DELFT UNIVERSITY OF TECHNOLOGY

REPORT 14-03

Development and validation of the cylindrical Mass Conserving Level Set method

G.Oud

ISSN 1389-6520

Reports of the Delft Institute of Applied Mathematics

Delft 2014

Summary

The work described in this report is part of the PhD project 'Computational Fluid Dynamics for 3D multiphase flow instabilities' sponsored by Shell, TNO and Deltares. The project is meant to develop and validate a time-dependent, three-dimensional numerical/physical model for immiscible two-phase flow in pipe lines. This includes the extension of an existing algorithm in cartesian coordinates into cylindrical coordinates to obtain a boundary-fitting grid. Care should be taken to properly represent the interface between two immiscible phases (such as gas and liquid) to obtain conservation of mass.

In this report the Mass Conserving Level Set (MCLS) method for the interface is described and validated using a number of common benchmark test cases. The MCLS method is a numerical method for the simulation of incompressible, immiscible two-phase flows based on an efficient algorithm that combines both the Level Set (LS) method and the Volume of Fluid (VoF) method. The discretization of the Navier-Stokes equations is analyzed and shown to be sufficiently reliable and accurate. The interface representation is then tested through progressively more complex scenarios, starting with a bubble at rest, and concluding with a long bubble rising through a vertical pipe. The overall performance is satisfactory, and no significant obstacles have been encountered. Certain small parts of the algorithm need further improvement, mainly due to accuracy and robustness issues.

This work is part of the four year PhD project of the author at Delft University of Technology, Department of Numerical Analysis. The report is a summary of the work carried out in the first two years of the project.

Contents

Chapter 1

Introduction

The numerical simulation of physical problems involving immiscible two-phase flows (such as gas-liquid flow or liquid-liquid flow) generally relies on the use of one of two methods to represent the sharp interface: the Level Set (LS) method [Fedkiw and Osher, 2003] and the Volume of Fluid (VoF) method [Hirt and Nichols, 1981]. The former method uses a contour of a higher dimensional function to define the interface, i.e. for a fluid in some n-dimensional domain Ω, the interface Γ consists of:

$$\Gamma = \{\mathbf{x} \mid \phi(\mathbf{x}) = \mathbf{0},\ \mathbf{x} \in \Omega\} \tag{1.1}$$

for some scalar function $\phi : \Omega \to \mathbb{R}$. The function ϕ is advected by the underlying flow field $\mathbf{u}$ according to:

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0 \tag{1.2}$$

The VoF method involves a scalar function $\psi : \Omega \to [0, 1]$ which assigns to every computational cell a number representing the volume fraction of one of the two fluids in that cell. A value of $\psi = 0$ or $\psi = 1$ implies a cell filled by either of the two fluids, while a cell with $0 < \psi < 1$ is a mixed cell that is intersected by the interface. Just as with the LS method, the function ψ is advected through:

$$\frac{\partial \psi}{\partial t} + \nabla \cdot (\mathbf{u}\psi) = 0 \tag{1.3}$$

where $\mathbf{u}$ is the underlying flow field. The reconstruction of the sharp interface as well as determining the fluxes of ψ has to be done by considering neighbouring cells.

Both methods mentioned have clear advantages that explain their popularity: the LS method is relatively easy to implement, also for higher dimensions, and the reconstruction of the interface is elegant and nearly trivial. The VoF method is volume-conservative and in general very robust. However, both methods also come with serious drawbacks that may limit their use to incompressible flows: the LS method is not volume-conservative, which should be an essential requirement for any incompressible fluid simulation. Even though the use of higher order methods may improve conservation, long-term simulations will always suffer from loss of volume of one or both fluids. The VoF method, although conservative and robust, generally requires an elaborate algorithm to reconstruct the interface and to calculate fluxes, which can become computationally expensive. Additionally, in the absence of a sharp interface, properties like normals and curvature are more difficult to determine than when using the LS method.

From the considerations above, it seems worthwile to construct an interface representation method that combines the advantageous elements of both LS and VoF method (easy interface reconstruction and volume conservation) and simultaneously avoids the mentioned drawbacks. Such attempts have resulted in for example the CLSVOF method [Sussman and Puckett, 2000] and the MCLS method [Van der Pijl et al., 2005], and more recently in the CLSMOF method [Jemison et al., 2013]. Broadly speaking, all these methods use VoF values to correct any volume loss by the LS method, thereby being formally conservative.

Shell, TNO and Deltares, who have requested the present project, are interested in exploring the possibilities of advanced simulation tools for analyzing multiphase flow in pipeline systems. This includes the combined transport of gas/oil, oil/water, or air/water. Therefore in the present project the existing algorithm with the MCLS method for rectangular geometries using a cartesian grid has been extended to circular pipe geometries using cylindrical coordinates. In this report, the development of the latter approach is described. The cylindrical MCLS method is tested and numerical convergence analysis is performed to validate the various sub-algorithms. Some physical test cases for two-phase flow are used to verify the capabilities of the implemented method. Furthermore opportunities are identified for further improvement. This work is part of the four year PhD project of the author at Delft University of Technology, Department of Numerical Analysis. The report is a summary of the work carried out in the first two years of the project.

Chapter 2

Overview of the algorithm

The computational method used is the Mass Conserving Level Set approach introduced by Van der Pijl et al. [2005]. Both LS and VoF methods are combined to guarantee conservation of mass, whereas the use of only the LS method unavoidably leads to mass loss. This chapter describes the method in cylindrical coordinates.

2.1 Interface representation

The LS function is used to implicitly define the location of the interface. After advection with the flow field during a time step, the newly located interface will in general not enclose the same amount of fluid as at the previous time level, which in time can ultimately lead to complete loss of one of the fluids present initially. To quantify the volume loss during a time step, the LS function is linearized in every computational cell as:

$$\phi(\mathbf{x}) \approx \tilde{\phi}(\mathbf{x}) = \phi(\mathbf{x}_0) + \nabla\phi(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0), \tag{2.1}$$

where $\mathbf{x}_0$ is the location of the cell center. Then for the linearized LS function $\tilde{\phi}$, an analytical expression is used to calculate the associated VoF value $\tilde{\psi}$ in every cell, such that:

$$\tilde{\psi}(\mathbf{x}_0) = f(\tilde{\phi}(\mathbf{x}_0), \nabla\tilde{\phi}(\mathbf{x}_0) \tag{2.2}$$

Notice that, except for rounding errors, this expression is exact. Hence after every time step, a VoF value $\tilde{\psi}$ can be obtained based on the advected LS function. Then there is in addition to the LS function also a VoF function ψ that is advected with the flow from the beginning on, and this VoF value will serve as a reference based on the knowledge that its advection is achieved in a conservative manner. The calculated VoF values $\tilde{\psi}$ are now cellwise compared to the reference values ψ, and the difference $|\tilde{\psi}(\mathbf{x}_0) - \psi(\mathbf{x}_0)|$ serves as the primary indicator of the accuracy of volume conservation. If the difference is smaller than a user-defined value ϵ, the interface advection was accurate enough. If not, corrective action needs to be taken as too much volume is apparently lost during the advection of the interface in that particular cell.

The correction process (or 'coupling' of the LS and VoF functions) provides a LS function $\tilde{\phi}(\mathbf{x}_0)$ which satisfies:

$$|f(\tilde{\phi}(\mathbf{x}_0), \nabla\tilde{\phi}(\mathbf{x}_0)) - \psi(\mathbf{x}_0)| < \epsilon. \tag{2.3}$$

This is achieved through Newton iterations on expression 2.3. Since $\tilde{\phi}(\mathbf{x}_0)$ and $\nabla\tilde{\phi}(\mathbf{x}_0)$ are coupled (the latter being calculated as the central difference of the former), the algorithm incorporates two nested iterations.

