Emulsion Design. Analysis of Drop Deformations in Mixed Flows

Egholm, Runi Ditlev; Szabo, Peter

Emulsion Design
Analysis of Drop Deformations in Mixed Flows

Rúni Ditlev Egholm

2008

Ph.D. Thesis

DTU

TECHNICAL UNIVERSITY OF DENMARK
DEPARTMENT OF CHEMICAL ENGINEERING

ISBN 978-87-91435-73-0
Printed by Frydenberg a/s, Copenhagen, Denmark

Preface

The work presented in this thesis was carried out at the Danish Polymer Centre, Department of Chemical Engineering, Technical University of Denmark in the period from July 2004 to October 2007 under the supervision of Associate Professor Peter Szabo.

The subject of the thesis is the study of flow induced droplet deformations in liquid-liquid dispersions. These types of studies are relevant in connection with mechanical processing of emulsions. The work includes model implementation, numerical simulations and experiments.

The experimental part which is based on drop deformation studies in a rotor-stator device were carried out during an external research stay at the Laboratory of Food Process Engineering at ETH, Zürich, in the period from October 2005 to January 2006.

I wish to thank my supervisor Peter Szabo for his indispensable advice and support during this project.

Furthermore I wish to thank Dr. Peter Fischer from the Laboratory of Food Process Engineering at ETH for his advice on the rotor-stator device and for making my stay in Zürich a great experience. Also I wish to thank Professor Kathleen Feigl (Michigan Technological University, Houghton, MI) for the numerical flow field calculations associated with the rotor-stator device and for the Boundary Integral calculations which made the experiments much more coherent. I also wish to thank José Manuel Roman for his help on the use of the PARDISO linear solver and for many interesting and fruitful discussions on numerical methods in general.

Finally I acknowledge the financial support for this project given by The Danish Research Council for Technology and Production Sciences (FTP grant 26-03-0282).

Lyngby, October 2007

Rúni Ditlev Egholm

Abstract

The work presented in this thesis concerns numerical and experimental studies of flow induced deformation of drops suspended in a second and immiscible liquid.

In the numerical part a model is implemented which is based on a Finite Element (FE) Stokes solver coupled with a Volume of Fluid (VOF) tracking procedure. The FE solver is based on Q_2P_0 elements while the VOF procedure is based on PLIC (Piecewise Linear Interface Calculation) interface reconstruction and a split operator Lagrangian advection procedure which conserves mass rigorously. The model is fully 3D and can be used for simulating the transient behavior of two phase liquid systems with moving interface topologies. In order to include interfacial tension in the flow calculations both the Continuous Surface Stress (CSS) model of Lafaurie, Nardone, Scardovelli, Zaleski & Zanetti (1994) and the Continuous Surface Force (CSF) model of Brackbill, Kothe & Zemach (1992) are implemented. Due to the high interface curvatures associated with highly deformed drops it is necessary to use a high resolution mesh for our calculations. This leads to extensive computation times mainly due to factorization and back substitution of the discretized flow field equations. In order to reduce the computational cost a 2-level procedure is implemented where the fluid tracking algorithms are associated with a fine VOF mesh while the flow field variables are associated with a coarser FE mesh. In the 2-level algorithm the calculation of interfacial tension terms is carried out as a summation of contributions from the VOF mesh. This corresponds to letting the curvature vary within elements of the FE mesh.

The implemented model is tested in terms of spatial and temporal convergence by simulating the deformation of a single drop in a simple shear flow field. Furthermore wall effects are also investigated by varying the size of the computational domain which consists of a box with variable mesh size. In the center of the domain, where the drop resides, the mesh consists of a fine region whereas closer to the walls the elements gradually increase in size. Tests show that wall effects are negligible when the distance from a drop with initial radius r_0 to the domain boundaries is $24r_0$. In the spatial convergence tests the resolution of the fine mesh region is varied and it is found that a VOF mesh with side lengths $h_{vof} = r_0/18$ is adequate when the viscosity ratio, λ, between the drop and the continuous phase is one. More thorough tests are carried out both in simple shear and planar elongation. These simulations include dependence of steady-state deformations on the capillary number, drop-break and drop merging. Generally the test results agree well with results reported in the literature. However, simulations carried out for λ different from one indicate that the resolution of the FE mesh needs to be increased compared to simulations carried out with $\lambda = 1$. This is probably related to the method used for calculating the viscosity in elements which include both liquid phases.

In the experimental part of the thesis the deformation of a single drop suspended in liquid undergoing a complex dispersing flow is studied. The experimental setup is based on a rotor-stator device consisting of two concentric cylinders with teethed walls. In order to monitor the drop deformation and drop position a twin camera system is applied. In the subsequent data analysis the recorded movies are analysed using an automated image analysis procedure which leads to the deformation history of the drop and the drop trajectory in the device. However, due to the geometric complexity of the rotor-stator device numerical calculations are necessary in order to obtain the generated flow field. The obtained experimental data is analysed by two different methods. In the first method the recorded drop deformations are time averaged and compared to a defined apparent shear rate which does not rely on numerical flow field calculations. The results from this analysis indicate that there is a relationship between the average drop deformation and the apparent shear rate.

In the second method the experimentally obtained particle track is used together with numerical calculations in order to obtain the local flow experienced by the drop along its track. The data from these calculations lead to time-dependent shear and elongation rates which are used for generating time dependent boundary conditions for the FE-VOF simulations. By using this procedure the flow field experienced by the drop in the rotor-stator device is emulated in the computational box used for carrying out drop shape simulations. Comparison of simulated and experimentally obtained deformations show that in general the agreement is acceptable on a qualitative level. However, the simulations predict deformations which are up to 100% larger than experimentally observed. We have also compared our FE-VOF simulations with results from Boundary Integral (BI) simulations and find good agreement between the two numerical methods.

A number of the conducted experiments resulted in drop break-up. The break-up behavior in the rotor-stator device is analyzed qualitatively by relating the configuration of the cylinders with the initiation of the break-up sequence. Here we observe that drop break-up is initiated when a drop travels from a region of minimum gap width into a region with maximum gap width where there is a relaxation in the flow field. Furthermore we observe that for small viscosity ratios ($\lambda \lesssim 0.1$) tip streaming is predominant while for larger viscosity ratios either binary or capillary break-up is predominant.

Dansk resumé

Denne afhandling omhandler numeriske og eksperimentelle studier af dråbedeformationer i dispergerede systemer, hvor der er påtrykt et flydefelt.

Den implementerede model er baseret på en Finite Element (FE) Stokes løser, som er koblet til en Volume of Fluid (VOF) sporingsprocedure. Til diskretisering af beregningsdomænet anvendes Q_2P_0 elementer, mens VOF algoritmerne er baseret på PLIC (Piecewise Linear Interface Calculation) grænsefladerekonstruktion og en såkaldt "split operator" Lagrange transportprocedure. Den implementerede transportalgoritme er i stand til at bevare volumenet af den dispergerede væskefase indenfor 0.01% i.f.t. det oprindelige volumen. Modellen er implementeret for 3D problemer og kan anvendes til at simulere den tidsafhængige opførsel af to-fase problemer med bevægelige grænseflader. Grænsefladespændingen beregnes enten v.h.a. Continuous Surface Stress (CSS) modellen (Lafaurie et al. 1994) eller Continuous Surface Force (CSF) modellen (Brackbill et al. 1992). Da der opstår regioner med stor krumning i.f.m. stærkt deformerede dråber, er det nødvendigt at anvende et meget fininddelt net i beregningerne. Dette medfører imidlertid lange beregningstider, især pga. det store antal ligninger, der er forbundet med løsningen af flydefeltet. For at reducere den nøvendige beregningstid anvendes en 2-niveaus (2-level) metode. I denne metode anvendes et fint net (VOF-nettet) til de algoritmer, som anvendes til at spore den dispergerede væskefase, mens et grovere net (FE-nettet) anvendes at bestemme flydefeltet. Når 2-niveaus metoden anvendes, beregnes grænsefladespændinger som en sum af bidrag fra VOF-nettet, hvilket i princippet betyder, at krumningen af grænsefladen får lov at variere i et givent element i FE-nettet.

Den implementerede model er blevet undersøgt mht. forfinelse af netindelingen samt forfinelse af tidsskridtet. Ved disse simuleringer studeres deformationen af en enkelt dråde i et simpelt forskydningsfelt. Herudover er vægeffeketer blevet undersøgt ved at variere størrelsen af beregningsdomænet, som består af en kasse indelt i et net med variable elementstørrelser. I midten af denne kasse, hvor dråber befinder sig, anvendes et fint net med konstante elementsidelængder. Tættere på væggene stiger elementstørrelsen gradvist. Simuleringerne viser, at vægeffekterne kan negligeres, når afstanden fra dråben til beregningsdomænets vægge er $24r_0$ (r_0 er dråbens begyndelsesradius). Simuleringerne med forfinelse af netindelingen viser, at et VOF-net, hvor sidelængden af VOF elementerne i den fine del af nettet er $h_{vof} = r_0/18$, er tilstrækkeligt når viskositetsforholdet mellem dråben og den kontinuerte fase er 1. Mere indgående undersøgelser af modellen er udført i simple forskydningsfelter og i forlængelsesfelter. Disse simuleringer er blevet udført ved forskellige

kapillartal og viskositetsforhold og inkluderer også dråbeopsplitning og sammensmeltning af to dråber. Resultaterne fra simuleringerne stemmer generelt godt overens med resultater beskrevet i litteraturen. Men simuleringerne viser også, at når viskositetsforholdet er forskelligt fra 1, er det nødvendigt at anvende et finere FE-net sammenlignet med resultater opnået med ens viskositet af begge faser. En af grundene til dette beror sandsynligvis på den metode, som anvendes til at beregne viskositeten i elementer, der indeholder begge væskefaser.

Den eksperimentelle del af afhandlingen omhandler deformationen af en enkelt dråbe, som påvirkes af et kompliceret flydefelt. Den eksperimentelle opstilling er baseret på et såkaldt rotor-stator apparat som består af to concentriske cylindre med tænder på væggene. Dråbens deformation og position i apparatet overvåges vha. et system bestående af to digitale kameraer. De indspillede videosekvenser behandles efterfølgende vha. automatiseret billedanalyse for at måle den tidsafhængige deformationsgrad samt dråbens partikelbane i apparatet. Pga. apparatets komplekse geometri er det imidlertid nødvendigt at anvende numeriske beregninger til at bestemme flydefeltet. To metoder anvendes til at analysere de målte data. Den første metode er baseret på gennemsnitlige værdier af deformationsgraden, som sammenholdes med en forskydningshastighed defineret ud fra geometrien af apparatet (apparent shear rate). Denne forskydningshastighed beror ikke på numeriske beregninger af flydefeltet. De opnåede resultater fra denne analyse indikerer, at der er en sammenhæng mellem den definerede apparent shear rate og gennemsnitsdeformationen.

I den anden metode anvendes den eksperimentelt opnåede partikelbane sammen med numeriske beregninger til at bestemme dråbens lokale flydefelt i rotor-stator apparatet. Fra disse beregninger fås tidsafhængige forskydnings- og forlængelseshastigheder, som efterfølgende anvendes til at påtrykke tidsafhængige grænsebetingelser til simuleringer af dråbedeformationen. Ideen med denne procedure er at efterligne dråbens lokale flydefelt i vores kasseformede beregningsdomæne, der anvendes til at simulere dråbens form. Den kvalitative overensstemmelse mellem de simulerede og eksperimentelt opnåede dråbedeformationer er acceptabel, selvom simuleringerne generelt resulterer i større deformationer sammenlignet med eksperimenterne. Resultater fra den implementerede FE-VOF model er også blevet sammenlignet med resultater fra en Boundary Integral (BI) metode. Resultaterne fra de to numeriske metoder stemmer godt overens.

En række af de udførte eksperimenter resulterede i dråbeopslitning. Selve opsplitningsforløbet er analyseret kvalitativt ved at sammenholde konfigurationen af rotor-stator apparatet med udviklingen af dråbens form. Her ses typisk, at opslitningsforløbet finder sted, når dråben passerer fra et område med en lille vægafstand til et område med en stor vægafstand. Herudover viser forsøgene, at når viskositetsforholdet er lavt ($\lambda \lesssim 0.1$) fås typisk et opsplitningsforløb, hvor små dråder knibes af enderne på hoveddråben (tip streaming). For større viskositetsforhold er forløbet karakteriseret enten ved binær- eller kapillæropsplitning.

Contents

List of Figures

List of Tables

List of symbols

Roman symbols

CFL	The Courant number	

Boldface roman symbols

$\mathbf{J}$	Jacobian matrix	
$\det \mathbf{J}$	Determinant of the Jacobian matrix	

Italic symbols

$A_{j,i}$	Area of face i on VOF cell j	[m^2]
B	Drop breadth	[m]
(c_1, c_2, c_3)	Side lengths of the standard element used for determining PLIC planes	[m]
Ca	Capillary number	
Ca_{app}	Capillary number based on the apparent shear rate	
Ca_{crit}	Critical capillary number	
D	Taylor deformation parameter	
d	Distance from inner cylinder wall to drop center	[m]
$\langle D \rangle$	Time averaged Taylor deformation parameter	
D_s	Steady state Taylor deformation parameter	
F	VOF function (F-field)	
F_i	VOF function value in VOF cell i	
$f_{j,i}$	Fluid flux across face i on VOF cell j	[m^3/s]
G	Straining rate (e.g. shear rate in simple shear)	[s^{-1}]

h_{fine}	Finite element side lengths in the fine mesh region	[m]
h_{vof}	VOF cell side lengths in the fine mesh region	[m]
I	Global node	
L	Drop length	[m]
L_1, L_2, L_3	Domain half lengths in direction $(1, 2, 3)$	[m]
L_i	Tri-linear interpolation function associated with linear node i	
$L_{i,k}$	Width of inner cylinder wall at positions $i \in \{0, 1, 2\}$	[m]
$L_{o,0}$	Width of outer cylinder	[m]
m_k	Plane segment normal component k in the local coordinate system	
N_e	Total number of finite elements	
N_n	Total number of velocity nodes	
N_1, N_2, N_3	Number of VOF cells per finite element in direction $(1, 2, 3)$ in the parent element	
N_v	Number of VOF cells per finite element when $N_1 = N_2 = N_3 = N_v$	
N_n^p	Local pressure shape function associated with element n	
N_i^v	Local velocity shape function associated with local node i	
N_{vof}	Total number of VOF cells	
p	Pressure	[Pa]
$\hat{p}$	Finite element approximation to the pressure	[Pa]
r_0	Initial drop radius	[m]
$\bar{R}$	Average radius of cylinder walls	[m]
Re_d	Drop Reynolds number	
Δt	Time step parameter	[s]
t	Time	[s]
Δt^*	Modified time step parameter	[s]

t^+	Dimensionless time based on the applied strain rate	
t^*	Dimensionless time based on characteristic drop time	
v_k	Velocity component k	$[\mathrm{m/s}]$
V	Volume of disperse phase in a VOF cell	$[\mathrm{m}^3]$
V^*	Volume bounded by the interception between a plane and a VOF cell	$[\mathrm{m}^3]$
V_{tot}	Total volume bounded by the plane and the coordinate axes	$[\mathrm{m}^3]$
$\hat{v}_k$	Finite element approximation to velocity component k	$[\mathrm{m/s}]$
$VF_{j,i}^{rel}$	Fluxed volume across face i relative to total VOF cell volume j	
$\hat{v}_{k,i}$	Velocity component k at local node i	$[\mathrm{m/s}]$
x_j	j'th component of the Cartesian coordinate	$[\mathrm{m}]$
$\hat{x}_{j,i}$	j'th component of the Cartesian coordinate for local velocity node i	$[\mathrm{m}]$

Boldface italic symbols

$\boldsymbol{e}_1, \boldsymbol{e}_2$	Unit vectors parallel ($\boldsymbol{e}_1$) and perpendicular ($\boldsymbol{e}_2$) to the flow	
$\boldsymbol{f}$	External body force per volume	$[\mathrm{N/m^3}]$
$\boldsymbol{F}_s$	Interfacial tension force per volume	$[\mathrm{N/m^3}]$
$\boldsymbol{I}$	Identity tensor	
$\boldsymbol{T}$	Capillary pressure tensor	$[\mathrm{N/m^2}]$
$\boldsymbol{v}$	Velocity vector	$[\mathrm{m/s}]$
$\boldsymbol{x}$	Coordinates in the Cartesian system	$[\mathrm{m}]$

Greek symbols

α	Interface plane position or relative magnitude of elongation in the flow	
α_{max}	Maximum value of α for which a plane cuts the sides of a VOF cell.	

