, G.-J. Li, and X.-J. Chen. A linear and non-linear analysis on interfacial instability of gas-liquid two-phase flow through a circular pipe. *International Journal of Heat and Mass Transfer*, 45(7):1525–1534, 2002.

[38] F.H. Harlow and J.E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, 8(12):2182, 1965.

[39] D.J.E. Harvie, M.R. Davidson, and M. Rudman. An analysis of parasitic current generation in volume of fluid simulations. *Applied Mathematical Modelling*, 30 (10):1056–1066, 2006.

[40] J. Hernández, J. López, P. Gómez, C. Zanzi, and F. Faura. A new volume of fluid method in three dimensions-part i: Multidimensional advection method with face-matched flux polyhedra. *International Journal for Numerical Methods in Fluids*, 58(8):897–921, 2008.

[41] C.W. Hirt and B.D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39(1):201–225, 1981.

[42] J.G. Hnat and J.D. Buckmaster. Spherical cap bubbles and skirt formation. *Physics of Fluids (1958-1988)*, 19(2):182–194, 1976.

[43] J.M. Hyman and M. Shashkov. Adjoint operators for the natural discretizations of the divergence, gradient and curl on logically rectangular grids. *Applied Numerical Mathematics*, 25(4):413–442, 1997.

[44] J.M. Hyman and M. Shashkov. Natural discretizations for the divergence, gradient, and curl on logically rectangular grids. *Computers & Mathematics with Applications*, 33(4):81–104, 1997.

[45] W. Jang, J. Jilesen, F.S. Lien, and H. Ji. A study on the extension of a VOF/PLIC based method to a curvilinear co-ordinate system. *International Journal of Computational Fluid Dynamics*, 22(4):241–257, 2008.

[46] A.M.C. Janse, P.E. Dijk, and J.A.M. Kuipers. The volume of fluid method in spherical coordinates. *International Journal of Numerical Methods for Heat & Fluid Flow*, 10(7):654–674, 2000.

[47] M. Jemison, E. Loch, M. Sussman, M. Shashkov, M. Arienti, M. Ohta, and Y. Wang. A coupled level set-moment of fluid method for incompressible two-phase flows. *Journal of Scientific Computing*, 54(2-3):454–491, 2013.

[48] S.Y. Kadioglu and D.A. Knoll. A fully second order implicit/explicit time integration technique for hydrodynamics plus nonlinear heat conduction problems. *Journal of Computational Physics*, 229(9):3237–3249, 2010.

[49] S.Y. Kadioglu and D.A. Knoll. A jacobian-free Newton Krylov implicit-explicit time integration method for incompressible flow problems. *Communications in Computational Physics*, 13(05):1408–1431, 2013.

[50] C. Kang, S. Quan, and J. Lou. Numerical study of a Taylor bubble rising in stagnant liquids. *Phys. Rev. E*, 81:066308, 2010.

[51] M. Kang, R. Fedkiw, and X. Liu. A boundary condition capturing method for multiphase incompressible flow. *Journal of Scientific Computing*, 15(3):323–360, 2000.

[52] J. Li and Y. Renardy. Numerical study of flows of two immiscible liquids at low Reynolds number. *SIAM review*, 42(3):417–439, 2000.

[53] K. Lipnikov, G. Manzini, and M. Shashkov. Mimetic finite difference method. *Journal of Computational Physics*, 257:1163–1227, 2014.

[54] J.-G. Liu and W.-C. Wang. Energy and helicity preserving schemes for hydro- and magnetohydro-dynamics flows with symmetry. *Journal of Computational Physics*, 200(1):8–33, 2004.

[55] E.W. Llewellin, E. Del Bello, J. Taddeucci, P. Scarlato, and S.J. Lane. The thickness of the falling film of liquid around a Taylor bubble. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 468(2140):1041–1064, 2012.

[56] J. López and J. Hernández. On reducing interface curvature computation errors in the height function technique. *Journal of Computational Physics*, 229(13):4855–4868, 2010.