2.2 Implementation of interface jump conditions

Under the assumption of a sharp interface representation in combination with incompressible flow, jump conditions at the interface can be derived [Liu et al., 2000]. Although formally the density ρ and the viscosity μ are constants within each fluid, i.e:

$$\rho = \rho_{\phi<0} + H(\phi)(\rho_{\phi\geq 0} - \rho_{\phi<0}) \tag{2.4}$$

$$\mu = \mu_{\phi<0} + H(\phi)(\mu_{\phi\geq 0} - \mu_{\phi<0}) \tag{2.5}$$

choosing only the viscosity μ to be continuous across the interface leads to the following single interface condition:

$$[p] = \sigma\kappa \tag{2.6}$$

where $[\cdot]$ is the jump operator, σ the surface tension coefficient and κ the local interface curvature (twice the mean curvature in 3D). The classic Continuous Surface Force (CSF) model [Brackbill et al., 1992] transforms the jump into a body force which is added to the momentum equations. In addition, the jump is mollified by a smoothed Heaviside function to only contibute in a narrow band around the interface. Although robust and proven in practise, its use in combination with non-uniform grids can lead to difficulties in choosing the bandwidth of the mollifier and inaccuracy when two interfaces become close. In this project, one of the aims is to minimize the number of user-defined input parameters.

The Ghost Fluid Method (GFM) [Liu et al., 2000] treats the jumps in a sharp fashion, and therefore removes the necessity of any smearing parameters. Because of the smoothed viscosity there is no jump in the gradient of the pressure. The GFM then provides an alternative way of discretizing the pressure jump in the momentum equations by incorporating the jump in the Poisson equation through an additional term on the right hand side without altering the stencil of the Laplacian. This allows any black box solver to be used for the Poisson equation and avoids tedious alterations during construction of the Laplacian matrix (or the associated matrix-vector product).

Because of the attractive property of the GFM that no problem-specific parameters are required as compared to the CSF approach, the former is chosen and implemented as surface tension model. As a temporary concession, the viscosity is kept continuous across the interface to simplify implementation. Although this is formally not a complete implementation of the GFM, it is frequently used in practise and proved reliable.

2.3 Accuracy of the interface representation

The use of a cell-wise linearized LS function as interface representation introduces a projection error in the estimation of the associated VoF value, compared to the VoF value of the true LS function. The linearization suggests that this error is $\mathcal{O}(1/N^2)$, and a simple test to verify this comprises a spherical bubble centered in a cylindrical domain with zero initial velocity. The fluid properties inside and outside the bubble are equal, and since there is no fluid movement, the sphere will stay at its initial position. The initial LS function ϕ for a sphere with radius $R = 1$ is given by:

$$\phi(r,\theta,z) = -1 + \sqrt{r^2 + z^2}, \qquad \{(r,\theta,z) \mid 0 \le r \le 2, 0 \le \theta < 2\pi, -2 \le z \le 2\}. \tag{2.7}$$

Since this function cannot be exactly represented by the discretization, errors will be made when calculating the gradient of ϕ required for the function f to calculate the associated VoF value. Besides round off effects, no further errors are expected since the function calculates associated VoF values of *linearized* LS functions exactly. The accuracy is measured by calculating the error ϵ between the exact volume of the sphere and the calculated volume V_{appr} after the projection onto the computational grid, i.e:

$$\epsilon(N) = \left| \frac{4\pi}{3} - V_{appr}(N) \right|, \tag{2.8}$$

where the approximated volume is a function of the degrees of freedom N.

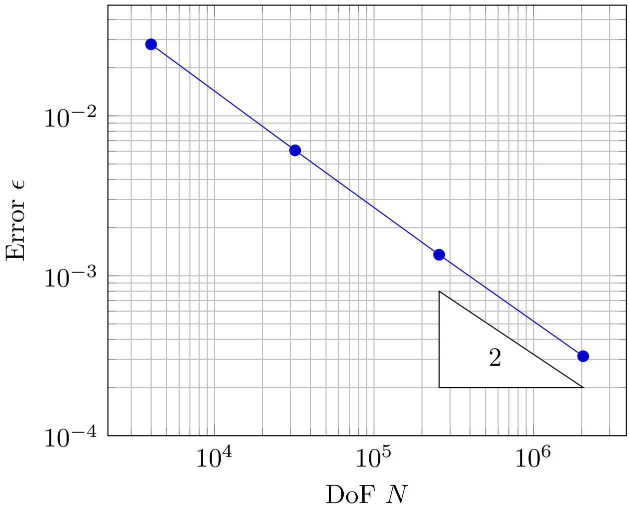


Figure 2.1: Accuracy of the function $f(\phi, \nabla\phi)$.

From figure 2.1 it follows that the error in approximation of the enclosed volume is indeed $\mathcal{O}(1/N^2)$.

Chapter 3

Validation of the flow field

The physical domain of the problem is fixed as a straight cylinder. To be able to use fast existing methods, the boundary fitting cylindrical coordinate system:

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

is chosen to form a computational grid. The staggered configuration places the velocities $\mathbf{u} = (u_r, u_\theta, u_z)$ at the cell faces, and the pressure p, the LS function ϕ and the VoF function ψ in the cell centers. The Navier Stokes equations are discretized using finite difference techniques described in Morinishi et al. [2004]. It is claimed that in the absence of viscosity, the discretization is (kinetic) energy-conserving, but numerical investigation has shown that this is untrue. This is caused by a non-consistent discretization of a limit-derived radial momentum equation at the pole where $r = 0$. Instead, an averaging procedure is used as was done by Eggels [1994], where the radial velocity $u_r(0, \theta, z)$ at $r = 0$ is defined by:

$$u_r(0, \theta, z) = \frac{u_r(\Delta r, \theta, z) - u_r(\Delta r, \theta + \pi, z)}{2}, \qquad 0 \leq \theta < 2\pi, -\frac{L}{2} \leq z \leq \frac{L}{2} \tag{3.2}$$

Using a projection method, the set of non-linear momentum equations discretized in space can be denoted by:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + f(\rho^{n+\frac{1}{2}}, \mathbf{u}^*) + \frac{1}{\rho^{n-\frac{1}{2}}} G(p^{n-\frac{1}{2}}) = \mathbf{0}, \tag{3.3}$$

where all superscript indices represent time evaluations. Here G is the discrete gradient operator and f contains all remaining convective, viscous and body force terms. The pressure p, together with the LS function ϕ, the VoF function ψ and the density $\rho = \rho(\phi)$ are staggered in time, implying that they are formally evaluated at semi-levels $n \pm \frac{1}{2}$ in time. Equation 3.3 is linearized using Newton linearization and solved by a GMRES algorithm to find a tentative velocity $\mathbf{u}^*$. A Poisson equation is solved for an updated pressure $p^{n+\frac{1}{2}}$ by a CG algorithm with an Incomplete Cholesky preconditioner:

$$D\left(\frac{1}{\rho^{n+\frac{1}{2}}} G(p^{n+\frac{1}{2}})\right) = \frac{1}{\Delta t} D(\mathbf{u}^*) + D\left(\frac{1}{\rho^{n-\frac{1}{2}}} G(p^{n-\frac{1}{2}})\right), \tag{3.4}$$

where D is the discrete divergence operator. The presence of the divergence of the pressure at the previous semi-time step $n - \frac{1}{2}$ in the right hand side of the Poisson equation above is due to its presence in the momentum equation 3.3. For single phase flows, the pressure term on the right hand side of the Poisson equation would generally be moved to the left hand side to form (by linearity) an equation for a pressure *correction* $(p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}})$ instead of the pressure at the new time. This is however only possible if the density ρ remains constant in time, which is clearly not the case for multiphase flows. Finally, the tentative velocity $\mathbf{u}^*$ is made divergence-free using the newly found pressure $p^{n+\frac{1}{2}}$ through:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{1}{\Delta t}\left(\frac{1}{\rho^{n+\frac{1}{2}}}G(p^{n+\frac{1}{2}}) - \frac{1}{\rho^{n-\frac{1}{2}}}G(p^{n-\frac{1}{2}})\right) \tag{3.5}$$

which yields the new velocity vector $\mathbf{u}^{n+1}$.