$\beta_{j,i}$	Geometric factor used for transforming fluid fluxes from the global to local coordinate system	
δ_s	Delta function associated with the interface region	$[\mathrm{m}^{-1}]$
$\dot{\epsilon}$	Elongation rate	$[\mathrm{s}^{-1}]$
$\dot{\gamma}$	Shear rate	$[\mathrm{s}^{-1}]$
$\dot{\gamma}_{app}$	Apparent shear rate	$[\mathrm{s}^{-1}]$
$\langle\dot{\gamma}_{app}\rangle$	Time averaged apparent shear rate	$[\mathrm{s}^{-1}]$
κ	Curvature	$[\mathrm{m}^{-1}]$
λ	Viscosity ratio between drop phase and contiuous phase	
μ	Fluid viscosity	[Pas]
μ_c	Viscosity of continuous phase	[Pas]
μ_d	Viscosity of drop phase	[Pas]
μ_n	Average viscosity in finite element n	[Pas]
Ω	Volumetric integration domain	
Ω_i	Volume of VOF cell i	$[\mathrm{m}^3]$
Ω_n	Volumetric integration domain of element n	
$\dot{\omega}$	Angular frequency	$[\mathrm{s}^{-1}]$
Φ	Arbitrary scalar function	
ϕ^p	Global Pressure trial function	
ϕ^v	Global Velocity trial function	
$\pi_{k,j}$	Components of the total stress tensor, $j \in \{1,2,3\}$ and $k \in \{1,2,3\}$	$[\mathrm{Kg}/\mathrm{m}^3]$
ρ	Fluid density	$[\mathrm{Kg}/\mathrm{m}^3]$
ρ_c	Density of continuous phase	$[\mathrm{Kg}/\mathrm{m}^3]$
ρ_d	Density of drop phase	$[\mathrm{Kg}/\mathrm{m}^3]$
ρ_n	Average density in finite element n	$[\mathrm{Kg}/\mathrm{m}^3]$

σ	Interfacial tension parameter	[N/m]
θ	Angle in degrees or radians	
(ξ, η, ζ)	Local coordinates in the parent element	[m]
(ξ_i, η_i, ζ_i)	Nodal coordinates in the parent element	[m]

Boldface greek symbols

$\boldsymbol{\delta}_k$	Unit vector in coordinate direction k	
$\dot{\boldsymbol{\gamma}}$	Rate of strain tensor	
$\boldsymbol{\pi}$	Total molecular stress tensor	[Pa]
$\hat{\boldsymbol{\pi}}$	Finite element approximation to the total molecular stress tensor	[Pa]
$\boldsymbol{\tau}$	Viscous stress tensor	[Pa]

Chapter 1

Introduction

The term emulsion is the general designation for liquid-liquid two-phase systems wherein one phase is dispersed as drops in the second continuous phase. Emulsions appear in a number of industries as process intermediates and/or final products, e.g. in connection with the production of foodstuffs, pharmaceuticals, cosmetics and in polymer processing. The micro-structure of an emulsion, particularly the drop size distribution and drop phase concentration, is of major significance to the properties of the emulsion based system, e.g. rheology and texture. In order to control (or tailor make) the micro structure one or more dispersion processes are usually applied. Such processes include static mixing, extrusion and pumping. The purpose of these processes is, through the applied flow field, to alter the drop size distribution through coalescence and break-up of the dispersed droplets. The stresses associated with the applied flow will tend to deform the dispersed drops and depending on flow type and strength the stresses can also induce drop break-up. The opposite process can also occur, i.e. the merging of two drops as a result of a flow induced collision. Hence there is a close interaction between the process conditions and the micro structure (morphology) during a dispersion process. In order to optimize a dispersion process with respect to the final micro structure it is of major importance to understand the relationship between flow and drop deformation behavior. This problem has been subject to numerous investigations since the seminal work of G. I. Taylor in the 1930's where he carried out both theoretical and experimental work on drop deformation behavior in simple model flow fields (Taylor 1932, Taylor 1934). In his work Taylor studied the deformation of a single drop suspended in a second imiscible liquid undergoing either simple shear or planar elongation. The experimental setups were based on the shear band apparatus and on the 4-roll mill, cf. Figure 1.1 (a) and (b).

The research on dispersion phenomena following Taylor's work, has mainly concentrated on single drop studies in well defined and homogeneous flows. In real dispersing devices the flow is, however, generally highly complex and inhomogeneous. The theoretical and empirical principles obtained from the model flow fields can therefore in general not be extended to real process applications. The subject of this thesis is to setup and implement a full 3D numerical method for the direct simulation of drop deformation behavior in general laminar flows. The purpose of the model is to use it as a tool for analyzing flow induced

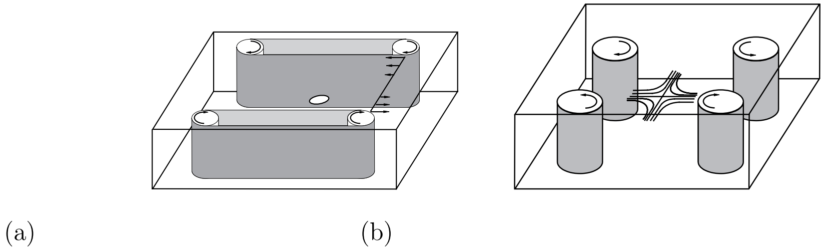


Figure 1.1: The shear band apparatus (a) and four roll mill (b)

drop deformations in real dispersing flows (except for turbulent flows). One important requirement to the model is that it has to be able to handle large deformations and also drop break-up and drop merging.

In addition to the modeling work, experimental investigations of drop behavior in complex flow fields have been carried out. The experimental results are also compared to numerical simulations.

In the following subsections results reported in the literature concerning the understanding of flow induced drop deformations are described. In section 1.1 focus is on experimental and theoretical work while in section 1.2 the focus is on direct numerical methods used for simulating the behavior of dispersed systems.

1.1 Drop deformation: Experiments and theories

Due to the simplicity of the simple shear and planar elongation flow fields they have been used extensively as model flows for studying droplet deformations. In order to characterize the strength of the flow working to deform the drop relative to the strength of the interfacial tension forces working to resist the deformation, the capillary number is defined as:

$$Ca = \frac{Gr_0\mu_c}{\sigma} \qquad (1.1\text{-}1)$$

In (1.1-1) G is the strength of the flow field, e.g. shear rate $\dot{\gamma}$ for simple shear, r_0 is the initial drop radius, μ_c is the viscosity of the continuous phase and σ is the interfacial tension coefficient. In addition to the capillary number the viscosity ratio λ between the drop phase and the continuous phase has a large impact on the drop deformation behavior. Taylor's theoretical treatment which is based on the small deformation assumption relates the degree of deformation to the capillary number and viscosity ratio (Taylor 1932, Taylor 1934):

$$\frac{L-B}{L+B} = D = Ca\frac{19\lambda+16}{16\lambda+16} \qquad (1.1\text{-}2)$$

In (1.1-2) L is the length of the drop along its major axis and B is the breadth of the drop, cf. Figure 1.2.

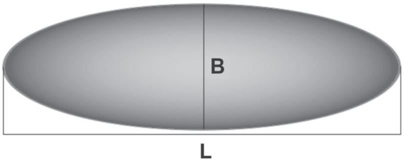


Figure 1.2: Definition of length L and breadth B of a deformed drop.

If the strength of the flow field exceeds a certain critical value which leads to a critical capillary number Ca_{crit}, the drop will no longer be able to withstand the deformation and eventually breaks up into a number of smaller drops. Very coarsely the break-up process can be classified as either binary or capillary. In binary break-up a drop breaks up into two new drops of equal size with a number of smaller drops in between. In capillary break-up the drop is first deformed into a very long and thin liquid thread before it breaks up into a large number of smaller drops. The two break-up types are sketched in Figure 1.3.

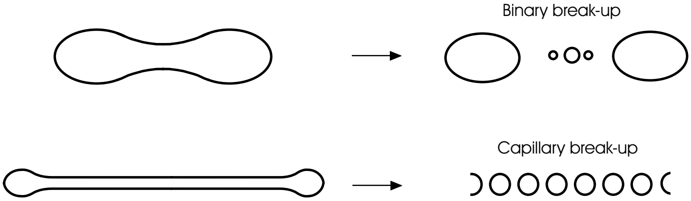


Figure 1.3: Sketch illustrating binary and capillary break-up.

In addition to equation 1.1-2 Taylor also gave an expression for the critical capillary number which for shear flow is:

$$Ca_{crit} = \frac{8\lambda + 8}{19\lambda + 16} \tag{1.1-3}$$

The critical capillary number for planar elongation is half of that in shear flow. The type of break-up, i.e. binary or capillary, depends on the type of flow field, the capillary number and the viscosity ratio. For example if the capillary number is much larger than the critical value the drop is typically elongated into a thin thread (threading) before it breaks up. The threading phenomena is also seen for low viscosity ratios in simple shear. For $\lambda = \mathcal{O}(1)$ and Ca above but close to the critical value binary break-up is observed in simple shear. In elongational flows capillary break-up is observed if the viscosity of the drop phase is comparable to or smaller than the continuous phase viscosity.

The theoretical work of Taylor, which is valid for $\lambda = \mathcal{O}(1)$, has been verified experimentally by e.g. Taylor (1934) and Rumscheidt & Mason (1961). An improvement to Taylor's theoretical description was made by Chaffey & Brenner (1967) by including second order terms of the deformation in their calculations. This theory is frequently termed second order theory while Taylor's (1934) is termed first order theory. A direct extension of Taylor's first order theory to include arbitrary values of the viscosity ratio λ and capillary number was made by Cox (1969). In his analysis the only requirement is that λ and Ca have such values that the resulting drop deformation is low. Torza, Cox & Mason (1972) compare experimental results with the theory by Cox (1969) and find reasonable qualitative agreement. However, Rallison (1980) clarifies and makes some corrections to the theory which leads to predictions which are in better quantitative agreement with the experimental results of Torza et al. (1972).

At the 3rd Engineering Conference on Mixing in 1971 H. P. Grace presented experimental results on critical capillary numbers for a very large range of λ, both in simple shear and planar elongation. The results were later published in (Grace 1982). His results are shown in Figure 1.4 and amongst other things it can be seen that the critical capillary number is always lower for elongation than for shear. Furthermore, the data also show that, in simple shear break does not occur for $\lambda \gtrsim 4$. This behavior was also seen by Taylor (1934). As mentioned above a drop will deform into a long thin thread before break-up when λ is small. Thus the small deformation theory is no longer valid.

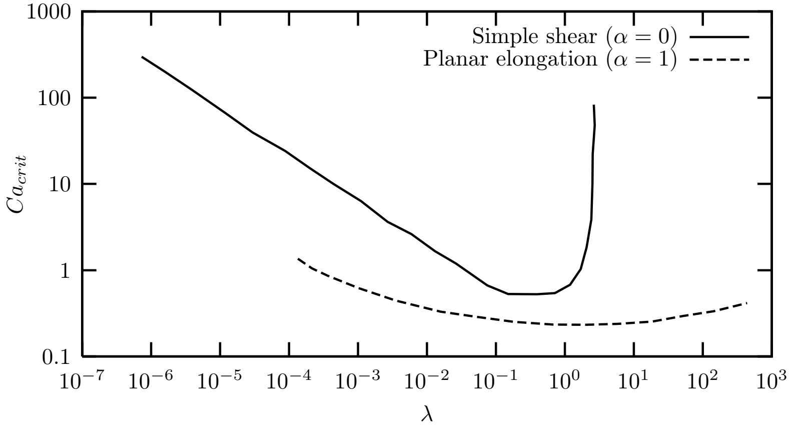


Figure 1.4: Experimental data from (Grace 1982). The curves show the critical capillary number as a function of the viscosity ratio for experiments carried out in simple shear and planar elengation.

In order to handle the deformation behavior at low viscosity ratios ($\lambda < 0.1$) slender body theory was developed by Taylor (1964) with later work by Buckmaster (1972, 1973), Acrivos & Lo (1978), Rallison & Acrivos (1978) and Hinch & Acrivos (1979, 1980). The slender body theory allows for prediction of the drop shape at large deformations as well as the following criteria for the critical capillary number

Simple shear (Hinch & Acrivos 1980):

$$Ca_{crit} = 0.0541\lambda^{-2/3} \tag{1.1-4}$$

Planar elongation (Hinch & Acrivos 1979):

$$Ca_{crit} = 0.15\lambda^{-1/6} \tag{1.1-5}$$

In 1986 Bentley & Leal (1986*a*) presented a computer controlled version of the 4-roll mill device which utilized camera technology and computerized image analysis. One of the great advantages of this device is that as well as pure elongation and pure shear flows mixtures of both can be generated. The amount of elongation relative to shear is determined through a parameter α. A series of experiments have been carried out in this apparatus for $\alpha \in [0; 1]$, e.g. (Bentley & Leal 1986*b*, Stone, Bentley & Leal 1986). These give detailed information about the effect of the flow field on the deformation behavior and the relationship between the critical capillary number and α.

Computer controlled versions of the shear band apparatus have also been developed, e.g. (Guido & Villone 1998, Birkhofer, Eischen, Megias-Alguacil, Fischer & Windhab 2005) which allow for extended experimental times and experiments with time dependent shear rates. In the shear flow cell of Guido & Villone (1998) the full 3D drop shape can be analyzed, i.e. both in the vorticity direction and in the flow direction. Some recent experimental studies on drop deformation are reported in (Guido & Villone 1998, Guido, Greco & Villone 1999, Guido, Minale & Maffetone 2000, Guido & Greco 2001, Megias-Alguacil, Feigl, Dressler, Fischer & Windhab 2005).

Both the small deformation theory and the slender body theory give good predictions of the critical capillary number in homogeneous and steady flow fields. However, as discussed earlier, the flow in industrial dispersion devices is usually complex and unsteady. Thus the theoretical and experimental results based on the model flows cannot be used for the design of emulsion processes except maybe for 'rule of thumb' parameters. Furthermore no theory is able to predict the drop shape accurately during the break-up process and thus detailed information on final drop size distributions in general flow situations cannot be extracted from theory. In order to overcome this problem the development of direct numerical methods for the simulation of drop deformations has been pursued during the last 25 years - and with success.

1.2 Drop deformation: Direct numerical methods

In order to simulate the shape of a dispersed drop deformed by a flow field the applied numerical method has to be able to handle two-phase systems with large distortions of the drop interface. Furthermore in order to simulate drop break-up and drop merging, the method also needs to be able to handle separation and coalescence of interfaces. The most widely used methods in connection with simulation of dispersed systems is the Boundary Integral (BI) method and the Volume of Fluid (VOF) method. In the BI method a calculational mesh is applied onto the boundary of the drop. This mesh is then deformed with the flow field. The BI method has been used for studying the deformation and break-up of single drops in shear flows (Rallison 1981, Kennedy, Pozrikidis & Skalak 1994, Kwak & Pozrikidis 1998, Cristini, Blawzdziewicz & Loewenberg 2001, Cristini, Guido, Alfani, Blawzdziewicz & Loewenberg 2003) and for studying systems consisting of multiple drops (Loewenberg & Hinch 1996, Loewenberg & Hinch 1997, Cristini et al. 2001, Cunha & Loewenberg 2003). Furthermore a numerical method is described in (Feigl, Kaufmann, Fischer & Windhab 2003) where the drop deformation behavior in a time dependent dispersing flow is analyzed using a combination of the BI method and flow field calculations along streamlines. We will return to this method in chapter 5 where more details are given. The BI method has also been used for investigating the effects of surfactants on drop deformation behavior, e.g. (Yon & Pozrikidis 1998, Pozrikidis 2001). One drawback of the BI method is that break-up and merging of drops cannot be handled in a 'natural' way since the onset of break-up or coalescence needs to be defined by the user.

The VOF method is a volume tracking method where, for two phase systems, the position of one of the phases is defined through a VOF function (or color function). The method requires a flow solving routine and a volume tracking routine. A number of studies on drop deformation and particularly drop break-up have been carried out using the VOF method, e.g. (Mashayek & Ashgriz 1995, Li, Renardy & Renardy 2000, Renardy, Cristini & Li 2002, Renardy & Renardy 2002, Renardy, Renardy & Cristini 2002, Khismatullin, Renardy & Renardy 2006, Renardy 2007). These studies include investigations of the effect of inertia on drop shape (Li et al. 2000), drop size distributions after drop break-up (Renardy, Cristini & Li 2002), effects of domain confinement on the critical capillary number (Renardy 2007) and effects of non-Newtonian continuous and drop phase liquids (Khismatullin et al. 2006) (based on the Giesekus constitutive equation). Furthermore Renardy, Renardy & Cristini (2002) present a method for including surfactants in VOF interfacial tension calculations. One main advantage of the VOF method compared to the BI method is that drop break-up and coalescence does not need any special consideration from the user which makes it very well suited for studying problems involving break-up and merging. On the other hand it requires a full 3D time dependent flow solver and thus can become very computationally expensive. In our work we have chosen to base our numerical model on the VOF method due to the ease with which drop break-up can be handled. In our implementation we have devoted a great deal of attention on minimization of computational cost without loosing (too much) accuracy.

It is noted that other numerical methods exist for handling two-phase flows, these include particle marker methods, level-set methods and Arbitrary Eulerian-Lagrangian (ALE) finite element methods. However, each of these methods present some severe limitations in their use for the simulation of droplet deformations. For example the particle marker methods are too time consuming in 3D problems, the level-set method gives rise to difficulties when drop break-up occurs and does not conserve volume rigorously and the ALE methods requires extensive remeshing in order to handle the large interface distortions associated with large drop deformations.

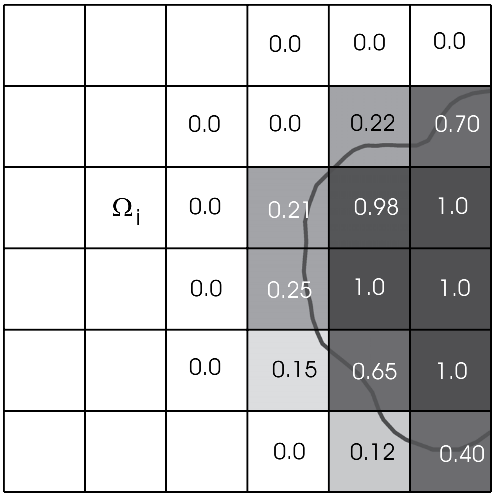
In order to facilitate the presentation of the numerical model in the forthcoming chapters the next section will be devoted to a brief introduction to the VOF method.