[57] X. Lu and A. Prosperetti. A numerical study of Taylor bubbles. *Industrial & Engineering Chemistry Research*, 48(1):242–252, 2008.

[58] P. Macklin and J. Lowengrub. An improved geometry-aware curvature discretization for level set methods: application to tumor growth. *Journal of Computational Physics*, 215(2):392–401, 2006.

[59] H.K. Moffatt and A. Tsinober. Helicity in laminar and turbulent flow. *Annual Review of Fluid Mechanics*, 24(1):281–312, 1992.

[60] Y. Morinishi, T.S. Lund, O.V. Vasilyev, and P. Moin. Fully conservative higher order finite difference schemes for incompressible flow. *Journal of Computational Physics*, 143(1):90–124, 1998.

[61] Y. Morinishi, O.V. Vasilyev, and T. Ogi. Fully conservative finite difference scheme in cylindrical coordinates for incompressible flow simulations. *Journal of Computational Physics*, 197(2):686–710, 2004.

[62] K. Mukundakrishnan, S. Quan, D.M. Eckmann, and P.S. Ayyaswamy. Numerical study of wall effects on buoyant gas-bubble rise in a liquid-filled finite cylinder. *Physical Review E*, 76(3):036308, 2007.

[63] K. Najmi, B.S. McLaury, S.A. Shirazi, and S. Cremaschi. Experimental study of low concentration sand transport in wet gas flow regime in horizontal pipes. *Journal of Natural Gas Science and Engineering*, 24:80–88, 2015.

[64] R. Nourgaliev, S. Kadioglu, and V. Mousseau. Marker redistancing/level set method for high-fidelity implicit interface tracking. *SIAM Journal on Scientific Computing*, 32(1):320–348, 2010.

[65] G. Ooms. The hydrodynamic stability of core-annular flow of two ideal liquids. *Applied Scientific Research*, 26(1):147–158, 1972.

[66] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[67] G.T. Oud, D.R. van der Heul, C. Vuik, and R.A.W.M. Henkes. A fully conservative mimetic discretization of the Navier-Stokes equations in cylindrical coordinates with associated singularity treatment. *Journal of Computational Physics*, 325:314–337, 2016.

[68] M. Owkes and O. Desjardins. A Discontinuous Galerkin conservative level set scheme for interface capturing in multiphase flows. *Journal of Computational Physics*, 249:275–302, 2013.

[69] S. Popinet and S. Zaleski. A front-tracking algorithm for accurate representation of surface tension. *International Journal for Numerical Methods in Fluids*, 30 (6):775–793, 1999.

[70] L.G. Rebholz. An energy- and helicity-conserving finite element scheme for the Navier-Stokes equations. *SIAM Journal on Numerical Analysis*, 45(4):1622–1638, 2007.

[71] M. Renardy. A possible explanation of âĂIJbamboo wavesâĂİ in core-annular flow of two liquids. *Theoretical and Computational Fluid Dynamics*, 4(2):95–99, 1992.

[72] Y. Renardy and M. Renardy. Prost: a parabolic reconstruction of surface tension for the volume-of-fluid method. *Journal of Computational Physics*, 183(2):400–421, 2002.

[73] G. Russo and P. Smereka. A remark on computing distance functions. *Journal of Computational Physics*, 163(1):51–67, 2000.

[74] K. Salari and P. Knupp. Code verification by the Method of Manufactured Solutions. Technical report, Sandia National Labs., Albuquerque, NM (US); Sandia National Labs., Livermore, CA (US), 2000.

[75] Y. Salhi, E.-K. Si-Ahmed, J. Legrand, and G. Degrez. Stability analysis of inclined stratified two-phase gas-liquid flow. *Nuclear Engineering and Design*, 240 (5):1083–1096, 2010.

[76] B. Sanderse. Energy-conserving Runge-Kutta methods for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 233:100–131, 2013.