3.1 Analysis of the temporal discretization

The set of semi-discrete momentum equations:

$$\frac{\partial \mathbf{u}}{\partial t} = -C(\mathbf{u}) + V(\mathbf{u}, \rho, \mu) - \frac{1}{\rho}G(p) + \mathbf{f}, \tag{3.6}$$

where $\mathbf{u}$ is the unknown velocity vector, $C(\mathbf{u})$ the convective part, $V(\mathbf{u}, \rho, \mu)$ the viscous part and $\mathbf{f}$ any remaining body force is integrated in time using the Implicit Midpoint rule, given by:

$$\mathbf{y}^{n+1} = \mathbf{y}^n + \Delta t\mathbf{k} \qquad \text{with} \qquad \mathbf{k} = f\left(t^n + \frac{\Delta t}{2}, \mathbf{y}^n + \frac{\Delta t}{2}\mathbf{k}\right) \tag{3.7}$$

for any ordinary differential equation $\mathbf{y}' = f(t, \mathbf{y})$. To this end, the first step is to solve the non-linear equation:

$$\mathbf{k} = -C\left(\mathbf{u}^n + \frac{\Delta t}{2}\mathbf{k}\right) + V\left(\mathbf{u}^n + \frac{\Delta t}{2}\mathbf{k}, \rho^{n+\frac{1}{2}}, \mu^{n+\frac{1}{2}}\right) - \frac{G(p^{n-\frac{1}{2}})}{\rho^{n-\frac{1}{2}}} + \mathbf{f}^n \tag{3.8}$$

Notice the inclusion of the pressure term at the previous time level. The viscous operator V is linear, so that we can rewrite the evaluation as a matrix-vector product:

$$V\left(\mathbf{u} + \frac{\Delta t}{2}\mathbf{k}, \rho^{n+\frac{1}{2}}, \mu^{n+\frac{1}{2}}\right) = (V\mathbf{u})(\rho^{n+\frac{1}{2}}, \mu^{n+\frac{1}{2}}) + \frac{\Delta t}{2}(V\mathbf{k})(\rho^{n+\frac{1}{2}}, \mu^{n+\frac{1}{2}}) \tag{3.9}$$

where V is now a matrix. The convective operator C however is non-linear, and Newton linearization is used to obtain two matrices C_0 and C_1 such that:

$$C(\mathbf{u}) = C_0 + C_1\mathbf{u} + \mathcal{O}(\Delta t^2), \tag{3.10}$$

where the right hand side again contains only a matrix-vector product. Hence, the convective term can by linearity be written as:

$$\begin{aligned} C\left(\mathbf{u} + \frac{\Delta t}{2}\mathbf{k}\right) &= C_0 + C_1\left(\mathbf{u} + \frac{\Delta t}{2}\mathbf{k}\right) + \mathcal{O}(\Delta t^2) && (3.11) \\ &= C_0 + C_1\mathbf{u} + \frac{\Delta t}{2}C_1\mathbf{k} + \mathcal{O}(\Delta t^2) && (3.12) \end{aligned}$$

With this second order linearization, expression 3.8 is approximated by:

$$\begin{aligned}\left(I+\frac{\Delta t}{2}C_1-\frac{\Delta t}{2}V(\rho^{n+\frac{1}{2}},\mu^{n+\frac{1}{2}})\right)\mathbf{k} &= -C_0-C_1\mathbf{u}^n+(V\mathbf{u}^n)(\rho^{n+\frac{1}{2}},\mu^{n+\frac{1}{2}}) \\ &\quad -\frac{G(p^{n-\frac{1}{2}})}{\rho^{n-\frac{1}{2}}}+\mathbf{f}^n \qquad (3.13)\end{aligned}$$

This linear system is solved using the GMRES algorithm, after which scaled addition to the previous velocity vector yields the predictor velocity $\mathbf{u}^*$. One could choose to further iterate on $\mathbf{k}$ for increased stability, but experience has shown no significant improvement while computational cost increases profoundly since each iteration adds an additional non-linear solve per time step. Therefore equation 3.13 is solved only once and the solution of the linearized system is assumed to be close enough to the solution of the non-linear system of equation 3.8.

3.2 Validation of the time integration method

The implementation of the time integration method is validated using an analytical solution of the viscous flow equations known as the Lamb-Oseen vortex. It represents an axisymmetric rotational flow that decays through viscous dissipation. The analytical velocity profile is given by:

$$u_r(r,z,t)=0, \qquad u_\theta(r,z,t)=\frac{\Gamma}{2\pi r}\left(1-e^{-\frac{r^2}{4\nu t}}\right), \qquad u_z(r,z,t)=0, \quad (3.14)$$

where Γ is the circulation and ν the kinematic viscosity. From the velocities it follows immediately that the flow is divergence-free, and the pressure p satisfies:

$$\frac{1}{\rho}\frac{\partial p}{\partial r}=\frac{u_\theta^2}{r} \qquad (3.15)$$

Since the velocity at $t=0$ is unbounded near $r=0$, the simulation is set to start at $t=2$ seconds when the velocity profile is smooth and bounded. The problem is solved on the domain $\{0\leq r\leq 1,-0.5\leq z\leq 0.5\}$ discretized uniformly by 19×19 computational cells. The initial conditions of the flow field are given by:

$$u_r(r_i,z_k,0)=0, \qquad u_\theta(r_i,z_k,0)=\frac{1}{2\pi r_i}\left(1-e^{-\frac{r_i^2}{4\cdot 10^{-3}\cdot 2}}\right), \qquad u_z(r_i,z_k,0)=0, \quad (3.16)$$

for a flow with viscosity $\nu = 10^{-3}$ m^2/s and circulation $\Gamma = 1$ m^2/s. The Dirichlet boundary conditions for the velocity are the exact solution values, while for the pressure homogeneous Neumann conditions exist at $z=-0.5$ and $z=0.5$, and the non-homogeneous Neumann condition of expression 3.15 at $r=1$.

The results are calculated with different values of the time step Δt and compared at time $t=27$ seconds. In addition, a reference solution $\mathbf{u}_{ref}$ is calculated with a much smaller time step. Before comparing, the reference solution is subtracted from all solutions to remove any significant contribution of the error in spatial discretization. Figure 3.1 shows the supremum- and ℓ_2-norm of both the velocity

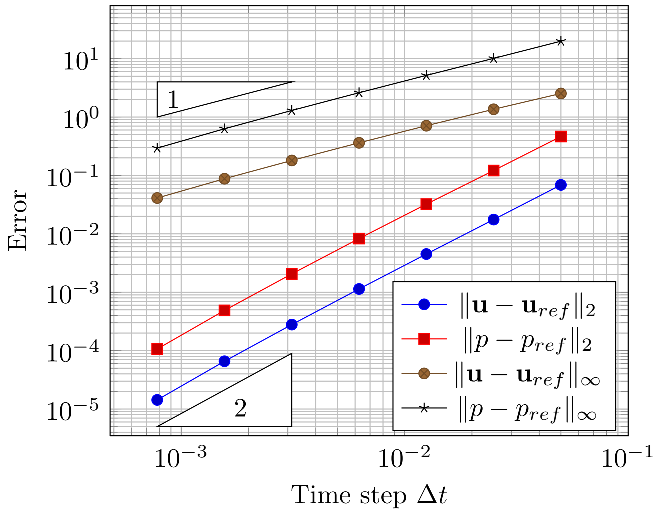


Figure 3.1: RMS- and maximum error of the velocity and the pressure field.

and the pressure field as a function of the magnitude of the time step Δt. The norms are defined as:

$$\|p-p_{ref}\|_2^2 = \sum_{i,j,k} r_i \Delta r_i \Delta\theta_j \Delta z_k |p_{ijk}-p_{ref_{ijk}}|^2, \qquad \|p-p_{ref}\|_\infty = \sup_{i,j,k} |p_{ijk}-p_{ref_{ijk}}| \tag{3.17}$$

The results show second order accuracy for the velocity in the ℓ_2-norm, which corresponds with the theoretical accuracy of the Implicit Midpoint method. The pressure also appears to be second order accurate in this norm, although there is little theoretical basis to justify this. For general flows though, the pressure should have at least first order accuracy. In the supremum norm, both velocity and pressure show first order convergence. It is likely that disturbances occur near the polar axis at $r = 0$, which are suppressed by the jacobian in the ℓ_2-norm but dominate the supremum norm.

3.3 Summary

A description was given of the discretization in time of the Navier Stokes equations. A projection scheme is used, where the Implicit Midpoint method is selected to calculate a tentative velocity field. The non-linear equation is linearized using second order Newton linearization, and subsequently solved using a GMRES algorithm. Then a Poisson equation is solved to update the pressure field, where a preconditioned CG solver is used, and the result is used to create a new divergence-free velocity field. The simulation of a Lamb-Oseen vortex reveals that both velocity and pressure are of order $\mathcal{O}(\Delta t^2)$ when evaluated in the ℓ_2-norm, but only $\mathcal{O}(\Delta t)$ is achieved in the supremum norm. Note that the considered test problem represents single phase flow only, i.e. there is no interface with a pressure jump. Test cases with two-phase flow will be considered in the next chapter.