1.3 Introduction to the VOF method

For a two-phase system consisting of disperse droplets we define the VOF function $F(\boldsymbol{x})$ to be the volume fraction of the disperse phase at position $\boldsymbol{x}$ in the domain. In practice F is a discrete function which is associated with a mesh consisting of a finite number (N_{vof}) of VOF cells with volumes $\Omega_i,\ i = 1, \dots N_{vof}$. The value of F in a given cell then corresponds to the volume fraction of the disperse phase in the cell. In summary we have:

$$\begin{aligned} F_i &= \begin{cases} 0 & \text{if } \Omega_i \text{ is outside the disperse phase} \\ 1 & \text{if } \Omega_i \text{ is inside the disperse phase} \end{cases} \\ 0 &< F_i < 1 \text{ if } \Omega_i \text{ holds part of the interface} \end{aligned} \tag{1.3-6}$$

Figure 1.5 shows a sketch of part of an interface and the corresponding F-field (not exact). Here it is seen that the interface between the disperse phase and the exterior phase is not sharply defined through the F-function and that F is discontinuous across the interface.



			0.0	0.0	0.0
		0.0	0.0	0.22	0.70
	Ω_i	0.0	0.21	0.98	1.0
		0.0	0.25	1.0	1.0
		0.0	0.15	0.65	1.0
			0.0	0.12	0.40

Figure 1.5: 2D sketch of an arbitrary interface (solid curve) described by the discrete VOF function F_i on VOF volumes Ω_i.

In order to track the movement of the disperse phase using the VOF method it is necessary to be able to transport the F-function in a given flow field. Details in transport methods will not be given here since this is the subject of sections 2.4.1 and 2.4.2. However, it is noted that all transport algorithms (advection algorithms) rely on some sort of reconstruction of the interface from the F-function. Two such reconstructions are illustrated in Figure 1.6 where the interface is reconstructed with line segments locally in each interface VOF cell. The left figure shows a reconstruction with line segments parallel to the VOF cell sides and right figure shows line segments with arbitrary orientation. The reason why interface reconstruction is needed is to avoid numerical diffusion (smearing) of the disperse phase. More details about reconstruction algorithms are given in section 2.4.1.

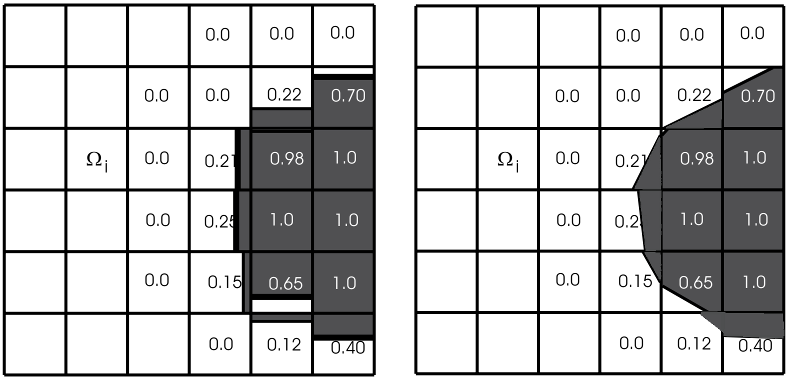


Figure 1.6: 2D sketch indication how the interface in figure 1.5 can be reconstructed using line segments which are either parallel to the VOF cell sides (left) or have arbitrary orientation (right). Sketch is not exact.

The VOF method is usually accredited to Hirt & Nichols (1981). However, DeBar (1974) and Noh & Woodward (1976) describe methods which bear a close resemblance to the method of Hirt & Nichols (1981) the main differences being details in the flux calculations when transporting the VOF function in the flow. In this thesis all volume tracking methods based on the transport of a scalar VOF function will be denoted as VOF methods regardless of the details in the actual transport algorithm.

The flow field in which to advect the VOF function is obtained using a flow solver. The flow solver is coupled to the position of the two phases relative to each other through differences in physical parameters such as viscosity μ and density ρ and also through the presence of interfacial tension. In Figure 1.7 a coarse overview of the main algorithms and information flows associated with the implementation of VOF methods can be seen. Inputs to the model are the viscosities and densities of the continuous phase (subscript c) and drop phase (subscript d), the interfacial tension coefficient σ and the flow field applied onto the domain boundaries. Hereafter the interfacial tension forces and suitable values of the viscosity and density are calculated and used in the flow solver. When the flow field has been solved the interfaces are reconstructed from the F-function where after a new position of the drop phase is obtained when F has been transported in the flow field. The distance the disperse phase is transported depends on the size of a time step parameter

Δt. The procedure is repeated until the final simulation time is reached.

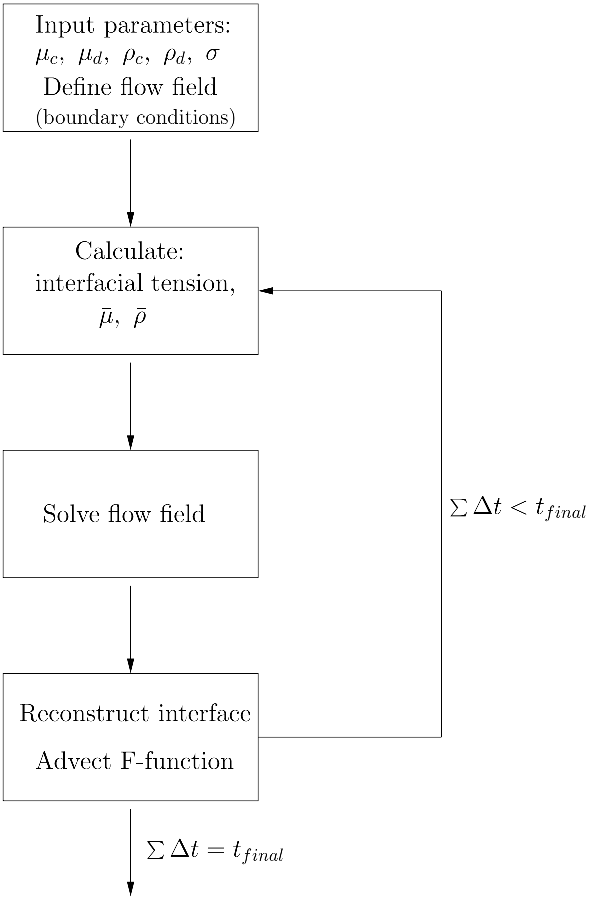


Figure 1.7: Flow chart showing the main algorithms and information flow of the implemented model.

The flow solver can be based on a number of different discretization schemes. In the older literature finite difference schemes were applied (these were the ones available at the time), e.g. (DeBar 1974, Hirt & Nichols 1981). In more recent VOF implementations Finite Volume schemes (Ginzburg & Wittum 2001) and Finite element schemes have become widely used. Especially the implementation of VOF methods with FE solvers has become popular due to the strength of FE methods in connection with complex domain geometries, e.g. (Mashayek & Ashgriz 1995, Wu, Yo & Jiang 1998, Shin & Lee 2000, Wang 2002, Pantuso, Jiang, Shankar & Skokov 2003). For this reason we have also chosen to base our flow solver on the FE method.

1.4 The lay-out of the thesis

The work presented in this thesis consists of a modeling part and an experimental part. In the modeling part we begin by presenting the governing flow equations and their transformation into the finite element formulation (sections 2.1 and 2.2). Then the implemented flow solver is tested and its solution verified in terms of spatial convergence (section 2.3). Section 2.4 is concerned with the VOF algorithms, these include surface reconstruction and fluid advection (section 2.4.2), coupling of the flow field obtained from the flow solver to the VOF mesh (section 2.4.2.4) and tests of the advection algorithm in predefined flow fields (section 2.4.3). Since the exact position of fluid interfaces is not known when using the VOF method a special method needs to be applied in order to include interfacial tension forces in the flow calculations. The inclusion of interfacial tension is the subject of section 2.5 with some simple tests given in section 2.7 and 2.8. Finally the full model, including interfacial tension, is tested in terms of spatial and temporal convergence in section 3.2. Further tests and example simulations on drop deformation behavior in simple flow fields are given in section 3.3 and 3.4. These simulations include drop break-up and drop coalescence.

The experimental work carried out is based on studies of drop deformation behavior in complex flow fields. The experimental setup and procedure are presented in section 4.1 and 4.3. A key part of the data analysis involves an automated image analysis procedure which is explained in section 4.4. Due to the complexity of the flow field generated in the experimental apparatus, numerical calculations are required in order to obtain the flow field. The analysis of the experimental data is split up into two parts. The first part (section 4.5.1) is based on time averaged values of the deformation and a geometry based apparent shear rate. In these data no information regarding the real flow field is used. In the second part (chapter 5) the flow field is calculated numerically for a number of experiments. Following the procedure described in (Feigl et al. 2003) the flow field is then used for carrying out simulations of the time dependent drop shape using the implemented VOF model. The results from the VOF model are compared to the experimentally obtained drop deformations (section 5.2) and to results obtained using a well established BI code[1] (section 5.3).

Finally the thesis is concluded with an overall summary of the work carried out and possibilities for future work.

[1] The original code is developed by (Loewenberg & Hinch 1996) and has later been adapted by Feigl et al. (2003) to handle time dependent flow conditions.

Chapter 2

The Finite Element-Volume of Fluid model

In this chapter the details of the Finite Element-Volume of Fluid (FE-VOF) method implemented in this project are presented. The model derivation and implementation can be separated into two distinctive parts. The first part is to solve the flow field given some domain boundary conditions and fluid properties. The second problem is to track the two fluids relative to each other in the flow. The flow field solution is linked to the configuration of the two fluid phases through differences in physical properties and the existence of interfacial tension associated with the interface separating the two fluids.

We will begin with the description of the equations governing the flow field with interfacial tension absent. Then the finite element discretization of these equations is explained and the resulting equation system is outlined. However, when solving fluid dynamics problems in 3D domains it is crucial that memory efficient storage schemes are used and that the solution procedure is highly optimized because the number of independent variables usually is very large in realistic problems. The solver and storage scheme used are explained after the discretization section and finally the solver is tested in terms of convergence.

In the second part details of the implemented VOF algorithms are given. Here emphasis is on the advection procedure and calculation of interfacial tension. In our implementation we have added the choice of using a 2-level algorithm in which different mesh element concentrations are used for the calculational mesh and the VOF mesh. An explanation of our 2-level algorithm is given in subsections 2.4.2.4 and 2.5.2.2. A discussion of divergence free interpolation used for interpolating the velocities from the calculational mesh onto the VOF mesh is also given. In section 2.4.3 the advection algorithm is tested in predefined flow fields (no interfacial tension) while in sections 2.7 and 2.8 some simple tests of the implemented interfacial tension models are carried out. More comprehensive tests of the full FE-VOF code is the subject of chapter 3.

2.1 Governing equations

The mathematical description of fluid motion is obtained through the conservation of momentum (the equation of motion) and conservation of mass (the continuity equation). The underlying physics of these conservation laws and their transformation into mathematical models can be found in standard text books on transport phenomena and fluid dynamics such as (Bird, Stewart & Lightfoot 2002) and (Bird, Armstrong & Hassager 1987). In the following these models, which form the basis for the flow solver, are presented.

2.1.1 The equation of motion

In tensor notation the equation of motion is given by (Bird et al. 2002):

$$\frac{\partial}{\partial t}\rho\boldsymbol{v} = -[\nabla \cdot \rho\boldsymbol{v}\boldsymbol{v}] - \nabla p - [\nabla \cdot \boldsymbol{\tau}] + \boldsymbol{f} \tag{2.1-1}$$

In (2.1-1) ρ is density, p is pressure, $\boldsymbol{v}$ is the velocity vector, $\boldsymbol{\tau}$ is the molecular (or viscous) stress tensor and $\boldsymbol{f}$ represents external body forces, e.g. gravitational forces. For an incompressible Newtonian fluid with constant density the viscous stress tensor $\boldsymbol{\tau}$ is given by:

$$\boldsymbol{\tau} = -\mu(\nabla\boldsymbol{v} + (\nabla\boldsymbol{v})^{\mathrm{T}}) \tag{2.1-2}$$

where μ is the viscosity and $(\nabla\boldsymbol{v})^{\mathrm{T}}$ is the transpose of the velocity gradient tensor. It is noted that in the above $\boldsymbol{v}$ and p are functions of time t and position $\boldsymbol{x} = (x_1, x_2, x_3)$. In creeping flows, i.e. for small Reynolds numbers ($Re << 1$), the first term on the RHS of equation 2.1-1 which is quadratic in the velocity and represents inertial effects can be neglected. The flow situations which we will focus on are not necessarily in the creeping flow limit. However, in many cases the local flow surrounding a dispersed drop is characterized by very small drop Reynolds numbers (Re_d) where Re_d is given by:

$$Re_d = \frac{\rho_c G r_0^2}{\mu_c} \tag{2.1-3}$$

In equation 2.1-3 ρ_c is the density of the continuous phase, G is the rate of strain, r_0 is the initial drop radius and μ_c is the viscosity of the continuous phase. We therefore neglect the non-linear term in our calculations as it is the local flow in the proximity of the drop which is of main interest. Furthermore we solve the fluid advection as a quasi steady state problem which means that we need only to solve the steady state momentum balance in each time step. The flow problem thus consists of solving the Stokes equation:

$$\mathbf{0} = -[\nabla \cdot \boldsymbol{\pi}] + \boldsymbol{f} \tag{2.1-4}$$

where $\boldsymbol{\pi} = p\boldsymbol{I} + \boldsymbol{\tau}$ and $\boldsymbol{I}$ is the identity tensor. Written out into its 3 components using Einstein notation the Stokes equation in Cartesian coordinates is given by:

$$0 = -\frac{\partial \pi_{k,j}}{\partial x_j} + f_k, \quad k = 1, \ldots, 3 \tag{2.1-5}$$

where $\partial/\partial x_j$ is the spatial derivative in the j^{th} direction.

2.1.2 The continuity equation

Conservation of mass is described by the continuity equation

$$\frac{\partial \rho}{\partial t} = -(\nabla \cdot \rho \boldsymbol{v}) \tag{2.1-6}$$

For a constant density fluid equation 2.1-6 reduces to

$$0 = (\nabla \cdot \boldsymbol{v}) \tag{2.1-7}$$

or $\partial v_j/\partial x_j = 0, j = 1, \ldots, 3$ in Einstein notation.

2.2 Finite element discretization

In this section we will derive the finite element formulation of the Stokes equations (momentum balance) and the equation of continuity. In the present implementation we have chosen to use quadratic finite element shape functions for the velocity field and discontinuous constant pressures for the pressure field. This combination of velocity and pressure approximations is known to yield stable equation systems, e.g. (Elman, Silverster & Wathen 2005). We will begin by describing the relevant shape functions and derivatives thereof. Hereafter the momentum balance and equation of continuity are cast into their respective finite element formulations and discretized.

2.2.1 The velocity shape functions

Finite element discretization involves diving the domain of interest into a number of subdomains in which the variables, i.e. velocity and pressure, are approximated locally by some type of interpolation (or shape) function. One great advantage of the finite element method is that one can choose from a class of standard geometrical shapes (elements) as representatives of the sub-volumes. This makes the finite element method ideal for problems involving complex geometries. In our case we use hexahedral elements as these types of elements are the ones that are the most 'compatible' with the VOF method, e.g. compared to tetrahedral elements. The element used in the present implementation is the 20 node brick element, cf. figure 2.1. Here all nodes are positioned on the boundary of the element. This element differs from the standard quadratic Lagrange element in which additional nodes are positioned on the faces of the element and in the element centre (Zienkiewicz & Taylor 2000). However, by using the 20 node element one reduces the number of degrees of freedom (DOF) in the final system of equations compared to the standard Lagrange element while still locally approximating the velocities quadratically.

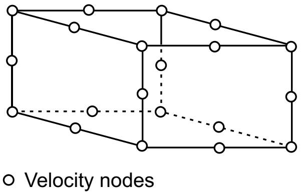


Figure 2.1: 20 node hexahedral element. ∘ symbolizes the position of quadratic nodes.

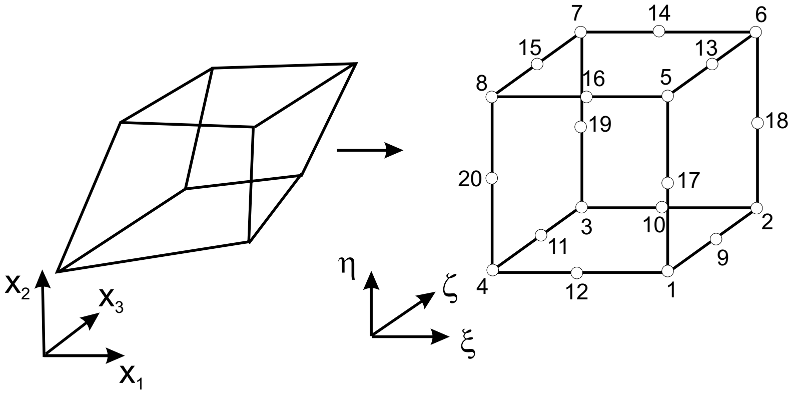


Figure 2.2: Mapping from the calculational domain onto the parent element

Interpolation within a given element is carried out by an iso-parametric mapping from the calculational domain onto a parent element defined in the local coordinate system (ξ, η, ζ), cf. figure 2.2.