[77] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, second edition, 1999. ISBN 9780521645577.

[78] M. Shashkov. *Conservative Finite-Difference Methods on General Grids*. Symbolic & Numeric Computation. Taylor & Francis, 1995. ISBN 9780849373756.

[79] K. Sidi-Ali and R. Gatignol. Interfacial friction factor determination using cfd simulations in a horizontal stratified two-phase flow. *Chemical Engineering Science*, 65(18):5160–5169, 2010.

[80] L. Štrubelj and I. Tiselj. Numerical simulations of basic interfacial instabilities with incompressible two-fluid model. *Nuclear Engineering and Design*, 241(4): 1018–1023, 2011.

[81] D.L. Sun and W.Q. Tao. A coupled volume-of-fluid and level set (VOSET) method for computing incompressible two-phase flows. *International Journal of Heat and Mass Transfer*, 53(4):645–655, 2010.

[82] M. Sussman and M. Ohta. High-order techniques for calculating surface tension forces. In *Free Boundary Problems*, pages 425–434. Springer, 2006.

[83] M. Sussman and E.G. Puckett. A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. *Journal of Computational Physics*, 162(2):301–337, 2000.

[84] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114(1):146–159, 1994.

[85] M. Sussman, K.M. Smith, M.Y. Hussaini, M. Ohta, and R. Zhi-Wei. A sharp interface method for incompressible two-phase flows. *Journal of Computational Physics*, 221(2):469–505, 2007.

[86] T. Taha and Z.F. Cui. CFD modelling of slug flow in vertical tubes. *Chemical Engineering Science*, 61(2):676–687, 2006.

[87] Y. Taitel and A.E. Dukler. A model for predicting flow regime transitions in horizontal and near horizontal gas-liquid flow. *AIChE Journal*, 22(1):47–55, 1976.

[88] S.A. Thorpe. Experiments on the instability of stratified shear flows: immiscible fluids. *Journal of Fluid Mechanics*, 39(01):25–48, 1969.

[89] A. Ullmann and N. Brauner. Closure relations for two-fluid models for two-phase stratified smooth and stratified wavy flows. *International Journal of Multiphase Flow*, 32(1):82–105, 2006.

[90] S.P. Van der Pijl, A. Segal, C. Vuik, and P. Wesseling. A mass-conserving level-set method for modelling of multi-phase flows. *International Journal for Numerical Methods in Fluids*, 47(4):339–361, 2005.

[91] R.W.C.P. Verstappen and A.E.P. Veldman. Symmetry-preserving discretization of turbulent flow. *Journal of Computational Physics*, 187(1):343–368, 2003.

[92] R. Verzicco and P. Orlandi. A finite difference scheme for three-dimensional incompressible flows in cylindrical coordinates. *Journal of Computational Physics*, 123(2):402–414, 1996.

[93] M. Vinokur. Conservation equations of gasdynamics in curvilinear coordinate systems. *Journal of Computational Physics*, 14(2):105–125, 1974.

[94] G.D. Weymouth and D. Yue. Conservative volume-of-fluid method for free-surface simulations on cartesian-grids. *Journal of Computational Physics*, 229(8): 2853–2865, 2010.

[95] E.T. White and R.H. Beardmore. The velocity of rise of single cylindrical air bubbles through liquids contained in vertical tubes. *Chemical Engineering Science*, 17(5):351–361, 1962.

[96] X. Yang, A.J. James, J. Lowengrub, X. Zheng, and V. Cristini. An adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids. *Journal of Computational Physics*, 217(2):364–394, 2006.

[97] C. Yih. Stability of two-dimensional parallel flows for three-dimensional disturbances. *Quarterly of Applied Mathematics*, 12(4):434–435, 1955.

[98] M. Zoeteweij. *Long liquid slugs in horizontal tubes*. PhD thesis, Delft University of Technology, 2007.