Chapter 4

Test case 1: a stationary bubble with high surface tension

In the previous section, the velocity field was validated as a single-phase flow without any contact interface. In multi-phase flows on the other hand, the presence of a second fluid induces complex behaviour of the flow variables at the interface. Formal jump conditions for incompressible flows with an infinitessimally thin interface can be derived [Liu et al., 2000], which are sharply modelled by the GFM. This test case is meant to validate the implementation of these interface conditions on a stationary bubble that is subjected to a high, uniform surface tension. Results will be compared with Denner and van Wachem [2014].

4.1 The presence of parasitic currents

A confined domain is assumed with two fluid phases in equilibrium separated by a sharp interface. No gravity is present, and both the location and the shape of the interface remain stationary through a pressure jump $[p] = p_0 - p_1$ that is balanced by surface tension through the Young-Laplace relation:

$$[p] = \sigma\kappa, \tag{4.1}$$

where σ is the surface tension coefficient and κ the local interface curvature. With zero initial velocities, the configuration described above is a stationary solution of the Navier Stokes equations. For these initial conditions, the momentum equations reduce to:

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{1}{\rho}\nabla p = \mathbf{0}, \qquad \mathbf{u}(\mathbf{x}, 0) = \mathbf{0} \tag{4.2}$$

together with the interface jump condition equation 4.1. To guarantee stationary behaviour of the solution at the discrete level, it follows from equation 4.2 that the numerical treatment of the pressure term is pivotal: if at any point in time the pressure term is not identically zero, the momentum equations induce velocities which are commonly called 'parasitic' or 'spurious currents'. Their existence is the result of several purely numerical errors.

- The jump condition (4.1) requires the calculation of the local curvature κ of the interface. In LS methods, this quantity follows from the finite difference discretization of:

  $$\kappa = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \tag{4.3}$$

  where ϕ is the LS function. Clearly, the discretization will contain a truncation error.

- In the finite difference context, curvature values are often calculated at fixed locations (usually at the cell centers). However, most surface tension models require the actual curvature *at* the interface, and therefore this value has to be approximated. It seems that there is no universally agreed upon way to do this approximation yet as it also depends on the surface tension model used. In the GFM however, the curvature is required at the interface, and it is obtained by linear interpolation in the coordinate directions. This generally introduces an interpolation error.

- The momentum equations can be considered as a dynamical system:

  $$\frac{\partial \mathbf{u}}{\partial t} = f(\mathbf{u}, p, \phi). \tag{4.4}$$

  For a generic discretization of the system it is not a priori clear what level of stability the resulting system will possess. Initially, the system is given by expression 4.2, but any erroneously induced velocity after one time step complicates the system since the convective and diffusive parts of the momentum equations start contributing. This signifies the importance of the initialization of the pressure field p and the treatment of the pressure jump $\sigma\kappa$ at the interface. One could define the pressure throughout the domain if its values are known a priori, which is generally not the case. A different option is to calculate its values by solving a Poisson equation like equation 3.4, and in the MCLS method this is done using a PCG solver. Since it is generally too expensive to solve up to machine precision, a residual value is used as stopping criterium, and therefore it can not be expected that the pressure term is identically zero in this case: even though perhaps small initially, in time its contribution may grow and cause increasing parasitic velocities.

Hence, the major errors come from the discretization and interpolation of the curvature, and from the pressure field initialization. These sources will be investigated further.

4.2 Computational results

We will consider both two- and three-dimensional cases, which will be computed in polar and cylindrical coordinates respectively. The aim of the test is to validate the implementation of the surface tension model (GFM) by investigating the discretized equation 4.2.

4.2.1 A stationary interface in 2D

We start by considering a two-dimensional case of a circular bubble in a polar domain. The bubble has radius $R = 1$ and the domain is:

$$\{(r, \theta) \mid 0 \leq r \leq 2, 0 \leq \theta < 2\pi\}. \tag{4.5}$$

The LS function ϕ is initialized as:

$$\phi(r, \theta) = -1 + r, \tag{4.6}$$

and the circle contains fluid 0 and is surrounded by fluid 1. The density ratio ρ_0/ρ_1 can vary, but for simplicity the viscosity ratio μ_0/μ_1 is kept constant at 1.

Curvature discretization error As the LS function is linear in polar coordinates, it should hold that the numerical values of the right hand side of equation 4.3 (calculated in the cell centers) are exact up to machine precision due to vanishing discretization error (central differences). To verify this, we consider the error $\epsilon(N)$:

$$\epsilon(N) = \left\| \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \Big|_{numerical} - \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \Big|_{exact} \right\|_{L_1} \tag{4.7}$$

where N is the degrees of freedom. Since the exact value is $1/r$, the error is calculated as:

$$\epsilon(N) = \sum_{j=1}^{N_\theta} \sum_{i=1}^{N_r} \left| \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \Big|_{i,j} - \frac{1}{r_i} \right| r_i \Delta r_i \Delta \theta \tag{4.8}$$

It is expected that:

1. the error ϵ is not dependent on the number of degrees of freedom (in the absence of truncation error).
2. the magnitude of the error is around machine precision.

Figure 4.1 shows the behaviour of ϵ for increasing degrees of freedom. It is clear that indeed the magnitude of ϵ lies near machine precision and exhibits no significant dependence on N.

Curvature interpolation error The GFM requires the curvature κ at the interface to determine the pressure jump. The interface curvature is determined by interpolation of curvature values between cell centers in the coordinate direction to the interface location. The values of the LS function ϕ act as indicator of the presence of an interface; if $\phi(x_i) \cdot \phi(x_{i+1}) < 0$ then it must follow that an interface lies in the interval (x_i, x_{i+1}). Using linear interpolation of the LS function, the interface location is found as:

$$\frac{x}{\Delta x} = \frac{|\phi_i|}{|\phi_i| + |\phi_{i+1}|}, \qquad 0 < x < \Delta x, \tag{4.9}$$

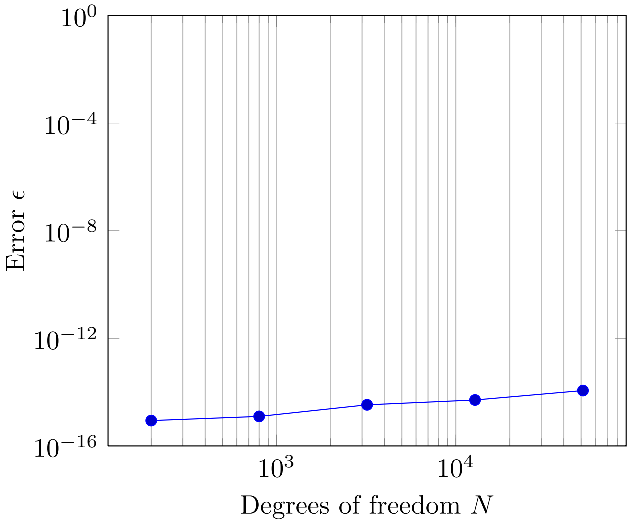


Figure 4.1: Accuracy of curvature discretization.

so that the curvature in this interval can be calculated as:

$$\kappa = \frac{\kappa_i |\phi_{i+1}| + \kappa_{i+1} |\phi_i|}{|\phi_i| + |\phi_{i+1}|} \tag{4.10}$$

where $\phi_i = \phi(x_i)$ for ease of notation. The procedure is repeated analogously in the other coordinate directions. Given the intial LS function as in equation 4.6, it should follow that because of its linearity in the radial coordinate the calculated location of the interface is exact: this can be verified by substitution into equation 4.9. However, since this expression is only implicitly used to derive equation 4.10, the calculated curvature will not be exact as linear interpolation of the true solution:

$$\kappa(r) = \frac{1}{r} \tag{4.11}$$

has an interpolation error of $\mathcal{O}(1/N^2)$. To verify the interpolation of the curvature, and therefore the accuracy of the pressure jump at the interface, the following error is considered:

$$\epsilon(N) = \left\| \sigma\kappa|_{numerical} - \sigma\kappa|_{exact} \right\|_\infty , \tag{4.12}$$

where $\sigma\kappa|_{numerical}$ is the calculated pressure jump according to equation 4.10 but using *exact* values for the cell centered curvatures κ_i so that the error is dominated by the interpolation error. The exact jump is calculated as:

$$\sigma\kappa|_{exact} = \frac{\sigma}{R} \tag{4.13}$$

Figure 4.2 confirms that the error in the calculated pressure jump due to interpolation is $\mathcal{O}(1/N^2)$ as expected. Repeating the test using *calculated* values for the cell centered curvatures κ_i gives virtually indistinguishable results, which is to be expected from the results of the previous section.