The parent element is given by $([-1, 1] \times [-1, 1] \times [-1, 1])$ with the nodes residing at the following positions:

Node	(ξ, η, ζ)	Node	(ξ, η, ζ)
1	$(+1, -1, -1)$	2	$(+1, -1, +1)$
3	$(-1, -1, +1)$	4	$(-1, -1, -1)$
5	$(+1, +1, -1)$	6	$(+1, +1, +1)$
7	$(-1, +1, +1)$	8	$(-1, +1, -1)$
9	$(+1, -1, 0)$	10	$(0, -1, +1)$
11	$(-1, -1, 0)$	12	$(0, -1, -1)$
13	$(+1, +1, 0)$	14	$(0, +1, +1)$
15	$(-1, +1, 0)$	16	$(0, +1, -1)$
17	$(+1, 0, -1)$	18	$(+1, 0, +1)$
19	$(-1, 0, +1)$	20	$(-1, 0, -1)$

Interpolation within an element is then carried out using locally defined shape functions. At a given point (ξ, η, ζ) in the parent element the shape functions are given by (Zienkiewicz & Taylor 2000):

$$\begin{aligned}&\text{Corner nodes,} \quad \xi_i = \pm 1 \quad \eta_i = \pm 1 \quad \zeta_i = \pm 1 \\ &N_i(\xi,\eta,\zeta) = \frac{1}{8}(1+\xi_0)(1+\eta_0)(1+\zeta_0)(\xi_0+\eta_0+\zeta_0-2), \quad i = 1,\dots,8\end{aligned} \tag{2.2-8}$$

$$\begin{aligned}&\text{Mid-side nodes,} \quad \xi_i = 0 \quad \eta_i = \pm 1 \quad \zeta_i = \pm 1 \\ &N_i(\xi,\eta,\zeta) = \frac{1}{4}(1-\xi^2)(1+\eta_0)(1+\zeta_0), \quad i = 10, 12, 14, 16\end{aligned} \tag{2.2-9}$$

$$\begin{aligned}&\text{Mid-side nodes,} \quad \xi_i = \pm 1 \quad \eta_i = 0 \quad \zeta_i = \pm 1 \\ &N_i(\xi,\eta,\zeta) = \frac{1}{4}(1-\eta^2)(1+\xi_0)(1+\zeta_0), \quad i = 17,\dots,20\end{aligned} \tag{2.2-10}$$

$$\begin{aligned}&\text{Mid-side nodes,} \quad \xi_i = \pm 1 \quad \eta_i = \pm 1 \quad \zeta_i = 0 \\ &N_i(\xi,\eta,\zeta) = \frac{1}{4}(1-\zeta^2)(1+\xi_0)(1+\eta_0), \quad i = 9, 11, 13, 15\end{aligned} \tag{2.2-11}$$

where in (2.2-8) through (2.2-11) the variables ξ_0, η_0 and ζ_0 are given by

$$\xi_0 = \xi\xi_i \quad \eta_0 = \eta\eta_i \quad \zeta_0 = \zeta\zeta_i$$

and $i = 1,\dots,20$ are the element nodes. Velocities can then be interpolated everywhere within an element by:

$$\hat{v}_k(\xi,\eta,\zeta) = \sum_{i=1}^{20} N_i(\xi,\eta,\zeta)\hat{v}_{k,i}\ , \quad k = 1,\dots,3 \tag{2.2-12}$$

where $\hat{v}_{k,i}$ is velocity component k at local node i and $\hat{v}_k$ is the interpolated velocity. Globally the velocities can be approximated using global shape functions, i.e.

$$\hat{v}_k(\boldsymbol{x}) = \sum_{I=1}^{N_n} \phi_I^v(\boldsymbol{x})\hat{v}_{k,I}, \quad k = 1,\dots,3 \tag{2.2-13}$$

where $I = 1,\dots,N_n$ corresponds to the global nodes. The global shape function ϕ_I^v is defined in terms of the local shape functions:

$$\phi_I^v(\boldsymbol{x}) = \begin{cases} N_{i(I,n)}(\xi(\boldsymbol{x}),\eta(\boldsymbol{x}),\zeta(\boldsymbol{x})) & ,\boldsymbol{x} \in \bigcup_{n_1=1}^{N_e} \begin{cases}\Omega_{n_1} & \text{if node I} \in \text{element } n_1 \\ \emptyset & \text{otherwise}\end{cases} \\ 0 & ,\text{otherwise}\end{cases} \tag{2.2-14}$$

In (2.2-14) I and n refer to the global node and the relevant element number respectively.

Since we use iso-parametric elements the geometrical mapping between an element in the calculational domain and the local system (ξ, η, ζ) is given by:

$$\hat{x}_k(\xi,\eta,\zeta) = \sum_{i=1}^{20} N_i(\xi,\eta,\zeta)\hat{x}_{k,i}\ , \quad k = 1,\ldots,3 \tag{2.2-15}$$

where $\hat{x}_{k,i}$ are nodal coordinates of the element in question. The derivative of the shape functions with respect to the global coordinates can be obtained through the chain rule:

$$\begin{aligned} \frac{\partial N_i}{\partial \xi} &= \frac{\partial N_i}{\partial x_1}\frac{\partial x_1}{\partial \xi} + \frac{\partial N_i}{\partial x_2}\frac{\partial x_2}{\partial \xi} + \frac{\partial N_i}{\partial x_3}\frac{\partial x_3}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} &= \frac{\partial N_i}{\partial x_1}\frac{\partial x_1}{\partial \eta} + \frac{\partial N_i}{\partial x_2}\frac{\partial x_2}{\partial \eta} + \frac{\partial N_i}{\partial x_3}\frac{\partial x_3}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} &= \frac{\partial N_i}{\partial x_1}\frac{\partial x_1}{\partial \zeta} + \frac{\partial N_i}{\partial x_2}\frac{\partial x_2}{\partial \zeta} + \frac{\partial N_i}{\partial x_3}\frac{\partial x_3}{\partial \zeta} \end{aligned} \tag{2.2-16}$$

Since we can write

$$\frac{\partial x_j}{\partial \xi} = \frac{\partial}{\partial \xi}\left(\sum_{i=1}^{20} N_i \hat{x}_{j,i}\right) = \sum_{i=1}^{20} \frac{\partial N_i}{\partial \xi}\hat{x}_{j,i}, \quad j = 1,\ldots,3 \tag{2.2-17}$$

and similarly for $\partial x_j/\partial\eta$ and $\partial x_j/\partial\zeta$, the local derivatives of the shape functions can be written as

$$\begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{bmatrix} = \mathbf{J} \begin{bmatrix} \frac{\partial N_i}{\partial x_1} \\ \frac{\partial N_i}{\partial x_2} \\ \frac{\partial N_i}{\partial x_3} \end{bmatrix} \tag{2.2-18}$$

where $\mathbf{J}$ is the Jacobian matrix given by:

$$\mathbf{J} = \begin{bmatrix} \sum \frac{\partial N_i}{\partial \xi}\hat{x}_{1,i} & \sum \frac{\partial N_i}{\partial \xi}\hat{x}_{2,i} & \sum \frac{\partial N_i}{\partial \xi}\hat{x}_{3,i} \\ \sum \frac{\partial N_i}{\partial \eta}\hat{x}_{1,i} & \sum \frac{\partial N_i}{\partial \eta}\hat{x}_{2,i} & \sum \frac{\partial N_i}{\partial \eta}\hat{x}_{3,i} \\ \sum \frac{\partial N_i}{\partial \zeta}\hat{x}_{1,i} & \sum \frac{\partial N_i}{\partial \zeta}\hat{x}_{2,i} & \sum \frac{\partial N_i}{\partial \zeta}\hat{x}_{3,i} \end{bmatrix} \tag{2.2-19}$$

The global derivatives are then obtained by inverting equation 2.2-18:

$$\begin{bmatrix} \frac{\partial N_i}{\partial x_1} \\ \frac{\partial N_i}{\partial x_2} \\ \frac{\partial N_i}{\partial x_3} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{bmatrix} \tag{2.2-20}$$

As will be evident later we will need to transform the integrals arising from the finite element formulation of the momentum balance and the continuity equation from the global to the local coordinate system. This requires evaluation of the determinant of the Jacobian matrix which is given by

$$\begin{aligned} \det \mathbf{J} = {} & \frac{\partial x_1}{\partial \xi}\frac{\partial x_2}{\partial \eta}\frac{\partial x_3}{\partial \zeta} + \frac{\partial x_1}{\partial \zeta}\frac{\partial x_2}{\partial \xi}\frac{\partial x_3}{\partial \eta} + \frac{\partial x_1}{\partial \eta}\frac{\partial x_2}{\partial \zeta}\frac{\partial x_3}{\partial \xi} \\ & - \frac{\partial x_1}{\partial \zeta}\frac{\partial x_2}{\partial \eta}\frac{\partial x_3}{\partial \xi} - \frac{\partial x_1}{\partial \xi}\frac{\partial x_2}{\partial \zeta}\frac{\partial x_3}{\partial \eta} - \frac{\partial x_1}{\partial \eta}\frac{\partial x_2}{\partial \xi}\frac{\partial x_3}{\partial \zeta} \end{aligned} \tag{2.2-21}$$

where the derivatives of the global coordinates with respect to the local coordinates are calculated according to equation 2.2-17.

2.2.2 The pressure shape functions

Since the pressures are constant within each element, the local pressure shape function is simply given by:

$$N_n^p(\xi,\eta,\zeta) = 1$$

in element n. The local pressure approximation thus is:

$$\hat{p}(\xi,\eta,\zeta) = N_n^p(\xi,\eta,\zeta)\hat{p} = \hat{p}_n \tag{2.2-22}$$

Globally the pressure approximation can be written as:

$$\hat{p}(\boldsymbol{x}) = \sum_{n=1}^{N_e} \phi_n^p(\boldsymbol{x})\hat{p}_n$$

Here the global pressure interpolation function $\phi^p(\boldsymbol{x})$ is given by:

$$\phi_n^p(\boldsymbol{x}) = \begin{cases} N_n^p(\xi(\boldsymbol{x}),\eta(\boldsymbol{x}),\zeta(\boldsymbol{x})) & ,\boldsymbol{x} \in \Omega_n \backslash \partial\Omega_n \\ \emptyset & ,\boldsymbol{x} \in \partial\Omega_n \\ 0 & ,\text{otherwise} \end{cases} \tag{2.2-23}$$

i.e. the pressure at a given position in the domain is given by the constant and discontinuous pressure in the relevant element n. However, since the pressures are discontinuous they are not defined on element boundaries ($\partial\Omega_n$).

2.2.3 The finite element formulation

Now the momentum balance and the continuity equation are recast into their respective finite element formulations (i.e. weak form). First the momentum balance, equation 2.1-4, is multiplied with an arbitrary velocity shape function ϕ^v and integrated over the global domain Ω:

$$-\int_\Omega \phi^v[\nabla\cdot\boldsymbol{\pi}]dV + \int_\Omega \phi^v\boldsymbol{f}dV = \boldsymbol{0} \tag{2.2-24}$$

Partial integration of the total molecular stress term in (2.2-24) leads to:

$$-\int_\Omega [\nabla\cdot(\phi^v\boldsymbol{\pi})]dV + \int_\Omega [\nabla\phi^v\cdot\boldsymbol{\pi}]dV + \int_\Omega \phi^v\boldsymbol{f}dV = \boldsymbol{0} \tag{2.2-25}$$

By applying the Gauss-Ostrogradskii theorem to the first term on the LHS of (2.2-25) one obtains:

$$-\int_\Gamma \phi^v[\boldsymbol{n}\cdot\boldsymbol{\pi}]dS + \int_\Omega [\nabla\phi^v\cdot\boldsymbol{\pi}]dV + \int_\Omega \phi^v\boldsymbol{f}dV = \boldsymbol{0} \tag{2.2-26}$$

where Γ corresponds to a free surface. The surface integral in (2.2-26) can be used for applying natural boundary conditions on interfaces. This will be used later to incorporate interfacial tension to the interface between the disperse and continuous phase.

Similarly to the momentum balance the continuity equation is multiplied by an arbitrary shape function and integrated over the domain. However, in this case the shape function is associated with the pressure (ϕ^p):

$$\int\limits_{\Omega} \phi^p \nabla \cdot \boldsymbol{v} dV = 0 \tag{2.2-27}$$

2.2.4 The discrete formulation

In order to discretize equation 2.2-26 and 2.2-27 the global integrals are rewritten as sums of integrals over each finite element, $\Omega_n, n = 1, \ldots, N_e$. Furthermore the arbitrary shape functions ϕ^v and ϕ^p are substituted by the local shape functions and velocities and pressures are discretized according to (2.2-12) and (2.2-22) ($\hat{\boldsymbol{\pi}}$ symbolizes the discrete equivalent to $\boldsymbol{\pi}$). For the momentum balance this results in:

$$\sum_{n=1}^{N_e} \int\limits_{\Gamma_n} N^v_{i(I,n)} [\boldsymbol{n} \cdot \hat{\boldsymbol{\pi}}] dS - \sum_{n=1}^{N_e} \int\limits_{\Omega_n} [\nabla N^v_{i(I,n)} \cdot \hat{\boldsymbol{\pi}}] dV + \sum_{n=1}^{N_e} \int\limits_{\Omega_n} N^v_{i(I,n)} \boldsymbol{f} dV = \mathbf{0} \tag{2.2-28}$$

In (2.2-28) $I = 1, \ldots, N_n$ corresponds to the global nodes. It is noted that there is a unique correlation between the local node i on element n and the global node I.

The continuity equation is treated in the same manner which yields

$$\sum_{n=1}^{N_e} \int\limits_{\Omega_n} N^p_n \nabla \cdot \boldsymbol{v} dV = 0 \tag{2.2-29}$$

where in practice $N^p_n = 1$ for all n.

The volume integral over the total molecular stress tensor $\boldsymbol{\pi}$ in the momentum equation can, in its discrete form, be written as:

$$\sum_{n=1}^{N_e} \int\limits_{\Omega_n} \nabla N^v_{i(I,n)} \cdot \hat{\boldsymbol{\pi}} dV = \sum_{n=1}^{N_e} \sum_{k=1}^{3} \left(C_k^{i(I,n)} + \sum_{k_1=1}^{3} A_{kk_1}^{i(I,n)} \right) \boldsymbol{\delta}_k \tag{2.2-30}$$

where e.g. the term $A_{11}^{i(I,n)}$ is given by

$$A_{11}^{i(I,n)} = -\mu \sum_{i_1=1}^{20} \hat{v}_{1,i_1} \int\limits_{\Omega_n} 2 \frac{\partial N_{i(I,n)}}{\partial x_1} \frac{\partial N_{i_1}}{\partial x_1} + \frac{\partial N_{i(I,n)}}{\partial x_2} \frac{\partial N_{i_1}}{\partial x_2} + \frac{\partial N_{i(I,n)}}{\partial x_3} \frac{\partial N_{i_1}}{\partial x_3} dV \tag{2.2-31}$$

and the pressure term $C_k^{i(I,n)}$ is given by

$$C_k^{i(I,n)} = \hat{p}_n \int\limits_{\Omega_n} \frac{\partial N^v_{i(I,n)}}{\partial x_k} N^p_n dV \tag{2.2-32}$$

In (2.2-30) $\boldsymbol{\delta}_k$ is the unit vector in coordinate direction k. In appendix A all the terms are given. Since the local velocity shape functions $N^v_{i(I,n)}$ are only non-zero locally in each element (by definition) the only contributions to $A^{i(I,n)}_{kk_1}$ and $C^{i(I,n)}_k$ arise from elements which share the global node I.