Summary

This research entails the numerical simulation of physical flow instabilities that can occur in two-phase pipe flows with a new efficient algorithm. The fluids are assumed to be immiscible, and the flow is incompressible and isothermal in a straight circular pipe section with a certain inclination. The numerical algorithm that was developed consists of a flow solver and a sharp interface model that solve the Navier-Stokes equations in cylindrical coordinates. The simulation results obtained with the new method are validated through comparison with other models and with experiments. The focus lies on obtaining an accurate and efficient algorithm that can ultimately be used for Direct Numerical Simulations or Large Eddy Simulations of turbulent two-phase pipe flows.

First, an improved approximation of the Navier-Stokes equations in cylindrical coordinates was derived based on mimetic principles. Instead of minimizing the truncation error, this approach aims at discretely satisfying as many of the continuous vector identities as possible. The resulting finite difference discretization spatially conserves mass, momentum and kinetic energy on non-uniform grids for inviscid flows when used in conjunction with a suitable time integration method. Extensive numerical testing demonstrates that the discretization is second order accurate. In particular, the coordinate singularity at the pipe centre, which causes significant problems in existing numerical methods, is incorporated consistently in the mimetic framework.

The proposed discretization is currently only applicable to constant-viscosity flows, and therefore a different variable-viscosity Navier-Stokes discretization (that does not conserve kinetic energy) is used for the two-phase flow simulations. In this approach, the Navier-Stokes equations in cylindrical coordinates are solved on a staggered cylindrical grid. The spatial discretization is based on a second order finite difference discretization of the Navier-Stokes equations, and a projection scheme is

used to advance the flow field in time. The momentum equations are linearized using Newton linearization, and the resulting semi-discrete system is solved using the Implicit Midpoint time integration method with a Jacobi preconditioner. The Poisson equation for the pressure is solved using the Conjugate Gradient method with an Incomplete Cholesky preconditioner (block IC for the parallel algorithm).

In addition to the flow solver, a dual interface algorithm consisting of an efficient combination of the Level Set method and the Volume of Fluid (VoF) method was constructed in cylindrical coordinates. The straightforward interface representation of the Level Set method is supplemented with the inherent volume conservation of the VoF method, thereby creating a method that is superior to either of its constituents. Both the Level Set function and the VoF function are retained on the cylindrical grid, and they are coupled through a computationally efficient function. A conservative dimensionally split advection algorithm that is second order accurate in space is used for both variables, which results in exact conservation of the volume fractions.

The analysis of the time-explicit advection of the VoF field shows that it is unable to capture the interface accurately whenever it traverses the origin in the polar plane. To remedy this, a VoF redistribution algorithm was derived that significantly improves the VoF advection accuracy near the polar origin.

The jump conditions at the interface are modeled with the Ghost Fluid method. The fluid density is based on the VoF value, while the viscosity is smeared out along a narrow tube around the interface. The approximation of the interface curvature, which is required when surface tension is present, is calculated with the VoF-based Height Function method that has been extended to cylindrical domains.

The complete algorithm has been implemented in Fortran 90 and was validated using a comprehensive test suite. The numerical results were compared with a variety of rising bubbles to investigate the performance and the accuracy of the method. Good agreement is generally found for the terminal rising velocities and for the bubble shapes. Based on modal stability theory, the algorithm shows an excellent ability in capturing the predicted growth factor of a perturbed axisymmetric surface. The full 3D algorithm has finally been validated by comparison with experiments for Kelvin-Helmholtz waves in horizontally inclined pipes.