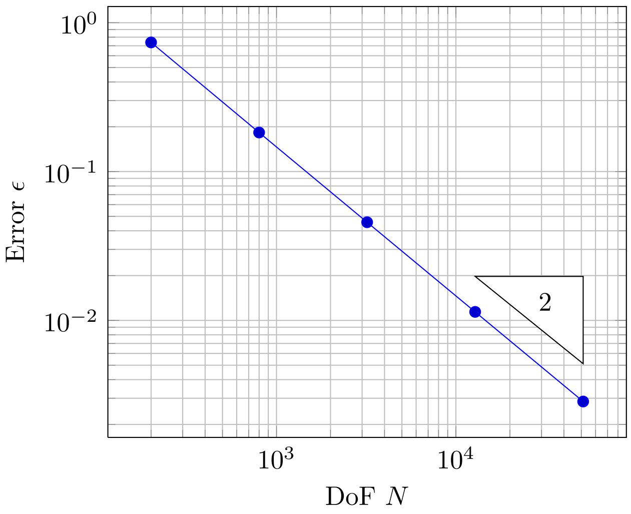


Figure 4.2: Accuracy of curvature interpolation.

Force balance error Focusing on the momentum equation (4.2) now, the question arises of how to initialize the pressure field p. As mentioned, the pressure field needs to be such that:

$$\frac{1}{\rho}\nabla p = \mathbf{0}, \qquad [p] = \sigma\kappa \tag{4.14}$$

holds numerically as to prevent velocities from originating. Expression 4.14 is occasionally called the initial 'force balance' for a stationary solution (although usually in the context of the CSF model, see Denner and van Wachem [2014] and Meland et al. [2007]): the pressure difference should be balanced by the surface tension. The importance of a correct force balance stems from the experimental observation that existing parasitic velocities are very difficult to prevent from growing in time, let alone be suppressed by the numerical model: it appears that keeping the velocities bounded in time is at present the best one can do. Because of this observation, one should at least aim to prevent their existence during initialization of the flow field.

The easiest way to initialize p is to provide the pressure values, but this of course requires knowledge of the initial pressure field. This can often only be done for relatively simpel geometries. For more general cases, the initial pressure can be calculated by solving the Poisson equation 3.4. Assuming an initial divergence-free velocity field, one further neglects the pressure term at the previous time so that the pressure follows from the Laplace equation:

$$\nabla \cdot \left(\frac{1}{\rho^{\frac{1}{2}}} \nabla p^{\frac{1}{2}} \right) = 0 \tag{4.15}$$

with appropriate boundary conditions (generally homogeneous Neumann, or non-homogeneous Neumann in case gravity is present). Notice that due to the staggering in time of the computational quantities, the pressure p and the density ρ are formally evaluated at semi-time steps while the velocities are evaluated at integer time steps. A serious drawback of this approach, however, is the fact that the numerical solution of equation 4.15 is obtained using only finite precision (depending on the tolerance chosen for the iterative solution algorithm), and therefore the accuracy of equation 4.14 is uncertain. It can be argued however that this accuracy is highly dependent on the solver accuracy. To validate the accuracy of the initial force balance, the following tests are considered (based on Denner and van Wachem [2014]):

1. To validate the global implementation of the projection method described in section 3, both the initial pressure $p(\mathbf{x}, 0)$ and the pressure jump $\sigma\kappa$ are prescribed as:

$$p(\mathbf{x}, 0) = \begin{cases} 73, & \phi(\mathbf{x}, 0) \leq 0 \\ 0, & \phi(\mathbf{x}, 0) > 0 \end{cases}, \qquad \sigma\kappa = 73 \cdot \frac{1}{2} = 36.5 \tag{4.16}$$

   The latter assures that through the GFM both:

$$\frac{1}{\rho}\nabla p = \mathbf{0} \qquad \text{and} \qquad \nabla \cdot \left(\frac{1}{\rho}\nabla p\right) = 0 \tag{4.17}$$

   are satisfied identically, and so no solver action is expected from the program since the numerical solution is satisfied by the initial conditions.

2. The ultimate test is to calculate the initial pressure field using the Poisson equation 4.15, and calculate the pressure jump $\sigma\kappa$ based on the initial LS function. This is the procedure followed in a general situation without any knowledge of the initial pressure field. For a fixed number of degrees of freedom, the accuracy of the Poisson solver should dominate the error. In Denner and van Wachem [2014] it is suggested that the magnitude of the parasitic currents may be a function of the density ratio ρ_0/ρ_1 or the time step Δt, and so these are also varied between simulations.

The first test case in which both the initial pressure and the jumps are prescribed results in no solver action in the predictor equation nor in the pressure Poisson equation, as was expected. This implies that for exact initial conditions, the algorithm performs as expected, and after one time step a velocity magnitude of $\|\mathbf{u}^1\|_{L_\infty} = 0.0$ is found.
For the second test case, the values of $\|\mathbf{u}^1\|_{L_\infty}$ and the initial pressure difference error:

$$\epsilon(\Delta p) = \left| \frac{\Delta p - \Delta p_{exact}}{\Delta p_{exact}} \right| \tag{4.18}$$

are calculated with the following parameters:

$$N = 800, \quad \Delta p_{exact} = 36.5 \tag{4.19}$$

Table 4.1 shows the magnitude of the velocity and pressure difference after one time step when the pressure is calculated by solving a Poisson equation. In this case, the value of the initial jump $\sigma\kappa$ is provided and is exact. With a relatively low solver tolerance ($\approx 10^{-12}$), this yields very small results, for both time steps and for all density ratios.

Table 4.1: Velocity and pressure difference after one time step with exact pressure jump.

	$\|\mathbf{u}^1\|_{L_\infty}$		$\|\epsilon\|_{L_\infty}$	
ρ_0/ρ_1	$\Delta t = 10^{-3}$ s	$\Delta t = 10^{-6}$ s	$\Delta t = 10^{-3}$ s	$\Delta t = 10^{-6}$ s
10^0	$7.689 \cdot 10^{-15}$	$7.689 \cdot 10^{-18}$	$1.225 \cdot 10^{-12}$	$1.224 \cdot 10^{-12}$
10^3	$1.749 \cdot 10^{-17}$	$1.743 \cdot 10^{-20}$	$3.504 \cdot 10^{-15}$	$3.309 \cdot 10^{-15}$
10^6	$2.539 \cdot 10^{-19}$	$2.539 \cdot 10^{-22}$	$2.920 \cdot 10^{-15}$	$2.920 \cdot 10^{-15}$
10^9	$1.887 \cdot 10^{-22}$	$1.887 \cdot 10^{-25}$	$3.893 \cdot 10^{-16}$	$3.893 \cdot 10^{-16}$

Table 4.2 shows the results of the originally implemented algorithm without any exact input values. The pressure jump is now calculated, and this has a significant effect on the resulting pressure difference. Clearly the inaccuracy of the curvature determination starts to dominate the error now. The parasitic velocities on the other hand stay very small, at about the same magnitude as with prescribed pressure jump. There appears to be a linear dependency on the density ratio, in both the magnitude of the induced velocities and pressure difference.

Table 4.2: Velocity and pressure difference after one time step with calculated pressure jump.

	$\|\mathbf{u}^1\|_{L_\infty}$		$\|\epsilon\|_{L_\infty}$	
ρ_0/ρ_1	$\Delta t = 10^{-3}$ s	$\Delta t = 10^{-6}$ s	$\Delta t = 10^{-3}$ s	$\Delta t = 10^{-6}$ s
10^0	$1.294 \cdot 10^{-14}$	$1.298 \cdot 10^{-17}$	$2.506 \cdot 10^{-3}$	$2.506 \cdot 10^{-3}$
10^3	$7.735 \cdot 10^{-17}$	$7.735 \cdot 10^{-20}$	$2.506 \cdot 10^{-3}$	$2.506 \cdot 10^{-3}$
10^6	$2.651 \cdot 10^{-19}$	$2.651 \cdot 10^{-22}$	$2.506 \cdot 10^{-3}$	$2.506 \cdot 10^{-3}$
10^9	$1.848 \cdot 10^{-22}$	$1.848 \cdot 10^{-25}$	$2.506 \cdot 10^{-3}$	$2.506 \cdot 10^{-3}$

4.2.2 A stationary interface in 3D

In the 3D cylindrical coordinate system:

$$\left\{ (r, \theta, z) \;\middle|\; 0 \leq r \leq 2, 0 \leq \theta < 2\pi \; -\frac{L}{2} \leq z \leq \frac{L}{2} \right\}, \tag{4.20}$$

the LS function associated to a sphere with radius $R = 1$ centered around the origin is given by:

$$\phi(r, \theta, z) = -1 + \sqrt{r^2 + z^2}. \tag{4.21}$$

The LS function in 3D is no longer linear in its coordinates and therefore there will be an additional truncation error from the curvature discretization that was absent in the 2D circular case. To analyse the behaviour of parasitic currents in 3D, the test case of table 4.2 without any prescibed or exact values is repeated. The results are shown in table 4.3.