The external force term can be written in a similar manner, i.e.:

$$\sum_{n=1}^{N_e} \int\limits_{\Omega_n} N^v_{i(I,n)} \boldsymbol{f} dV = \sum_{n=1}^{N_e} \sum_{k=1}^{3} f_k^{i(I,n)} \boldsymbol{\delta}_k \tag{2.2-33}$$

where

$$f_k^{i(I,n)} = \int\limits_{\Omega_n} N^v_{i(I,n)} f_k dV$$

Generally the free boundary term is discretized according to the particular Boundary Conditions (BC's) applied to the domain. In the present implementation one can chose between the following type of BC's:

1. Dirichlet BC's - i.e. velocities are specified on boundary nodes
2. Symmetry - velocity component normal to symmetry plane is set to 0
3. Periodic BC's
4. Neutral BC's - corresponds to a free surface without surface tension ($\boldsymbol{n} \cdot \boldsymbol{\pi} = \boldsymbol{0}$)
5. Interfacial tension associated with liquid-liquid interfaces (section 2.5)

In case of Dirichlet, Symmetry and periodic BC's the free boundary term does not need consideration. Furthermore in the case of free domain surfaces we simply apply $\boldsymbol{n} \cdot \boldsymbol{\pi} = \boldsymbol{0}$ and the surfaces integral becomes 0 in this case. However, the presence of interfacial tension associated with liquid-liquid interfaces also needs to be considered. In this case the surface integral in (2.2-28) is used to represent the stress difference across the interface. This problem is addressed in section 2.5. If we for now consider only a system containing a single fluid bounded in a domain with BC's 1-4 stated above the problem consists of calculating the coefficient contained in (2.2-31), (2.2-32) and (2.2-33) and also to include the equation of continuity which in its discrete form can be written as

$$\sum_{n=1}^{N_e} \int\limits_{\Omega_n} N^p_{n_1} (\nabla \cdot \boldsymbol{v}) dV = \sum_{n=1}^{N_e} \sum_{k=1}^{3} D_k^{n_1} = 0, \quad n_1 = 1, \dots N_e \tag{2.2-34}$$

where

$$D_k^{n_1} = \sum_{i=1}^{20} \hat{v}_{k,i} \int\limits_{\Omega_n} \frac{\partial N_i^v}{\partial x_k} N_{n_1}^p dV$$

2.2.5 The global equation system

In the global domain there are N_n nodes and N_e non-overlapping elements. Since the velocities are associated with the global nodes and the pressures are associated with the elements the total number of DOF is $3N_n + N_e$. Similarly there are N_n independent equations arising from the momentum balance in each coordinate direction and N_e independent equations arising from the continuity equation. One can thus obtain the nodal velocities and elemental pressures by solving a linear system with size $3N_n + N_e$. If we define a vector $\boldsymbol{b}$ containing the nodal velocities and the pressures as

$$\boldsymbol{b} = [\hat{v}_{1,1}, \ldots, \hat{v}_{1,N_n}, \hat{v}_{2,1}, \ldots, \hat{v}_{2,N_n}, \hat{v}_{3,1}, \ldots, \hat{v}_{3,N_n}, \hat{p}_1, \ldots, \hat{p}_{N_e}]^{\mathrm{T}} \tag{2.2-35}$$

the structure of the linear system will be

$$\begin{bmatrix} \boldsymbol{A_{11}} & \boldsymbol{A_{12}} & \boldsymbol{A_{13}} & \boldsymbol{C_1} \\ \boldsymbol{A_{21}} & \boldsymbol{A_{22}} & \boldsymbol{A_{23}} & \boldsymbol{C_2} \\ \boldsymbol{A_{31}} & \boldsymbol{A_{32}} & \boldsymbol{A_{33}} & \boldsymbol{C_3} \\ \boldsymbol{D_1} & \boldsymbol{D_2} & \boldsymbol{D_3} & \boldsymbol{0} \end{bmatrix} \boldsymbol{b} = \begin{bmatrix} \boldsymbol{f_1} \\ \boldsymbol{f_2} \\ \boldsymbol{f_3} \\ \boldsymbol{0} \end{bmatrix} \Longrightarrow \boldsymbol{Mb} = \boldsymbol{f} \tag{2.2-36}$$

The sub-systems of the LHS of equation 2.2-36, i.e. $\boldsymbol{A}_{11}, \ldots, \boldsymbol{A}_{33}$, $\boldsymbol{C}_1, \ldots, \boldsymbol{C}_3$ and $\boldsymbol{D}_1, \ldots, \boldsymbol{D}_3$, can be deduced from the discretization of the momentum balance and continuity equation, cf. equation 2.2-30 and 2.2-34. If we take subsystem $\boldsymbol{A}_{11}$ and $\boldsymbol{C}_1$ as examples the elements are given by

$$a_{11}^{I,J} = \sum_{n=1}^{N_e} \int\limits_{\Omega_n} 2\mu \frac{\partial N_{i(I,n)}^v}{\partial x_1} \frac{\partial N_{j(J,n)}^v}{\partial x_1} + \frac{\partial N_{i(I,n)}^v}{\partial x_2} \frac{\partial N_{j(J,n)}^v}{\partial x_2} + \frac{\partial N_{i(I,n)}^v}{\partial x_3} \frac{\partial N_{j(J,n)}^v}{\partial x_3} dV, \qquad I = 1, \ldots, N_n;\ J = 1, \ldots, N_n \tag{2.2-37}$$

and

$$c_1^{I,n} = \int\limits_{\Omega_n} \frac{\partial N_{i(I,n)}^v}{\partial x_1} dV,\ \ I = 1, \ldots, N_n;\ n = 1, \ldots, N_e \tag{2.2-38}$$

Here I and J are global nodes and n is the element number. The coefficients in (2.2-37) are only non-zero on elements which share both node I and J. Similarly the coefficients in (2.2-38) are only non-zero for elements sharing node I. The same applies for all the coefficients in the equation system. The resultant system will therefore be sparse with a sparsity pattern which is related to the numbering of the nodes and the organisation of equations system. If one writes up all the coefficients entering the system it is evident that the total coefficient matrix $\boldsymbol{M}$ is symmetric, i.e. $\boldsymbol{M} = \boldsymbol{M}^{\mathrm{T}}$, which halves the required memory for storage of $\boldsymbol{M}$.

The force terms in (2.2-33) are calculated explicitly and thus enter the RHS of equation 2.2-36.

2.2.6 Numerical integration

In order to evaluate the coefficients in the coefficient matrix $\boldsymbol{M}$ and the force terms in equation 2.2-36 we need to calculate integrals of the type:

$$\int\limits_{\Omega_n} F(\xi(\boldsymbol{x}), \eta(\boldsymbol{x}), \zeta(\boldsymbol{x}))dV \tag{2.2-39}$$

However, the integration boundaries are those of the element in the cartesian coordinate system and can therefore be very complicated. We therefore transform the integral from the cartesian domain to the local (ξ, η, ζ) domain in which the integration can be carried out over the simple regular hexahedral parent element. The transformation to the local coordinate system is carried out using a standard method where the differential volume element $dx_1dx_2dx_3$ can be written as (see also (Zienkiewicz & Taylor 2000)):

$$dx_1dx_2dx_3 = \det \mathbf{J} d\xi d\eta d\zeta \tag{2.2-40}$$

The determinant of the Jacobian matrix is calculated from equation 2.2-20. The integral in (2.2-39) can then be evaluated on the simple parent element, i.e.

$$\int\limits_{-1}^{1}\int\limits_{-1}^{1}\int\limits_{-1}^{1} F(\xi, \eta, \zeta) \det \mathbf{J} d\xi d\eta d\zeta \tag{2.2-41}$$

This greatly simplifies the problem, however, algebraic integration of $\bar{F}$ usually is far from trivial and in most cases 'defies our mathematical skill' as stated by Zienkiewicz & Taylor (2000). To overcome this problem numerical integration is applied. This numerical integration is carried out using gaussian quadrature formulas such that

$$\int\limits_{-1}^{1}\int\limits_{-1}^{1}\int\limits_{-1}^{1} F(\xi, \eta, \zeta) \det \mathbf{J} d\xi d\eta d\zeta = \sum_{i=1}^{Nint} F(\xi_i, \eta_i, \zeta_i) \det \mathbf{J}(\xi_i, \eta_i, \zeta_i) H_i \tag{2.2-42}$$

In equation 2.2-42 $i = 1, \ldots Nint$ correspond to the integration points and H_i are the corresponding quadrature weights. The weights can be found in e.g. (Abramowitz & Stegun 1972) or (Zienkiewicz & Taylor 2000). One advantage of using numerical integration compared to algebraic integration is that the code becomes more general because the same integration algorithm can be used for different types of finite elements. The accuracy of the numerical integration depends on the number of integration points $Nint$ used. It is important that the integration accuracy is sufficiently high in order to obtain the same rate of convergence with mesh refinement as would be obtained if exact integration was used. However, since numerical integration results in an increase in computation time the number of integration points should not be larger than necessary for obtaining the correct convergence rate. From (Zienkiewicz & Taylor 2000) the appropriate integration formula for quadratic elements with C_0 problems is shown to be the $2 \times 2 \times 2$ Gauss point formula (i.e. $Nint = 8$). In the present implementation the user can choose whether to use $2\times2\times2$ or $3\times3\times3$ integration points. And as will be evident later both integration schemes result in the same convergence order. However, the error associated with the 8 point scheme is slightly higher than that associated with the 27 point scheme.

2.2.7 Linear solver and sparse storage scheme

In order to take advantage of the sparsity of the resultant linear system we use the PARDISO sparse linear solver (Shenck, Gärtner & Fichtner 2000, Shenck & Gärtner 2004, Shenck & Gärtner 2006). This solver is available with no charge for academic purposes and can handle symmetric definite and indefinite systems as well as non-symmetric systems. Furthermore the solver is developed for shared memory multiprocessor systems and computation times can therefore be decreased by using multiple processors in parallel. In order to save memory associated with the coefficient matrix only non-zero entries are stored. The storage scheme used is the *compressed sparse row format* and for a symmetric matrix only the upper triangular part is stored. This storage scheme requires 2 pointer arrays and 1 array for storage of the non-zero matrix entries. If we take the symmetric 8×8 matrix given below as an example

$$\begin{bmatrix} a_{11} & a_{12} & 0 & 0 & a_{15} & 0 & a_{17} & 0 \\ a_{12} & a_{22} & a_{23} & 0 & 0 & a_{26} & 0 & 0 \\ 0 & a_{23} & a_{33} & 0 & 0 & 0 & 0 & a_{38} \\ 0 & 0 & 0 & a_{44} & a_{45} & 0 & a_{47} & 0 \\ a_{15} & 0 & 0 & a_{45} & a_{55} & 0 & 0 & a_{58} \\ 0 & a_{16} & 0 & 0 & 0 & a_{66} & a_{67} & a_{68} \\ a_{17} & 0 & 0 & a_{47} & 0 & a_{67} & a_{77} & 0 \\ 0 & 0 & a_{38} & 0 & a_{58} & a_{68} & 0 & a_{88} \end{bmatrix}$$

the non-zero entries of the upper triangle are stored in array A in row order, i.e.

$$\mathrm{A} = [a_{11}, a_{12}, a_{15}, a_{17}, a_{22}, a_{23}, a_{26}, a_{33}, a_{38}, a_{44}, a_{45}, a_{47}, a_{55}, a_{58}, a_{66}, a_{67}, a_{68}, a_{77}, a_{88}]$$

An array IA is used as a pointer for the row indices such that for a given row, I, IA(I) gives the position in A of the first non-zero entry of row I in the upper triangle of the full system. For the above example IA is given by:

$$\mathrm{IA} = [1, 5, 8, 10, 13, 15, 18, 19]$$

The column indices are stored in array JA:

$$\text{JA} = [1, 2, 5, 7, 2, 3, 6, 3, 8, 4, 5, 7, 5, 8, 6, 7, 8, 7, 8]$$

This means that for a given row I in the coefficient matrix the non-zero entries can be found in A(IA(I):IA(I+1)-1) with column indices JA(IA(I):IA(I+1)-1).

2.2.8 Boundary conditions

The types of BC's we apply onto the boundaries of the global domain have been mentioned above. Here some notes on the implementation of velocity BC's and periodic BC's are given. Specification of some or all the velocity components on a boundary results in a decrease in the DOF's and the equations associated with such boundary nodes are removed from the equation system. Furthermore, terms in the remaining equations which arise from nodes, on which velocities are specified, are multiplied by the specified velocity in question and moved to the RHS of the system.

In the case of periodicity source and destination boundaries need to be specified. The nodes on the source boundary are then copied and replaced by the nodes on the destination boundary. This type of boundaries then represent repetitions of the geometry extending from each periodic 'wall'. In Figure 2.3 a sketch of a 2D domain with a regular mesh can be seen. Here the right boundary is the source while left boundary is destination. The Periodicity is in this case simply implemented by making a shift in the node numbering and copying the numbering from the right wall to the left wall.

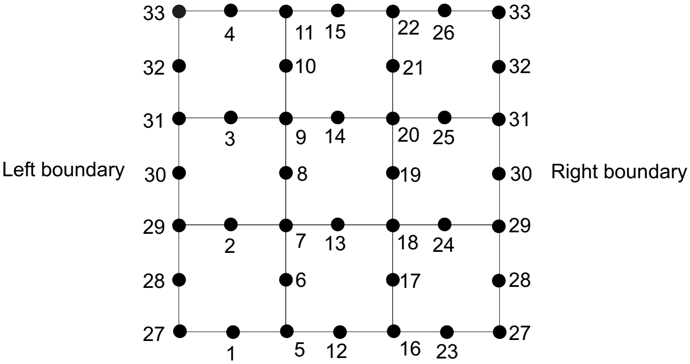


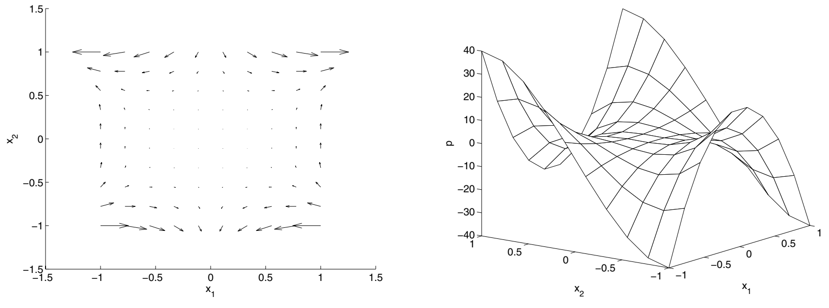
Figure 2.3: Shift in node numbering associated with periodic BC's.

2.3 Convergence of the Stokes solver

The validity of the implemented Stokes solver is checked by investigating the convergence behavior of the solution with respect to mesh refinement. In order to carry out such tests it is necessary to know the 'real' flow field of the benchmark problem to compare the numerical solution with. First an appropriate benchmark problem has to be defined. Here it is important that the problem is of sufficient complexity such that the solver cannot reproduce the solution exactly (to machine precision). Second the exact solution to the problem needs to be evaluated. If possible one can derive the analytical solution otherwise it needs to be evaluated numerically. In the latter case one can choose to generate a reference solution on a very fine mesh and then compare solutions on coarser meshes with the reference solution. The downside to this approach is that even though one obtains correct convergence behavior one cannot be certain that there is no constant bias in the solution. Therefore it is preferable to use a problem where an analytical solution exists. The test problem used in the present project is one which is described in Elman et al. (2005) and which has an analytical solution. The problem is based on a 2D flow field in a square domain and has the following analytical solution:

$$\begin{aligned} v_1 &= 20x_1x_2^3 \\ v_2 &= 5x_1^4 - 4x_2^4 \\ p &= 60x_1^2x_2 - 20x_2^3 + c \end{aligned} \tag{2.3-43}$$

Here c is an arbitrary constant. The flow and pressure fields described by (2.3-43) are illustrated in Figure 2.4.



(a) (b)

Figure 2.4: Velocity field and pressure distribution represented by equation 2.3-43. (a) velocity vector plot, (b) pressure distribution with the constant $c = 0$.

In order to solve this problem using the implemented Stokes solver a rectangular domain with a regular mesh is used. Since the flow in (2.3-43) is 2-dimensional one velocity component, say v_3, is set to zero on all domain boundaries while the velocities on the boundaries perpendicular to the x_1 and x_2 direction are specified according to (2.3-43). The v_1 and v_2 velocity components are not specified on boundaries which are perpendicular to the x_3 direction. In order to make sure that the convergence is correct for all flow directions, the test is performed for all 3 planes of flow, i.e. also cases where either v_1 or v_2 is set to 0. Here only the results from the analysis with v_3 set to 0 is shown as the results showed identical flow fields (to floating point precision) in all planes as expected. In order to characterize the error in the velocities we use the 2-norm of the residuals vector formed by the difference between the analytical and numerical velocities, $\boldsymbol{v}$ and $\hat{\boldsymbol{v}}$ respectively, on element nodes. The pressure error is found as the 2-norm of the residuals between the analytical pressure values calculated in the centre of the finite elements and the numerically calculated pressures (which are constant in each element). In all cases the 2-norm is normalized by the number of DOF for each of the variables. In equation 2.3-44 it is shown how the error in velocity component v_1 is calculated as an example:

$$\text{error}_{\text{v}_1} = \frac{||\boldsymbol{v}_1 - \hat{\boldsymbol{v}}_1||_2}{N_n} \tag{2.3-44}$$

where N_n is the number of velocity nodes. In the plane of the flow the number of elements used is the same in both directions while in the neutral flow direction the number of elements is always kept at 4. The domain size used in the convergence test has the dimension $[-1;1] \times [-1;1]$ in the plane of the flow while the dimension in the neutral direction is scaled such that all the element side lengths are equal. In Table 2.1 the meshes used in the convergence analysis can be seen.

Elements	$4 \times 4 \times 4$	$8 \times 8 \times 4$	$16 \times 16 \times 4$	$32 \times 32 \times 4$	$64 \times 64 \times 4$
Side length h	0.5	0.25	0.125	0.0625	0.03125

Table 2.1: Meshes used in the convergence analysis. Also shown is the corresponding element side length h.

In figure 2.5 the convergence behavior for mesh refinement can be seen. The figure shows the error as a function of the element side length h for velocity components v_1 and v_2 and pressure p. Also shown in the figure is the reference line Error = h^3.