Samenvatting

Dit onderzoek betreft de numerieke simulatie van fysische instabiliteiten die gevormd kunnen worden in twee-fase buisstromingen met een nieuw efficient algoritme. De fluida worden als onmengbaar beschouwd, en de stroming is incompressibel en isotherm door een rechte buis onder een gegeven helling. Het numerieke algoritme dat is ontwikkeld bestaat uit een stromingsmodel en een scherp interface model die beiden worden gebruikt om de Navier-Stokes vergelijkingen in cylindrische coordinaten op te lossen. De verkregen simulatieresultaten zijn gevalideerd door vergelijking met andere modellen en met experimenten. De nadruk ligt op het verkrijgen van een nauwkeurig en efficient algoritme dat uiteindelijk gebruikt kan worden voor Directe Numerieke Simulaties of Large-Eddy Simulaties van turbulente twee-fase buisstromingen.

De eerste stap in het onderzoek was het afleiden van een verbeterde benadering van de Navier-Stokes vergelijkingen in cylindrische coordinaten gebaseerd op mimetische principes. In plaats van het minimalizeren van de lokale discretizatiefout heeft een mimetische methode het doel om zo veel mogelijk vectoridentiteiten die gelden op continu niveau ook op het discrete niveau te laten gelden. De resulterende eindige differentie discretizatie behoudt zowel massa, momentum als kinetische energie op niet-uniforme roosters voor niet-visceuze stromingen indien een geschikte tijdsintegratiemethode wordt gebruikt. Uitgebreide numerieke testen laten zien dat de discretizatie een tweede orde nauwkeurigheid heeft. De aanpak van de coordinaatsingulariteit in het midden van de buis, welke voor aanzienlijke problemen zorgt in bestaande numerieke methoden, volgt consistent uit de mimetische methode.

De afgeleide discretizatie is momenteel alleen toepasbaar op stromingen met een constante viscositeit. Om die reden is een andere Navier-Stokes discretizatie gebruikt voor de twee-fase stroming simulaties. Deze discretizatie behoudt geen

kinetische energie, maar is wel toepasbaar op stromingen met variable viscositeit. Bij deze aanpak worden de Navier-Stokes vergelijkingen in cylindercoordinaten opgelost op een versprongen cylindrisch rooster. De ruimtelijke discretizatie is gebaseerd op een tweede orde eindige differentie methode, en de tijdsdiscretizatie maakt gebruikt van een projectie-algoritme. De momentumvergelijkingen worden gelinearizeerd met behulp van Newton linearizatie, en het resulterende semidiscrete systeem wordt opgelost door de Impliciete Middelpunt methode met Jacobi voorbewerking. De Poisson vergelijking voor de druk wordt opgelost door de Geconjugeerde Gradienten methode met Incomplete Cholesky voorbewerking (blok IC bij de parallelle implementatie).

Naast de stromingsdiscretizatie is ook een interface algoritme in cylindercoordinaten ontwikkeld dat bestaat uit een efficiente combinatie van de Level Set methode en de Volume of Fluid (VoF) methode. De eenvoudige interface representatie van de Level Set methode wordt aangevuld met het inherente volumebehoud van de VoF methode, waardoor een hybride algoritme ontstaat dat superieur is aan elk van beide individuele technieken. Zowel de Level Set functie als de VoF functie worden gebruikt op het rooster, en beiden worden gekoppeld door een numeriek efficiente relatie. Beide variabelen worden door een tweede orde nauwkeurig advectie algoritme getransporteerd, waarbij individuele volume fracties tot op machine precisie worden behouden.

Een analyze van de expliciete advectie van het VoF veld laat zien dat deze niet in staat blijkt het interface nauwkeurig te modelleren wanneer dit door de oorsprong van het poolvlak beweegt. Om de nauwkeurigheid te vergroten is een VoF herdistributie algoritme ontwikkeld dat significant verbeterde resultaten geeft in de buurt van de cylindrische oorsprong.

De discontinuiteiten die ontstaan bij het interface worden gemodelleerd met de Ghost Fluid methode. The dichtheid van de fluida is gebaseerd op de VoF waarden, en de viscositeit is uitgesmeerd over een dunne laag rond het interface. De benadering van de kromming van het interface, welke nodig is voor stromingen met oppervlaktespanning, is gebaseerd op de Hoogte Functie methode die is uitgebreid naar cylindrische domeinen.