The results are similar to the 2D case, with a constant magnitude for the pressure difference, but the dependence on the density ratio makes that for smaller density differences the velocities become significant.

Table 4.3: Velocity and pressure difference after one time step with calculated pressure jump.

	$\|\mathbf{u}^1\|_{L_\infty}$		$\|\epsilon\|_{L_\infty}$	
ρ_0/ρ_1	$\Delta t = 10^{-3}$ s	$\Delta t = 10^{-6}$ s	$\Delta t = 10^{-3}$ s	$\Delta t = 10^{-6}$ s
10^0	$1.741 \cdot 10^{-5}$	$1.741 \cdot 10^{-8}$	$1.107 \cdot 10^{-3}$	$1.107 \cdot 10^{-3}$
10^3	$5.397 \cdot 10^{-7}$	$5.397 \cdot 10^{-10}$	$1.133 \cdot 10^{-3}$	$1.133 \cdot 10^{-3}$
10^6	$5.573 \cdot 10^{-10}$	$5.573 \cdot 10^{-13}$	$1.133 \cdot 10^{-3}$	$1.133 \cdot 10^{-3}$
10^9	$5.573 \cdot 10^{-13}$	$5.573 \cdot 10^{-16}$	$1.133 \cdot 10^{-3}$	$1.133 \cdot 10^{-3}$

4.3 Summary

Parasitic currents are (and have been) a recurring problem in both LS and VoF methods. Their existence is caused by several reasons, but in general the inaccuracy in the curvature calculation is the most contributing factor. Using a systematic approach, it is shown that curvature approximation and interpolation are both second order accurate. In addition, through the simulation of a stationary bubble (of which the exact curvature is known) it is shown that the MCLS method in combination with the GFM deals very well with spurious velocities.

Chapter 5

Test case 2: a rising bubble in a vertical circular pipe

This test case comprises the motion caused by buoyancy of an initially spherical inclusion through a cylinder filled with fluid. A typical physical example is a rising gas bubble in a water column. Little theoretical data has been derived for this case; most results concern the shape and terminal rise velocity of the bubble [Joseph, 2003, Bozzano and Dente, 2001]. Therefore, validation of computations is generally achieved through comparison with experimental results.

5.1 Important flow parameters

The behaviour of the rising gas bubble is characterized by various flow parameters, including the Eötvos number Eo (also known as the Bond number), the Morton number Mo and the Reynolds number Re. For this geometry, they are defined as:

$$Eo = \frac{\rho_l g D^2}{\sigma}, \qquad Mo = \frac{g\mu_l^4}{\rho_l \sigma^3}, \qquad Re = \frac{\rho_l D|u|}{\mu_l}, \tag{5.1}$$

where ρ_l and μ_l are the density and viscosity of the liquid phase respectively, g the gravitational acceleration, $D = 2r_0$ the initial bubble diameter, σ the surface tension coefficient and u the terminal rise velocity. An experimental relation between these three parameters for a rising bubble was given by Grace [1973] (see figure 5.1).

5.2 Computational results

Initially, the results from Denner and van Wachem [2014], van Sint Annaland et al. [2005] and Mukundakrishnan et al. [2007] were used for comparison because of the detailed description of the test case and the clear presentation of the obtained results. However, a sensitivity analysis revealed that the influence of the rectangular channel walls used in all three of the studies made the comparison with the cylindrical MCLS results invalid due to the dominating wall effects. To obtain a physically comparable situation (if at all possible), it is

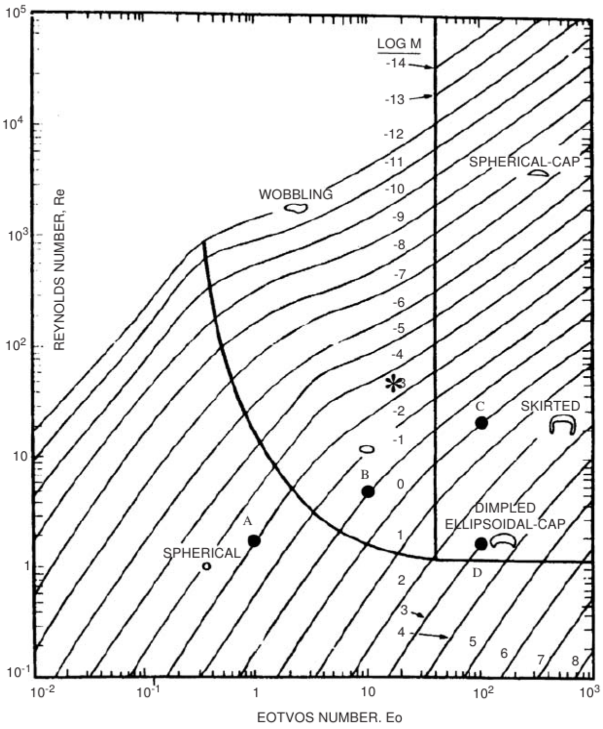


Figure 5.1: Grace chart relating bubble shape and Eotvos, Morton and Reynolds numbers. From Grace [1973].

a priori unclear what the associated cylinder radius R should be for a given channel width. Especially for relatively narrow geometries, choosing R as the radius of the circumscribed circle or as the radius of the inscribed circle generally gives a significantly different terminal rise velocity, with the channel result lying somewhere in between. Based on this observation, different (and less detailed) validation material was used where the domain consists of a cylinder [Cirilo and Castelo, 2011], [Liu et al., 2013], [Chakraborty et al., 2013].

Following Chakraborty et al. [2013], we consider a spherical bubble of radius $r_0 = 0.5\ m$ in an axi-symmetrical domain of radius $R = 4\ m$ and length $H = 8\ m$. The centre of the bubble is initially located at four bubble radii from the bottom. Five cases are considered with different combinations of Eötvos and Morton numbers, and the simulated results are compared with the results from Chakraborty et al. [2013] and experimental results from Bhaga and Weber [1981]. There is a time step constraint due to the surface tension:

$$\Delta t \leq \sqrt{\frac{(\rho_l + \rho_g) ds^3}{4\pi\sigma}}, \tag{5.2}$$

where ds is the length of the largest face of a computational cell. The flow is initially at rest, and a no-slip condition for the velocity holds on all walls. The LS function ϕ is initialized as:

$$\phi(r, z) = -\frac{1}{2} + \sqrt{r^2 + (z-2)^2}. \tag{5.3}$$

The quantity of interest is the terminal Reynolds number Re as defined in expression 5.1. Table 5.1 shows the different test cases that represent different regions of the Grace chart in figure 5.1. The column headed by Re_{exp} contains the experimental results from Bhaga and Weber [1981], and the column headed by Re_{ref} contains the numerical results from Chakraborty et al. [2013] (presumably on 200 × 400 or 400 × 800 grids). The last column contains the numerical results of the MCLS method.

Table 5.1: Terminal rise velocities for five difference bubble cases.

Bubble regime	Eo	Mo	Re_{exp}	Re_{ref}	Re_{MCLS}
Spherical	0.971	$1.26 \cdot 10^{-3}$	1.8	1.74	1.65
Oblate ellipsoid	10	$1.0 \cdot 10^{-2}$	11.0	10.8	11.1
Oblate ellipsoid disk-like	32.2	$8.2 \cdot 10^{-4}$	55.3	53.4	51.0
Oblate ellipsoidal cap	116	1.31	20.4	19.5	18.9
Spherical cap	39.6	$6.5 \cdot 10^{-2}$	19.6	19.0	19.1

The MCLS results were obtained for two grids, one with 100 × 200 cells and the second containing 200 × 400 cells. The time step satisfies expression 5.2 for both grids. The results shown are from the 200 × 400 cells computation.