From figure 2.5 it is seen that the convergence order for the error in the velocities is approximately 3 (by comparison with the reference line). On the other hand the convergence order of the pressure error seems to somewhat smaller. Table 2.2 shows the order in the error as the element length h is halved. This data indicates that the convergence order in the velocity approximation approaches 3 as the mesh is refined. Although the convergence order is smaller for the pressure it increases with increasing mesh refinement. This

h	Order in v_1	Order in v_2	Order in p
0.5 → 0.25	3.09	2.56	2.25
0.25 → 0.125	2.11	2.71	2.46
0.125 → 0.0625	2.85	2.84	2.63
0.0625 → 0.03125	2.92	2.92	2.73

Table 2.2: Convergence order of the error with mesh refinement.

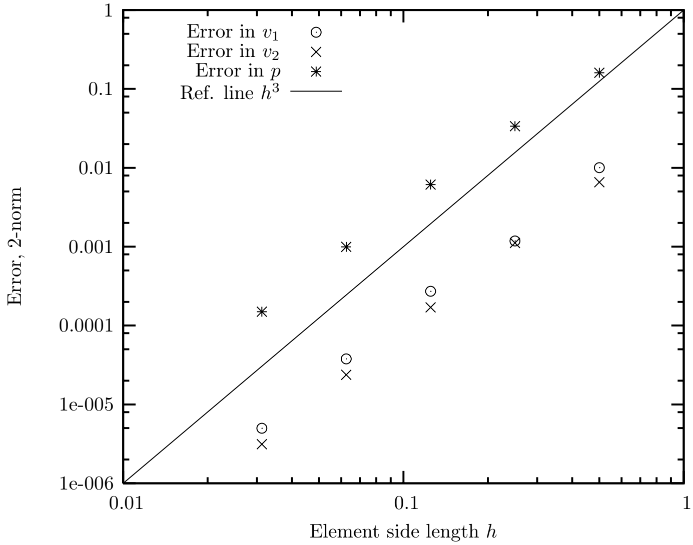


Figure 2.5: Log-log plot of the 2-norm of the error in v_1, v_2 and p as a function of the characteristic element side length h. Also shown is Error = h^3. 27 point integration was used in the calculations.

could indicate that the pressure also exhibits third order convergence. For a Q_p element, i.e. an element with the full Lagrange expansion of order p the expected discretization error is $\mathcal{O}(h^{2(p-m)+1})$ where m is the order of the highest derivative in the expression of interest (Zienkiewicz & Taylor 2000). For $m = 1$ one expects $\mathcal{O}(h^3)$ error for the quadratic element and $\mathcal{O}(h)$ error for constant and discontinuous elements (i.e. P_0). In the present implementation the quadratic serendipity element is used for the velocity approximation as described earlier and thus we do not have the full tri-quadratic expansion for the velocities. However, the observed convergence order in the velocities is 3 and thus we obtain the same order of convergence as would be expected from the full expansion. With regards to the pressures the observed convergence order is considerably higher than expected ($\mathcal{O}(h)$). This behavior cannot be explained at present. However, it is noted that finite elements based on discontinuous pressures have been reported to yield better results than e.g. elements with linear pressures due to the conservation of mass on the element level (Elman et al. 2005). Thus the high convergence order observed might be due to the fact that elements are forces to be divergence free.

The convergence behavior shown in Figure 2.5 and Table 2.2 is based on the 27 point integration scheme. Tests using the 8 point scheme showed similar convergence order, c.f. Table 2.3. However, the error in the velocities was found to be ≈ 0.2% higher for the 8-point scheme compared to the 27-point scheme. The differences in the pressure error for the two integration schemes was found to be in the order of $0.1\% - 0.01\%$.

Integration points	n_{v_1}	n_{v_2}	n_p
8	2.70	2.77	2.52
27	2.69	2.76	2.52

Table 2.3: Observed convergence order using 8-point and 27-point integration. The table shows the exponent of the polynomial fit Error $= ah^n$ to the error in v_1, v_2 and p (all data points are used in the fit).

In summary the convergence tests carried out on the Stokes solver show close to 3rd order convergence for both the velocities and the pressures. This behavior is expected for the velocities while the convergence order for the pressures is considerably higher than expected. Nevertheless the results indicate that the solution obtained from the implemented Stokes solver is valid.

2.4 The Volume of Fluid algorithms

In the previous sections the flow solver was described. The next step is to introduce a second fluid into the computational domain by means of the Volume of Fluid method. Since the flow solver is based on finite element discretization it can handle complex domain geometries. Therefore we will use as much as possible of the finite element framework in our implementation of the VOF method such that this part of the code too can handle complex domains (at least to some degree). The first part of this section will be on the fluid tracking problem in a domain which is subdivided into a finite number of hexahedral VOF cells and where the fluxes across each face of the cells are assumed to be known. Hereafter we will address the problem of calculating the fluid fluxes from velocities specified on the finite element nodes. Here the 2-level algorithm will also be presented as this requires the calculation of fluxes in the interior of the finite elements. Some tests of the advection algorithm are then carried out in predefined flow fields. Finally interfacial tension is introduced into the Stokes solver.

2.4.1 Fluid advection reviewed

The conservation of a scalar quantity Φ convected in a flow field is governed by equation 2.4-1. Although this equation seems relatively simple, it represents a hyberbolic conservation law for which the solution procedure is by no means trivial. The main reason for this is that direct solution procedures usually lead to extensive numerical diffusion such that Φ is smeared out as time proceeds.

$$\frac{\partial \Phi}{\partial t} + \boldsymbol{v} \cdot \nabla \Phi = 0 \tag{2.4-1}$$

If the flow field is divergence free, i.e. $\nabla \boldsymbol{v} = 0$, equation 2.4-1 can be rewritten into:

$$\frac{\partial \Phi}{\partial t} + \nabla \cdot [\Phi \boldsymbol{v}] = 0 \tag{2.4-2}$$

By integrating equation 2.4-2 over a domain Ω and applying the Gauss-Ostrogradskii divergence theorem one obtains:

$$\int_{\Omega} \frac{\partial \Phi}{\partial t} dV = -\int_{\partial\Omega} \Phi \boldsymbol{n} \cdot \boldsymbol{v} dS \qquad (2.4\text{-}3)$$

Here $\partial\Omega$ is the boundary of the domain. If explicit time stepping is used equation 2.4-3 becomes:

$$\Phi^{k+1} = \Phi^{k} - \frac{\Delta t}{V} \int_{\partial\Omega} \Phi \boldsymbol{n} \cdot \boldsymbol{v} dV \qquad (2.4\text{-}4)$$

where Φ^k and Φ^{k+1} are the values of Φ at time step k and $k+1$ respectively and V is the volume of the domain under consideration. From equation 2.4-4 it is clear that the convective transport of Φ amounts to calculating fluxes of Φ across the boundaries of Ω. The advection of the VOF function F is exactly the type of problem described by equation 2.4-4 and it is the calculation of the boundary fluxes which requires special attention in order to avoid smearing. This problem has been adressed by numerous researchers and various solutions procedures have been proposed. Before going into the details of the advection procedure implemented in the present project we will consider some of the algorithms described in the litterature.

In order to avoid numerical smearing VOF advection algorithms include geometrical reconstructions of the interface in order to calculate volume fluxes. The interface reconstruction is based on an approximation to the interface normal and position in interface cells. The various advection algorithms can be divided into two main groups. These are operator split algorithms and non-split algorithms. In the split algorithms equation 2.4-2 is solved in one dimension at a time which requires 3 (or 2 in 2D) advection sweeps per time step. In the non-split algorithms all 3 dimensions are treated simultaneously. The split and unsplit algorithms can then be divided further into Eulerian or Lagrangian type advections.

In its most simple form the reconstruction is based on a local approximation where the interface is considered to be parallel to one of the VOF-cell faces. This method is termed the Simple Line Interface Calculation (SLIC) method and was introduced by Noh & Woodward (1976). This is an operator split algorithm where the interface normal is approximated from a 3×1 block of VOF cells aligned in the advection direction and with the VOF cell under consideration being the center cell in the block. One of the strengths of this algorithm is that multi (more than two) fluid systems can be treated in a relatively easy manner, however, due to the coarse interface approximation the algorithm does not reproduce the interface very accurately, see e.g. (Pilliod & Puckett 2004).

Another algorithm which also is based on interface reconstruction with interface segments aligned with the VOF grid is the method by Hirt & Nichols (1981) which is based on a donor-acceptor scheme. In the algorithm by Hirt & Nichols (1981) the orientation of the interface, which is either horizontal or vertical, is estimated from a 3×3 (in 2D) block of VOF cells. The orientation of the interface is then used for determining whether to use up-winding or down-winding during the advection. The method by Hirt & Nichols (1981) is one-dimensional in nature and thus requires operator splitting. However, an extension to the unsplit 3D case has been carried out by Babaei, Abdollahi, Homayonifa, Varahram & Davani (2006). The interface approximation in the VOF method of Hirt & Nichols (1981) is often termed stair-case reconstruction as the geometrical interpretation of the interface during advection resembles that of a stair case. This method has become widely used because of its relative simplicity. However, one drawback of the method is the production of small fluid bodies which are ejected from the surface of the fluid phase tracked by F, especially in high vorticity flow fields. In the literature these fluid bodies are often termed as flotsam.

A major improvement to the SLIC and stair-case methods are methods in which the interface is reconstructed using line segments (planes in 3D) of arbitrary orientation. DeBar (1974) was the first to describe such a method. This procedure is based on an operator split algorithm where the interface is reconstructed locally in each VOF cell using information of the VOF function in neighbouring cells. However, only neighbouring cells in the advection direction are considered. Thus the procedure for determining the orientation of the line segment is one dimensional. When the interface segments have been set up the advection is carried out in a Lagrangian manner by stretching and compressing the line segments. After this Lagrangian advection step the VOF cell values of F are updated and the advection is carried out in the second direction.

Another line segment method is the Flux Line-Segment Model (FLAIR) of Ashgriz & Poo (1991). However, here the interface segments are set up at the boundaries of neighbouring interface cells. In this method only the two VOF cells shearing the boundary under consideration are used to determine the orientation of the interface segment. Furthermore the advection is carried out in an Eulerian manner where the approximated interface position is used for calculating the volume passing through each VOF cell boundary - without stretching or compressing the interface.

A more recent and very popular line (or plane) segment method is the Piecewise Linear Interface Calculation (PLIC) method described in (Gueyffier, Lie, Nadim, Scardovelli & Zaleski 1999) for the 3D case. Although the basic idea behind the PLIC method resembles that of DeBar (1974) it is more formalized. The PLIC interface reconstruction consists of approximating the normal to the interface from the gradient of the F function, i.e. $\boldsymbol{n} = \nabla F$, in the VOF cell under consideration (usually in its centre). The interface plane is then oriented perpendicular to this normal. Using the known orientation the plane is hereafter positioned in the cell such that the volume enclosed by the plane and the boundaries of the VOF cell satisfies the cell value of F. A general procedure for calculating the segment position is given in (Gueyffier et al. 1999) both for 2D and 3D problems. This problem will also be addressed in the next section of this thesis.

Various methods exist for calculating the gradient of F and in (Pilliod & Puckett 2004) a very thorough evaluation of a number of these methods is made in terms of convergence tests. Pilliod & Puckett (2004) also show that a criterion for a PLIC reconstruction to be second order accurate with spatial refinement requires that lines in 2D and planes in 3D can be reproduced exactly. Some of the most widely used numerical schemes to approximate the interface normal are the Youngs method (Youngs 1982), the Parker and Youngs method (Parker & Youngs 1992), the Least squares Volume-of-fluid Interface Reconstruction Algorithm (LVIRA) by Puckett (1991) and the Efficient Least squares Volume-of-fluid Interface Reconstruction Algorithm (ELVIRA).

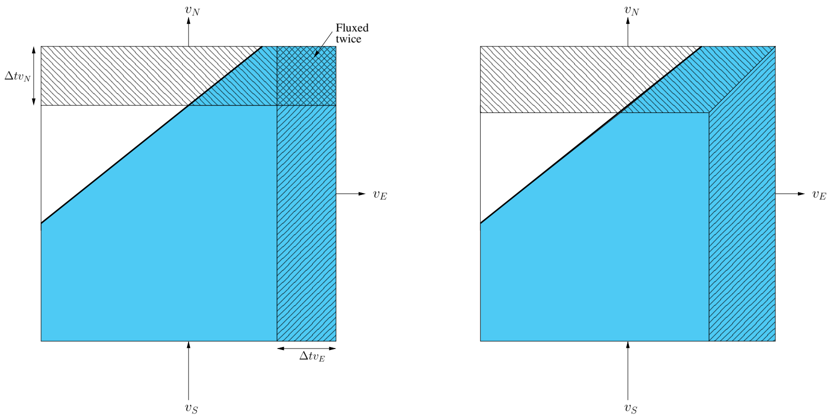
In the Youngs method the normal is first approximated on the vertices of the VOF cell under consideration using a finite difference scheme. Then the interface normal in the VOF cell is taken as the average of the vertex normals and thus information from a $3 \times 3 \times 3$ block of cells (in 3D) is used for the normal approximation.

In the Parker and Youngs method a 6-point finite difference stencil is used (4 point stencil in 2D). However, again information from $3 \times 3 \times 3$ block of cells is used to generate average F values at the stencil points. Furthermore a weighting parameter α is introduced in the calculations of average F values.

The analysis by Pilliod & Puckett (2004) shows that the Parker and Youngs method does not lead to exact approximations of lines or planes and is at best $\mathcal{O}(h^1)$ order accurate.

The LVIRA method is based on a least squares minimization of an error function associated with a reconstructed interface segment. This method requires an iterative scheme and thus is more computationally expensive than the Youngs and Parker & Youngs methods. However, as shown in (Pilliod & Puckett 2004) it is fully second order accurate and is able to reconstruct planes (and lines) accurately (to the tolerance used in the iterative scheme). In the ELVIRA method a number of candidates for the interface normal are specified (6 in 2D). A minimization procedure is then used for obtaining the best candidate for the normal. The ELVIRA method is second order accurate.

When the interface has been reconstructed, i.e. when interface normals and interface segment positions have been calculated, the next step is to advect the F-function in the flow field. Since the PLIC method is 3D in nature the advection can either be carried out using an operator split method, e.g. (Gueyffier et al. 1999, Pilliod & Puckett 2004, Scardovelli & Zaleski 2003) or and un-split method, e.g. (Harvie & Fletcher 2000, Harvie & Fletcher 2001, Pilliod & Puckett 2004, Biausser, Guignard, Marcer & Fraunié 2004). Furthermore the advection can be carried out using a purely Lagrangian method (Gueyffier et al. 1999, Biausser et al. 2004), a purely Eulerian method (Pilliod & Puckett 2004, Harvie & Fletcher 2000, Harvie & Fletcher 2001) or a mixture of both (Scardovelli & Zaleski 2003) (Implicit Eulerian-Explicit Lagrangian). One of the problems of solving the advection problem with an unsplit scheme is to avoid fluxing the same fluid across two cell faces in the same time step. An example of this problem is sketched in Figure 2.1 (a) where the total volume which leaves the cell through the north, N, and east, E, boundaries during time step Δt is marked by the patterned rectangles (calculated by an Eulerian scheme). The fluid that leaves the cell then corresponds to the overlap between the patterned rectangles and the filled area, i.e. the area enclosed by the PLIC interface and the cell boundaries. However, from the figure it is clear that part of the volume is fluxed through both the north and east boundary. In (Harvie & Fletcher 2001) this problem is overcome by geometrical considerations which include cell face velocities. Figure 2.1 (b) outlines how the flux problem can be avoided. Other methods include integration along streamlines (Harvie & Fletcher 2000) (the Stream-Scheme) and Lagrangian methods (Biausser et al. 2004).



(a) (b)

Figure 2.1: Sketch showing Eulerian multidimensional 2D flux calculations. (a) Example of a scheme where part of the volume (area in 2D) is fluxed through two faces during a given time step Δt and (b) a corrected flux scheme to avoid this problem.

Although the unsplit methods are efficient they are complex to implement for 3D problems due to geometrical nature of the problem. Furthermore some split algorithms perform almost as well as the unsplit counterparts except maybe for problems which involve sharp corners (Pilliod & Puckett 2004). Furthermore if a second order PLIC reconstruction algorithm is used one can obtain second order accuracy in split algorithms by alternating the advection direction, e.g. (Pilliod & Puckett 2004, Scardovelli & Zaleski 2003).

A parabolic reconstruction method is described in (Renardy, Cristini & Li 2002). In this method, termed Parabolic Reconstruction Of Surface Tension (PROST), a parabolic equation is fitted through a 5 × 5 × 5 block of VOF cells. This method leads to very accurate interface reconstructions and has been developed in order to calculate interfacial tension forces accurately. Also Ginzburg & Wittum (2001) describe a higher order reconstruction algorithm, however this algorithm is based on a combination of PLIC and spline interpolants.

Now we will turn to our own implementation of surface reconstruction and fluid advection.

2.4.2 Surface reconstruction and fluid advection

We have chosen to base our own implementation on a split operator scheme with PLIC interface reconstruction. Furthermore the approximation used for the interface normal is somewhat similar to the Youngs method. The main reasons for these choices are:

- Operator split schemes are much easier to implement than their un-split counterparts - especially for 3D problems. This out weights the small difference in accuracy between the two methods.
- Generalized procedures for interface positions calculations are readily available for the PLIC method.
- The Youngs method for calculating interface normals can be directly extended to the finite element methodology. It is though noted that it is only $\mathcal{O}(h^1)$ accurate.