Het volledige algoritme is geimplementeerd in Fortran 90, en vervolgens gevalideerd

aan de hand van een uitgebreid scala aan testen. De numerieke resultaten zijn vergeleken met een aantal experimentele resultaten van stijgende bellen om de nauwkeurigheid van het algoritme te onderzoeken. Er wordt een goede nauwkeurigheid gevonden voor de stijgsnelheden en de vorm van de bellen. Het algoritme toont bijzonder goede resultaten bij het voorspellen van groeifactoren van een verstoord axisymmetrisch interface op basis van modale stabiliteitstheorie. Het volledige 3D algoritme werd uiteindelijk ook gevalideerd door vergelijking met experimenten voor Kelvin-Helmholtz golven in een horizontale buis.

Publications

1. F. Denner, D.R. van der Heul, G.T. Oud, M.M. Villar, A. da Silveira Neto, and B.G.M. van Wachem. *Comparative study of mass-conserving interface capturing frameworks for two-phase flows with surface tension*. International Journal of Multiphase Flow, 61:37âĂŞ47, 2014.

2. G.T. Oud, *Development and validation of the cylindrical Mass Conserving Level Set method*, DIAM technical report, Delft University of Technology, 2014.

3. G.T. Oud, D.R. van der Heul, C. Vuik, and R.A.W.M. Henkes. *A fully conservative mimetic discretization of the Navier-Stokes equations in cylindrical coordinates with associated singularity treatment*. Journal of Computational Physics, 325:314âĂŞ337, 2016.

4. G.T. Oud, D.R. van der Heul, C. Vuik, and R.A.W.M. Henkes. *A dual interface capturing method for the simulation of incompressible immiscible two-phase pipe flows*, to be submitted.

5. G.T. Oud, D.R. van der Heul, C. Vuik, and R.A.W.M. Henkes. *Numerical prediction of two-phase flow instabilities in cylindrical pipes*, to be submitted.

Conferences

- ECCOMAS (2012), Vienna, Austria
  Title: '*CFD for flow instabilities in multiphase systems using a mass-conserving level set method in cylindrical coordinates*'.

- ICNMMF-II (2014), Darmstadt, Germany
  Title: '*CFD for flow instabilities in multiphase systems*'.

- ICMF (2016), Florence, Italy
  Title: '*A moment-based advection scheme for local sharp interface reconstruction in two-phase flows*'.

- WSC Woudschoten (2011-2015), Zeist, The Netherlands
  Title: '*CFD for flow instabilities in multiphase systems*'. (2012)

- Burgersdag (2012-2015), The Netherlands
  Title: '*CFD for flow instabilities in multiphase systems*'. (2012)
  Title: '*Simulation of multiphase pipe flows using a mass-conserving level set method*'. (2013)

Curriculum Vitae

1985	Born in Spaarndam, The Netherlands
1997–2003	Secondary School VWO, N&T profile Mendelcollege, Haarlem, The Netherlands
2003–2008	BSc Aerospace Engineering with minor Applied Mathematics Delft University of Technology, The Netherlands
jun-oct 2008	Scientific Intern at Total S.A. Centre Scientifique et Technique Jean Féger Pau, France
feb-may 2009	Visiting student at Brown University Department of Applied Mathematics Providence, RI, USA.
2008–2011	MSc Aerospace Engineering, Department of Aerodynamics Delft University of Technology, The Netherlands
2008–2011	MSc Applied Mathematics, Department of Analysis Delft University of Technology, The Netherlands
2011–2017	PhD Applied Mathematics, Department of Numerical Analysis Delft University of Technology, The Netherlands Thesis: 'A dual interface method in cylindrical coordinates for two-phase pipe flows' Promotors: prof.dr.ir. C. Vuik & prof.dr.ir. R.A.W.M. Henkes
2017–	Project Manager CFD Offshore Department MARIN, Wageningen, The Netherlands