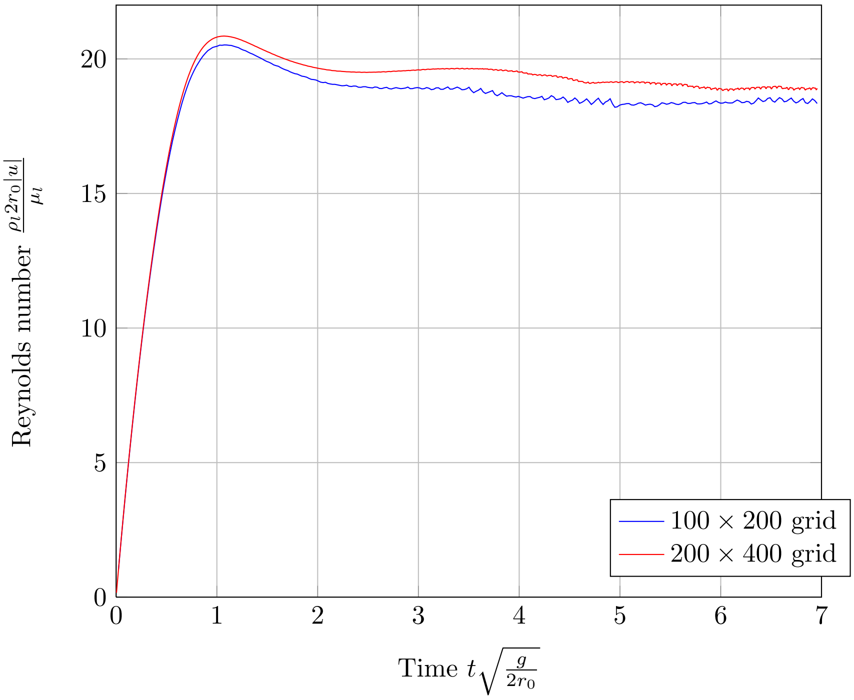


Figure 5.2: Reynolds number in time for the oblate ellipsoidal cap case. Notice the decrease in amplitude of the oscillations with grid refinement.

The rise velocity $|u|$ is determined through:

$$|u|(t) = \frac{\int_\Omega w(\mathbf{x},t)\psi(\mathbf{x},t)d\mathbf{x}}{\int_\Omega \psi(\mathbf{x},t)d\mathbf{x}} \approx \frac{\sum_i w_i\psi_i r_i \Delta r_i}{\sum_i \psi_i r_i \Delta r_i}, \tag{5.4}$$

with $w(\mathbf{x},t)$ the continuous velocity component in the direction of the gravity vector, w_i its discrete representation in cell i, ψ_i the VoF value in cell i and Ω the computational domain.

The MCLS results occasionally show a slight oscillation in the rise velocity in time (see figure 5.2). The oscillations, however, stay bounded, and the amplitude decreases with refinement in space and time. The phenomenon is more pronounced in cases of high surface tension, and further investigation is needed to explain this behaviour. Many similar methods also exhibit oscillatory rise velocities. Regardless of this effect, table 5.1 shows that the MCLS results are very close to the numerical results of Chakraborty et al. [2013] and to the experimental results.

5.3 Summary

The rising bubble test case shows that the MCLS method is very well capable of simulating a large variety of gas bubbles in a stagnant column. Both high- and low surface tension cases are predicted with reasonable accuracy, which validates the implementation of the surface tension algorithm. This test also shows the importance of the validation material: the domain geometries have to be exactly equal to objectively compare the resulting physical behaviour of the bubbles, especially for relatively narrow domains. But even for relatively wide domains one has to be cautious, as the literature provides different rules-of-thumb for the bubble radius to domain width ratio for which the walls are no longer considered to have influence on the terminal rise velocity.

Chapter 6

Test case 3: Taylor bubble

The final test case considered comprises the rise of a long gas bubble through a cylindrical pipe known as a Taylor- or Dumitrescu bubble. In practise this configuration is often found in vertical risers, and it is therefore 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. Also of interest are the thickness of the falling film between the bubble and the pipe wall [Llewellin et al., 2012], and the wake of the bubble [Kang et al., 2010, Bugg and Saad, 2002]. The simulations presented in this section were carried out with the axi-symmetric version of the code.

6.1 Analytical results for the rise velocity

From as early as 1940 on, theoretical research has yielded several succesful approaches for the construction of a model that predicts the rise velocity of long bubbles in stagnant liquids. Dumitrescu [1943] derived under certain simplified conditions that the Froude number Fr, defined as:

$$Fr = \frac{u_t}{\sqrt{gd}} \tag{6.1}$$

where u_t is the terminal rise velocity, g the gravitational acceleration and d the reference length, should remain constant. He found a value of $Fr = 0.351$. Others have found values of $Fr = 0.328$ [Davies and Taylor, 1950] and $Fr = 0.303$ [Brown, 1965] respectively [Kang et al., 2010]. In the literature the analytical solution of Dumitrescu is considered to be very accurate, in the limit of small liquid viscosity, small gas density, and small surface tension.

6.2 Computational results

A bubble is initially placed at a quarter length of a closed pipe segment with length-to-diameter ratio of 8 : 1. The walls are all no-slip, and the requirement 5.2 determines appropriate values for the magnitude of the time step Δt. It was demonstrated in Kang et al. [2010] that variation in surface tension has a much more profound effect on the rise velocity and tail shape than variation in density- and viscosity ratios. Therefore, to verify the correct simulation of a

Taylor bubble, four cases are considered in which the surface tension coefficient σ varies. Common dimensionless parameters are the Eötvos number, the Reynolds number and the variable of interest, the Froude number, which represents the dimensionless rise velocity. The former two are defined as:

$$Eo = \frac{\rho_l g D^2}{\sigma}, \qquad Re = \frac{\rho_l D|u|}{\mu_l}, \tag{6.2}$$

where ρ_l and μ_l are the density and viscosity of the liquid phase respectively, g the gravitational acceleration, $D = 2r_0$ the initial bubble diameter, σ the surface tension coefficient and u the terminal rise velocity. Table 6.1 summarizes the case parameters in decreasing surface tension magnitude. Note that the Eotvos number and the Reynolds number are input parameters for the simulation, whereas the Froude number is the result of the simulation (i.e. the scaled bubble velocity). The simulations were done on a single grid of 20 × 160 cells without much refinement because of robustness issues.

Table 6.1: Parameters of the four cases in figure 6.1.

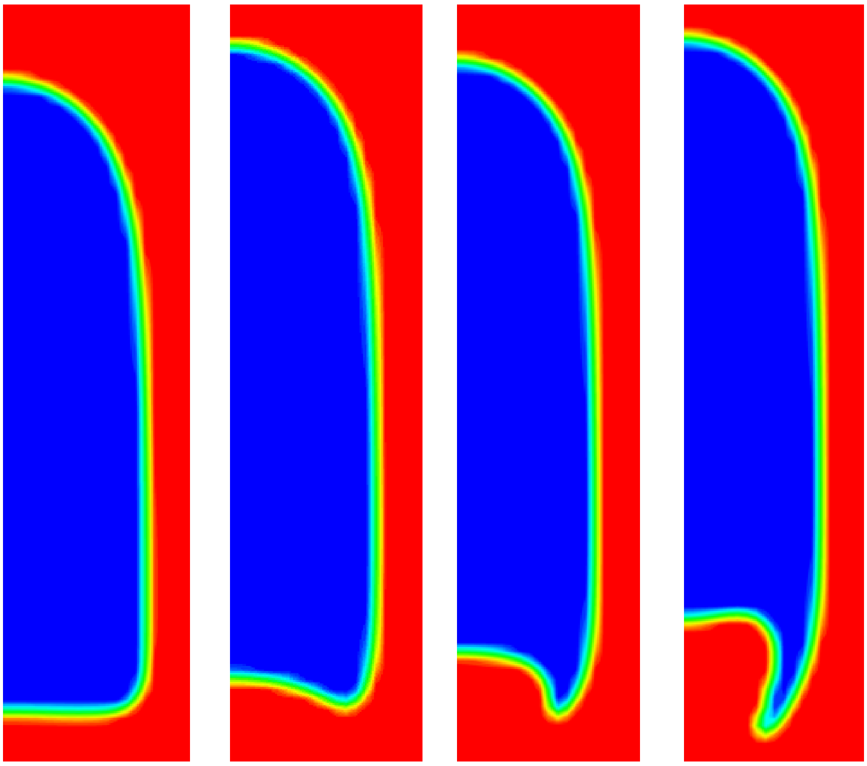
case	grid	Eo	Re	Fr
1	20 × 160	40	$1 \cdot 10^{-3}$	0.273
2	20 × 160	100	$1.56 \cdot 10^{-2}$	0.325
3	20 × 160	200	$1.25 \cdot 10^{-1}$	0.334
4	20 × 160	400	$1 \cdot 10^{2}$	0.338

Even though the theoretical values from section 6.1 do not incorporate the effect of surface tension, it seems that a small amount of surface tension is necessary to produce a stable bubble. Table 6.1 shows that for decreasing surface tension, the Froude number moves towards the theoretical value of 0.351.