Our split advection algorithm is based on the purely Lagrangian approach described in (Gueyffier et al. 1999) with some modifications in order to improve volume conservation. Here we will start by outlining how the interface normal is calculated on a general mesh.

2.4.2.1 Evaluation of $\boldsymbol{n}$

In order to evaluate the interface normal in a given interface VOF cell, which we hereafter call the active cell, the F-field is interpolated onto vertices of the cell. First the active cell and all its neighbouring cells are identified which leads to a $3 \times 3 \times 3$ block with the active cell in the centre. A ghost mesh is then overlaid this block of cells such that the vertices of the ghost mesh elements are positioned in the geometrical centre of each VOF cell. This leads to a ghost mesh with size $2 \times 2 \times 2$. In Figure 2.2 a sketch shows the procedure (in 2D for simplicity). In the figure full lines represent VOF cells while broken lines represent the ghost mesh. The idea is now to use the ghost mesh to make a mapping from the cartesian coordinate system to the local curvilinear system ξ, η, ζ. In order to do this we use tri-linear finite element shape functions which are defined in terms of the ghost mesh vertices and are given by (Zienkiewicz & Taylor 2000):

$$L_i = \frac{1}{8}(1+\xi_0)(1+\eta_0)(1+\zeta_0), \quad i = 1, \ldots, 8 \tag{2.4-5}$$

where

$$\xi_0 = \xi\xi_i, \;\; \eta_0 = \eta\eta_i, \;\; \zeta_0 = \zeta\zeta_i$$

We will use the ghost element in the lower right corner in Figure 2.2 as an example, c.f. also Figure 2.3. The values of the F-function on the ghost element nodes are then given by the F-values in corresponding VOF cells, i.e. cells 4,5,7 and 8 in Figure 2.3. The F

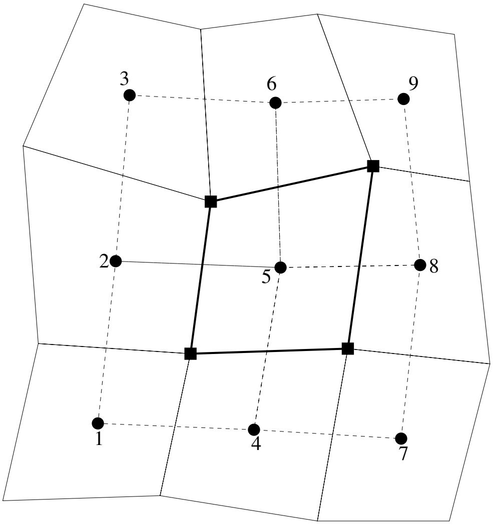


Figure 2.2: A 2d example of the 3 × 3 block of VOF cells used for calculating values of F on vertices of the central active VOF cell. Solid lines represent the VOF mesh and broken lines represent the ghost mesh used for the mapping to curvilinear coordinates. ●'s represent linear nodes on the ghost mesh and ■'s represent the local linear nodes on the active VOF element.

function can therefore be interpolated everywhere within the ghost element as (for the 3D case):

$$F(\boldsymbol{x}(\xi, \eta, \zeta)) = \sum_{i=1}^{8} L_i(\xi, \eta, \zeta) F_i \tag{2.4-6}$$

where F_i are the values of F on the ghost cell nodes.

In order to use equation 2.4-6 to evaluate F on the VOF cell node with global coordinate $\boldsymbol{x}$ we need to know the position of the node in the local coordinate system, i.e. (ξ, η, ζ) cf. Figure 2.3. Since we use an isoparametric transformation we can write:

$$\boldsymbol{x}(\xi, \eta, \zeta) = \sum_{i=1}^{8} L_i(\xi, \eta, \zeta) \boldsymbol{x}_i \tag{2.4-7}$$

where $\boldsymbol{x}_i$ are the global coordinates for the ghost element node points. For the point $\boldsymbol{x}$ within the ghost element in question the corresponding local coordinates in the curvilinear system can then be found by solving equation 2.4-7 as a set of 3 equations with the 3 unknowns ξ, η and ζ. However, equation 2.4-7 leads to a nonlinear problem which includes terms of the form $\xi\eta\zeta$ and thus the problem cannot be solved directly. In our

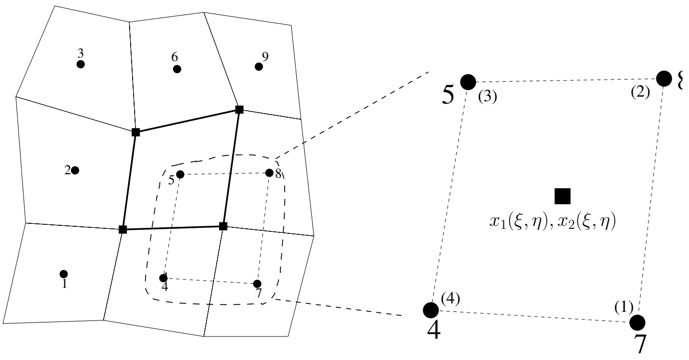


Figure 2.3: Example of a ghost mesh element used for calculating the value of F at a vertice on the active VOF cell. Numbers in brackets are local node numbers on the ghost element.

implementation the solution is obtained iteratively using a Newton-Raphson scheme. Thus in summary the nodal value of F is found by first calculating the position of the VOF cell node in the local coordinate system and then interpolating F according to (2.4-6). This procedure is repeated for all the vertex points of the active VOF cell. The interface normal is then calculated (in the center of the active cell) as:

$$n_k = \sum_{i=1}^{8} \left. \frac{\partial L_i}{\partial x_k} \right|_{(0,0,0)} F_i, \quad k = 1, \ldots 3 \tag{2.4-8}$$

where F_i and $\frac{\partial L_i}{\partial x_k}$ here are associated with the VOF cell nodes.

2.4.2.2 Interface segment position

When the interface normal is estimated in all interface VOF cells the next step is to set up the corresponding interface plane segments. For a given interface VOF cell the orientation of the interface plane segment is perpendicular to the normal and thus is locked. The position of the plane segment is then determined by requiring that the volume bounded by the interface segment and the boundaries of the VOF cell agrees with the value of F in the considered cell. This evaluation is entirely geometric in nature as it requires the calculation of intersections between planes and evaluation volumes of polyhedras. In a non-regular mesh defined in the global coordinate system this would involve some rather complex geometry. Therefore we will instead base these calculations on a square parent VOF cell defined in the local coordinate system (ξ, η, ζ). This parent VOF cell is equivalent to the parent finite element in Figure 2.2 with its vertex positions at $([-1, 1] \times [-1, 1] \times [-1, 1])$. First the interface normal is transformed from the global to the local coordinate system as:

$$\boldsymbol{n}_{(\xi,\eta,\zeta)} = \boldsymbol{J}|_{(0,0,0)} \cdot \boldsymbol{n}_{(\boldsymbol{x})} \tag{2.4-9}$$

where $\boldsymbol{J}$ is the Jacobian transformation matrix as defined earlier, cf. equation 2.2-19 (with N_i interchanged with L_i). In fact this means that no transformation between the global and local coordinate system is necessary when the normal is calculated as described above, cf. equation 2.4-8. Thus in this case the normal in terms of local coordinates is calculated simply as

$$n_\xi = \sum_{i=1}^{8} \left. \frac{\partial L_i}{\partial \xi} \right|_{(0,0,0)} F_i \tag{2.4-10}$$

and similarly for the other local coordinate directions. However, we will later use equation 2.4-8 in connection with interfacial tension calculations. Let us now use $\boldsymbol{m}$ to designate the interface normal in the local coordinate system. All planes with normal $\boldsymbol{m}$ can then be described by:

$$m_1\xi + m_2\eta + m_3\zeta = \alpha \tag{2.4-11}$$

where the parameter α determines the position of the plane. In (Gueyffier et al. 1999) and in (Scardovelli & Zaleski 2000) a general formula for calculating the volume enclosed by an arbitrary plane and 3 or more sides of a cuboid with side lengths c_1, c_2, c_3 is given. This formula is restated in equation 2.4-12 below:

$$\begin{aligned} V = &\frac{1}{6m_1m_2m_3} \left[\alpha^3 - \sum_{k=1}^{3} H(\alpha - m_kc_k)(\alpha - m_kc_k)^3 \right. \\ &\left. + \sum_{k=1}^{3} H(\alpha - \alpha_{max} + m_kc_k)(\alpha - \alpha_{max} + m_kc_k)^3 \right] \end{aligned} \tag{2.4-12}$$

In equation 2.4-12 $H(.)$ is the Heaviside step function and $\alpha_{max} = m_1c_1 + m_2c_2 + m_3c_3$. Furthermore it is noted that when using equation 2.4-12 it is assumed that all components of the normal vector $\boldsymbol{m}$ are positive. The geometric interpretation of equation 2.4-12 is as follows: If we take a given plane $m_1\xi + m_2\eta + m_3\zeta = \alpha$ with positive m_i and positive α and plot it in the coordinate system ξ, η, ζ with origin $(0, 0, 0)$ it will cut the coordinate axes at $(\alpha/m_1, \alpha/m2, \alpha/m3)$, cf. Figure 2.4. Thus the total volume bounded by CIJK in Figure 2.4 is given by:

$$V_{tot} = \frac{1}{6}\frac{\alpha^3}{m_1m_2m_3}$$

which gives the first term in the brackets on the RHS of equation 2.4-12. If now a cuboid with side lengths c_1, c_2 and c_3 is placed with its sides parallel to the coordinate axes the plane will cut some of the sides of this cuboid if $\alpha < \alpha_{max} = m_1c_1 + m_2c_2 + m_3c_3$. Figure 2.5 shows an example where $\alpha < \alpha_{max}$.

With reference to Figure 2.5 the 'cut volume', or volume bounded by the plane and the sides of the cuboid is given by ABCDLMNK. This volume is obtained by subtracting the volume of tetrahedras OJBN and IPDL from the total volume V_{tot}. These volumes are calculated by the second term in the brackets of the RHS of equation 2.4-12. However,

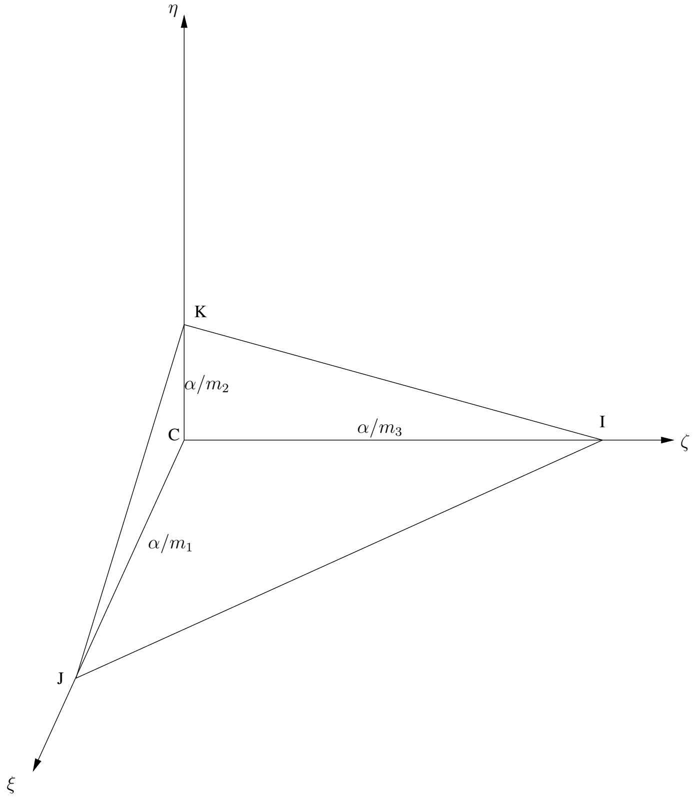


Figure 2.4: Intersection between the coordinate axes and the plane $m_1\xi + m_2\eta + m_3\zeta = \alpha$.

the volume AOPM is subtracted twice and needs to be added back which is done by the third term in equation 2.4-12. The Heaviside step function is used for only subtracting (and adding back) volumes outside the cuboid.

Given the normal vector and parameter α it is thus possible to calculate the volume bounded by the plane segment and sides of a VOF cell using equation 2.4-12. However, until now we only know the orientation of the plane but not its position so it is the inverse problem which is of interest, namely to calculate the position (α) when we know the volume (from F) and the normal. This means that we need to solve equation 2.4-12 for α. This can be done numerically using e.g. bisection (first order) or Newton's method (second order). Bisection is used in the current implementation due to its robustness (this should though be changed in the future). A good initial guess for the numerical procedure is obtained by the following considerations: First we rearrange the indices such that

$$m_1c_1 \leq m_2c_2 \leq m_3c_3$$

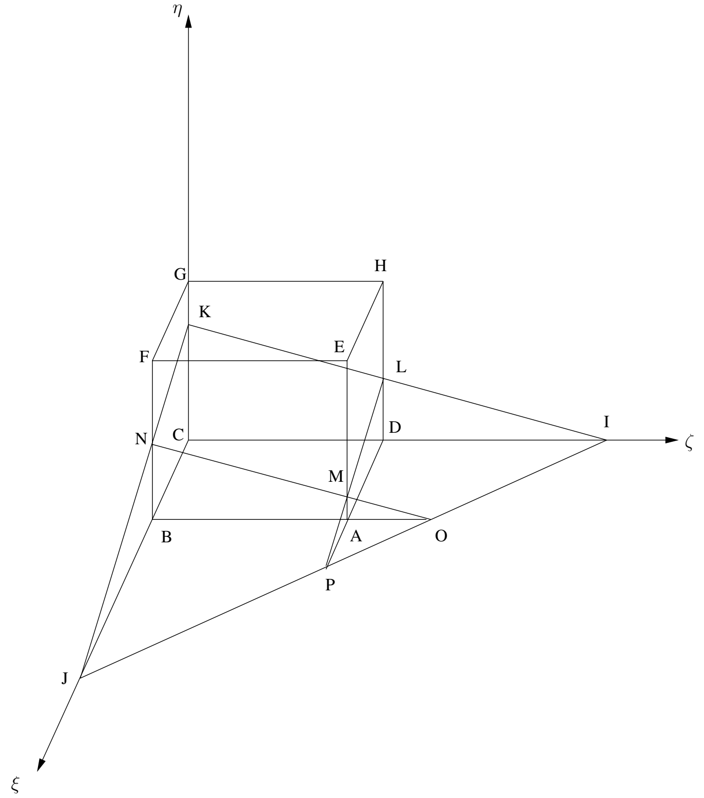


Figure 2.5: Illustration showing the 'cut volume' calculated by equation 2.4-12. The volume enclosed by the plane and the cuboid is given by ABCDLMNK. Figure is redrawn after (Gueyffier et al. 1999).

Then we first set $\alpha = m_1c_1$ which means that the plane will go through the point $(\xi, \eta, \zeta) = (c_1, 0, 0)$, but not protrude the sides of the element, cf. Figure 2.6 (a). If now the known volume V is less than (or equal to) the volume V^* calculated by equation 2.4-12 with $\alpha = m_1c_1$ then we know that $0 < \alpha \leq m_1c_1$. If this is not the case we set $\alpha = c_2m_2$, cf. Figure 2.6 (b), compute the volume again and compare it to the known volume. In case $V < V^*$ we know that $m_1c_1 < \alpha \leq m_2c_2$. This procedure is repeated until we find the correct range of α, cf. Figure 2.6 (a)-(g). When the appropriate range is found it is used as input for the bisection routine.

In case one of the normal components is zero equation 2.4-12 cannot be used. However, since this problem corresponds to the case where the plane is parallel to one of the element sides it can be treated using the 2 dimensional equivalent to equation 2.4-12 which is given

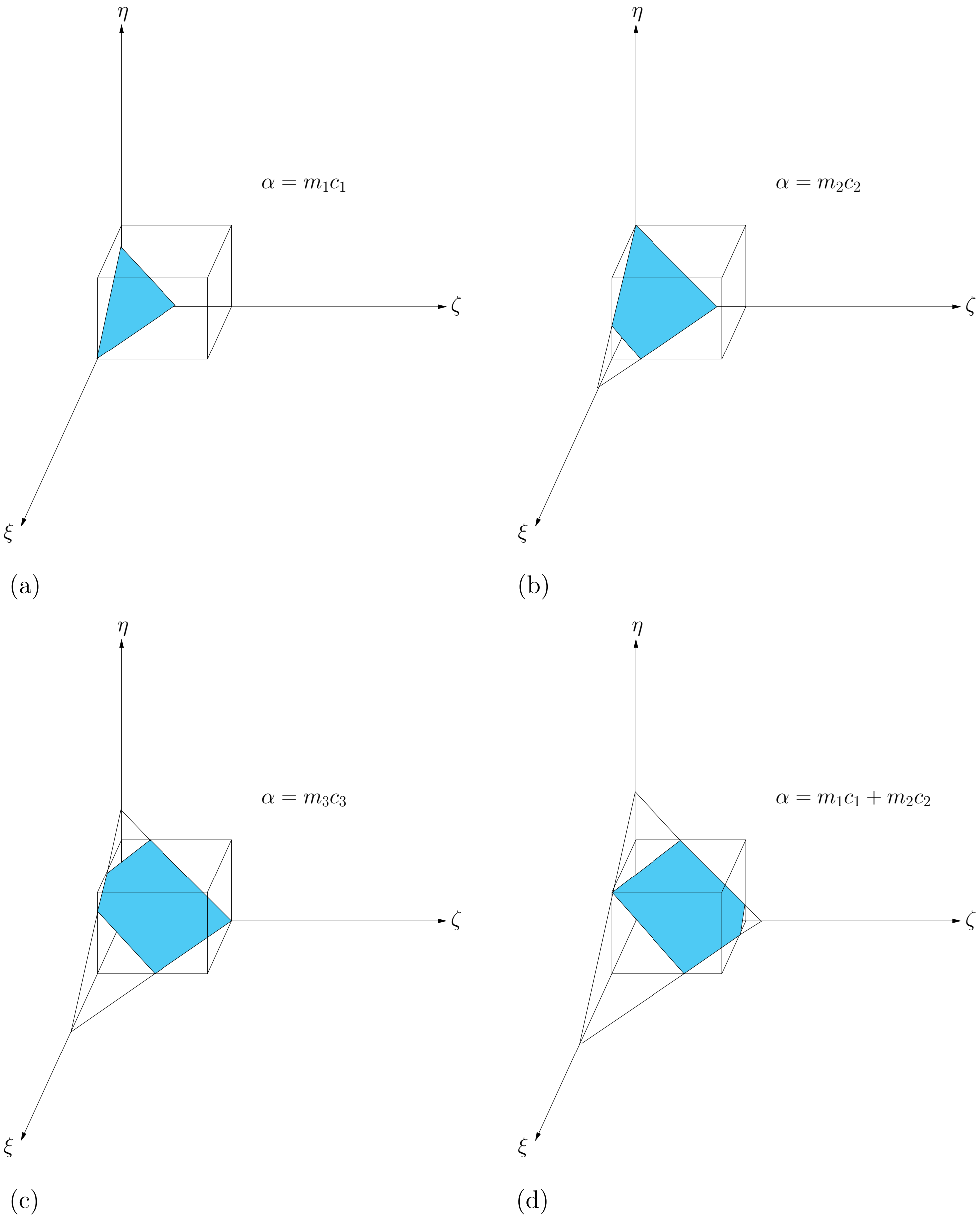


Figure 2.6: Plane positions at various values of α.