The reduction of surface tension has a significant effect on the shape of the bubble, in particular the lower end. Figure 6.1 shows a snapshot of the four cases. With relatively high surface tension, the bubble has a rather flat or even convex tail. As the magnitude of the surface tension decreases, the lower end becomes more and more concave, and at some point a phenomenon close to a Taylor-Rayleigh instability can be observed protruding from below. This eventually leads to break up of some small pockets of the bubble, but with too little surface tension, the core of the bubble may even get penetrated and a complete breakup follows.

6.3 Summary

The axi-symmetric version of the MCLS code is quite convenient to calculate appropriate test cases like the Taylor bubble, that generally require relatively fine meshes to capture the complex structure of the wake. Until now, only some basic investigation has been done on the capacity of the code to qualitatively simulate a Taylor bubble, which has turned out to be satisfactory. The resulting Froude numbers correspond well with theoretical values.



(a) Case 1.

(b) Case 2.

(c) Case 3.

(d) Case 4.

Figure 6.1: Snapshot of Taylor bubble for the four cases. Number of cells used is 20 × 160.

Chapter 7

Conclusions and recommendations

The new algorithm with the Mass Conserving Level Set (MCLS) method in cylindrical coordinates has been subjected to three dynamical test cases of increasing complexity to validate the implementation of various components and algorithms. In addition, the underlying flow field is assessed in a convergence study. It was shown that the time integration of the spatially discretized Navier Stokes equations is $\mathcal{O}(\Delta t^2)$ when considered in the ℓ_2-norm, which corresponds to the analytical estimate of the Implicit Midpoint method. When evaluated in the sup-norm, the convergence drops to $\mathcal{O}(\Delta t)$, most likely due to perturbations near the axis $r = 0$.

The implementation of the Ghost Fluid method (GFM) as surface tension model was validated by considering the key elements of the algorithm separately. Both the curvature calculation and interpolation are $\mathcal{O}(1/N^2)$ with N the amount of degrees of freedom. The tolerance of the Poisson solver plays only a minor role, as the inaccuracy in the pressure jump (i.e. the determination of the curvature at the interface) dominates the magnitude of spurious currents. With exact jump values, the MCLS code performs very well with velocities around machine precision. With calculated jump values, the error in pressure difference becomes much larger, and the velocities increase but (depending on the density ratio) remain at acceptable values. The overall performance of the surface tension model is therefore satisfactory, but improvements should include the determination of the curvature at the interface to obtain more accurate jump values.

The rising bubble case tests the ability of the MCLS algorithm to replicate a well-documented physical experiment. The main quantity of interest, which is the terminal rise velocity, is found to be very close to both reference calculations and experimental values for a large variety of bubble shapes. The results validate the implementation of the interface representation and the surface tension model of the MCLS computational code. The appearance of small oscillations for cases with high surface tension however requires further investigation to be able to trace their origins.

Also a single Taylor bubble in a vertical pipe has been simulated as a first step to the modelling of slug flow. Recognizing that the surface tension significantly influences the shape and stability of the bubble, four cases have been considered with fixed density and viscosity but with varying values of the surface tension. It is found that a stable bubble can be obtained for various combinations of Eötvos and Reynolds numbers. Partial breakup is observed with very small surface tension. The corresponding rise velocity of the bubble, expressed as the Froude number, approaches a theoretically predicted value for vanishing viscosity and surface tension.

Bibliography

Dahya Bhaga and ME Weber. Bubbles in viscous liquids: shapes, wakes and velocities. *Journal of Fluid Mechanics*, 105:61–85, 1981.

G Bozzano and M Dente. Shape and terminal velocity of single bubble motion: a novel approach. *Computers & chemical engineering*, 25(4):571–576, 2001.

JU Brackbill, Douglas B Kothe, and C1 Zemach. A continuum method for modeling surface tension. *Journal of computational physics*, 100(2):335–354, 1992.

RAS Brown. The mechanics of large gas bubbles in tubes: I. bubble velocities in stagnant liquids. *The Canadian Journal of Chemical Engineering*, 43(5): 217–223, 1965.

JD Bugg and GA 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.

I Chakraborty, G Biswas, and PS Ghoshdastidar. A coupled level-set and volume-of-fluid method for the buoyant rise of gas bubbles in liquids. *International Journal of Heat and Mass Transfer*, 58(1):240–259, 2013.

Eliandro R Cirilo and A Castelo. Numerical study of a gas bubble rising through stagnant fluid. In *Proceedings of the 21st International Congress of Mechanical Engineering, Natal, Brazil*, volume 20011, 2011.

RM Davies and Geoffrey Taylor. The mechanics of large bubbles rising through extended liquids and through liquids in tubes. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 200(1062):375–390, 1950.

Fabian Denner and Berend GM 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.

Dumitru Theodor Dumitrescu. Strömung an einer luftblase im senkrechten rohr. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 23(3):139–149, 1943.

Jacobus Gerardus Maria Eggels. Direct and large eddy simulation of turbulent flow in a cylindrical pipe geometry. 1994.

Stanley Fedkiw and Ronald Osher. Level set methods and dynamic implicit surfaces. 2003.

J.R. Grace. Shapes and velocities of bubbles rising in infinite liquids. *Trans. Inst. Chem. Eng.*, 51(2):116–120, 1973.

Cyril W Hirt and Billy D Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of computational physics*, 39(1):201–225, 1981.

Matthew Jemison, Eva Loch, Mark Sussman, Mikhail Shashkov, Marco Arienti, Mitsuhiro Ohta, and Yaohong Wang. A coupled level set-moment of fluid method for incompressible two-phase flows. *Journal of Scientific Computing*, 54(2-3):454–491, 2013.

Daniel D Joseph. Rise velocity of a spherical cap bubble. *Journal of Fluid Mechanics*, 488:213–223, 2003.

Chang-Wei Kang, Shaoping Quan, and Jing Lou. Numerical study of a taylor bubble rising in stagnant liquids. *Phys. Rev. E*, 81:066308, Jun 2010. doi: 10.1103/PhysRevE.81.066308.

Lili Liu, Hui Tang, and Shaoping Quan. Shapes and terminal velocities of a drop rising in stagnant liquids. *Computers & Fluids*, 81:17–25, 2013.

Xu-Dong Liu, Ronald P Fedkiw, and Myungjoo Kang. A boundary condition capturing method for poisson's equation on irregular domains. *Journal of Computational Physics*, 160(1):151–178, 2000.

EW Llewellin, E Del Bello, J Taddeucci, P Scarlato, and SJ 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.

R. Meland, I.R. Gran, R. Olsen, and S.T. Munkejord. Reduction of parasitic currents in level-set calculations with a consistent discretization of the surface-tension force for the csf model. In *16th Australasian Fluid Mechanics Conference (AFMC)*, pages 862–865, 2007.

Youhei Morinishi, Oleg V Vasilyev, and Takeshi Ogi. Fully conservative finite difference scheme in cylindrical coordinates for incompressible flow simulations. *Journal of Computational Physics*, 197(2):686–710, 2004.

Karthik Mukundakrishnan, Shaoping Quan, David M Eckmann, and Portonovo 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.

Mark Sussman and Elbridge Gerry 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.

SP 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.

M van Sint Annaland, NG Deen, and JAM Kuipers. Numerical simulation of gas bubbles behaviour using a three-dimensional volume of fluid method. *Chemical Engineering Science*, 60(11):2999–3011, 2005.