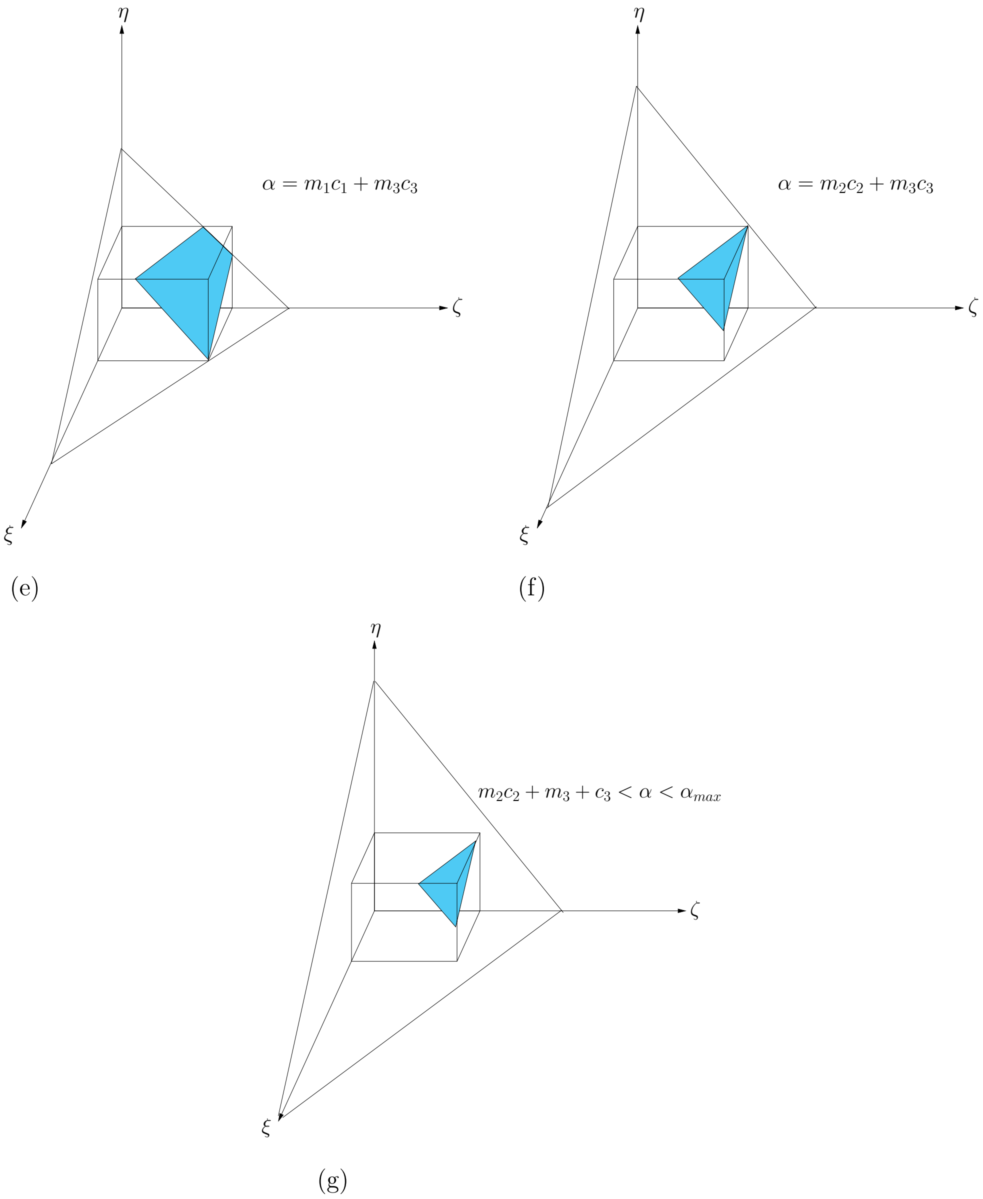

η
$\alpha = m_1c_1 + m_3c_3$
ζ
ξ
(e)
η
$\alpha = m_2c_2 + m_3c_3$
ζ
ξ
(f)
η
$m_2c_2 + m_3 + c_3 < \alpha < \alpha_{max}$
ζ
ξ
(g)

Figure 2.6: Continued.

by e.g. Scardovelli & Zaleski (2000). If we arrange indices such that m_3=0 the 'cut volume' is given by:

$$V = \frac{1}{2m_1m_2}\left[\alpha^2 - \sum_{k=1}^{2} H(\alpha - m_kc_k)(\alpha - m_kc_k)^2\right] c_3 \qquad (2.4\text{-}13)$$

Since this is a second order polynomial in α it it solved analytically. First it is checked how many terms in the square bracket on the RHS of equation 2.4-13 need to be taken into account (there are 3 cases) and then the appropriate polynomial is solved. In practice a normal component is treated as zero if it is smaller than some tolerance *tol*.

The last and simplest case is where 2 components of the normal vector are zero, say m_2 and m_3, in which case $\alpha = (Vm_1)/(c_2c_3)$.

It is noted that when calculating α the absolute values of the normal components are used. The correct plane segment position and orientation is then obtained by proper rotation and/or mirroring. Furthermore, the positioning of the interface segment is only based on the value of the F in the active cell and thus there is no requirement that the interface is continuous across VOF cell boundaries. This means that generally there will be a discontinuity of the interface segments across each cell boundary.

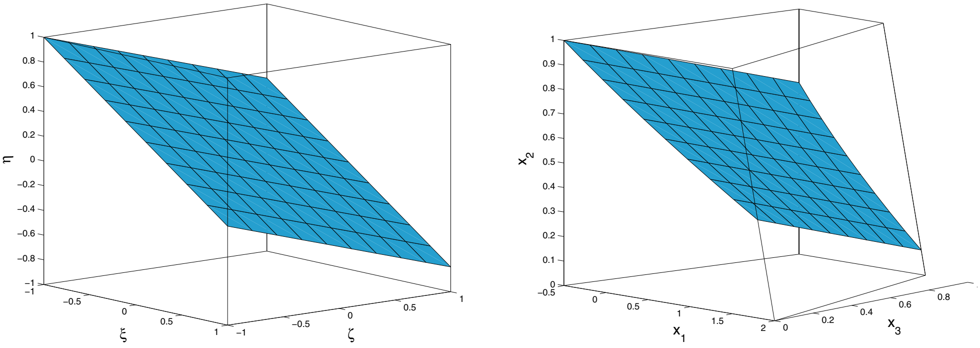
Due to the mapping from the global to local coordinate system the geometrical problems associated with the calculation of the 'cut volume' in the global mesh are avoided. However, the fact that the PLIC reconstruction is made in the parent element means that if the corresponding element in the global domain is distorted, then if one transforms the PLIC face from the parent element back to the global element the interface segment will be curved to some degree - dependent on the degree of distortion of the element. In Figure 2.7 this is illustrated where in figure (a) a plane with $\boldsymbol{m} = [0.6\ 1.0\ 0.3]^{\mathrm{T}}$ and $\alpha = 1.0$ is shown and in figure (b) the corresponding segment is shown in a distorted VOF cell. The transformation from the local to global coordinates is carried out by using:

$$x_k^s = \sum_{i=1}^{8} L_i(\xi^s, \eta^s, \zeta^s)x_{k,i}, \quad k = 1, \ldots, 3 \qquad (2.4\text{-}14)$$

where x_k^s and ξ^s, η^s, ζ^s refer to plane segment coordinates in the global and local coordinate system respectively. In practice this means that, unless the VOF mesh consists of cells with only right angles, the reconstruction leads to curved surfaces in the global domain. However, since the reconstruction is only performed with the purpose of advecting the F-function and the advection is also carried out in the local coordinate system, the interface is never reconstructed in the global coordinate system (except for plotting purposes).

2.4.2.3 Advection

In order to facilitate the explanation of the advection algorithm we will begin by looking at the 2D case for a mesh with rectangular elements. As mentioned earlier the implementation is based on the purely Lagrangian approach using a split operator technique. We start here



(a) (b)

Figure 2.7: (a) interface segment reconstructed using PLIC. (b) interface segment when mapped back to the global coordinate system.

by restating the procedure described in (Gueyffier et al. 1999). Let us take an advection step with time step Δt along the x_1 direction as an example. Suppose we have an interface VOF cell with fluxes f_W and f_E on the west and east (left and right) boundaries then the corresponding boundary flux velocities are found as $v_W = f_W/\int_{\Gamma_W} dS$ and $v_E = f_E/\int_{\Gamma_E} dS$ where the integral just is the length (area in 3D) of the cell boundary, see also Figure 2.8 top. It is then assumed that the velocity inside the cell varies linearly along the x_1 direction. If for simplicity it is assumed that the west cell face has coordinate $x_W = 0$ and the east face has coordinate $x_E = h$ then the velocity inside the cell in the x_1 direction is given by:

$$v_1(x_1) = v_W\left(1 - \frac{x_1}{h}\right) + v_E\frac{x_1}{h} \qquad (2.4\text{-}15)$$

For points initially at x_1 it is assumed that the velocity is constant and given by equation 2.4-15 during the advection step. This means that the x_1-coordinate of all points on the interface segment will change to new values given by:

$$x_1^* = x_1 + v_1(x_1)\Delta t = \left[1 + \frac{v_E - v_W}{h}\Delta t\right] x_1 + v_W\Delta t \qquad (2.4\text{-}16)$$

During the advection, coordinate x_2 remains constant (in 3D x_3 is also constant). If the initial coordinate x_1 is isolated in equation 2.4-16 one obtains:

$$x_1 = \frac{x_1^* - v_W\Delta t}{1 + ((v_E - v_W)/h)\Delta t)} \qquad (2.4\text{-}17)$$

The original line segment before advection is given by:

$$m_1x_1 + m_2x_2 = \alpha \qquad (2.4\text{-}18)$$

If now x_1 from equation 2.4-17 is inserted into equation 2.4-18 one obtains:

$$m_1 \left[\frac{x_1^* - v_W \Delta t}{1 + ((v_E - v_W)/\Delta t)} \right] + m_2 x_2 = \alpha \Longrightarrow m_1^* x_1^* + m_2 x_2 = \alpha^* \tag{2.4-19}$$

where

$$m_1^* = \frac{m_1}{1 + ((v_E - v_W)/h)\Delta t} \tag{2.4-20}$$

and

$$\alpha^* = \alpha + \frac{m_1 v_W \Delta t}{1 + ((v_E - v_W)/h)} \tag{2.4-21}$$

Hence the advection changes the m_1 component of the interface segment normal into m_1^* and the position α into α^*. In Figure 2.8 an example shows the advection procedure. The top figure shows an interface segment before advection. In the bottom figure the interface has been advected with the above described method. Here the left and right edges of the interface are advected the distances $\Delta t v_W^*$ and $\Delta t v_E^*$ respectively during time step Δt. The velocities v_W^* and v_E^* are those interpolated onto the edges of the interface before advection.

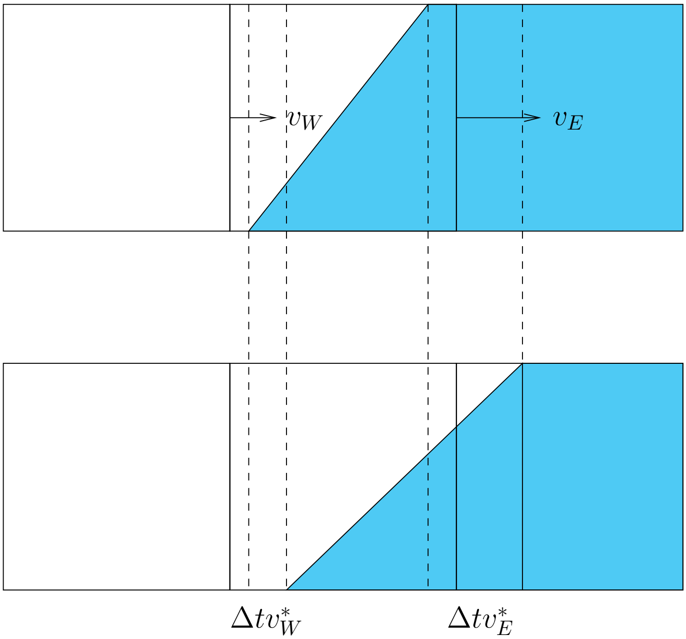


Figure 2.8: 2D split operator Lagrangian advection. Top illustration shows an interface line segment before advection and bottom figure shows the interface after advection.

It is straight forward to extend the procedure to the 3D case, since, only one direction is considered at a time. Thus in 3D equations 2.4-19 is just replaced by its 3D counterpart. When an interface segment has been advected in a given VOF cell it is necessary to calculate the volume which has left the cell if the interface has protruded into neighbouring cells. Furthermore the remaining volume in the cell also needs to be evaluated. These volumes are calculated using equation 2.4-12 with the new values m_1^* and α^* and appropriate values of c_1, c_2 and c_3, see e.g. (Gueyffier et al. 1999) for more details. When this procedure has been carried out for all active VOF cells we know what volume remains in the cells and what volume has left the cells. The final step is to update F by adding the volume entering from the neighbouring cells. After advection is carried out completely in one direction the interface normals and positions are updated and the advection is carried out in the second direction (and then in the third direction for 3D problems). Thus for 3D problems each time step requires three runs through the PLIC and advection algorithms.

One of the main advantages with the Lagrangian approach is that the requirement $0 \leq F \leq 1$ is always fulfilled and thus no extra care has to be taken in order to avoid overshoots or undershoots in F. However, one problem with the method is that it does not conserve volume accurately. This is a severe drawback especially in our case where we want to simulate the deformation of dispersed droplets because if the volume changes the effective capillary number also changes. It is easy to show that the method is not volume conserving by considering a simple 2D example consisting of a square VOF cell with side lengths h and origin at $(0,0)$, cf. Figure 2.9 (a). Let us assume that the velocities on the east and north boundaries and time step are such that $v_E \Delta t = -v_N \Delta t = 1/4h$ while the west and south velocities are zero. Furthermore the components of the outward directed normal vector are $m_1 = 1$ and $m_2 = 1$ and $\alpha = 1/2h$. Then if advection is carried, say, in the x_1 direction, the component of the normal vector will, according to equation 2.4-19, change to $m_1^* = 1/(1+1/4)$ while $\alpha^* = \alpha$. Hence due to the stretching of the interface the interface will now intersect with the south boundary of the VOF cell at $1/2h + 1/4h$. Similarly advection in the x_2 direction will move the line segment downwards such that it intersects the west boundary of the VOF cell at $1/2h - 1/4h$, see Figure 2.9 (b). The initial value of F in the VOF cell was $1/8$ while it after advection is $3/32$ and thus volume is lost even though the sum of the cell boundary fluxes is zero.

In order to overcome the problem associated with volume conservation in the split Lagrangian approach Scardovelli & Zaleski (2003) device a scheme which is based on a combined Eulerian-Lagrangian procedure. In this scheme advection is carried out using Eulerian flux calculations in the first direction and Lagrangian flux calculation in the second direction. The details of the scheme will not be restated here, however, the procedure does indeed result in exact volume conservation. It is though noted here that this method is only applicable to 2D problems because using either the Eulerian or Lagrangian method in the third direction would destroy the volume conservation. One might, however, improve the average volume conserving character of the method by alternating between the Eulerian and Lagrangian approach in the third direction.

In stead of using a mixture of the Eulerian and Lagrangian method we will make a

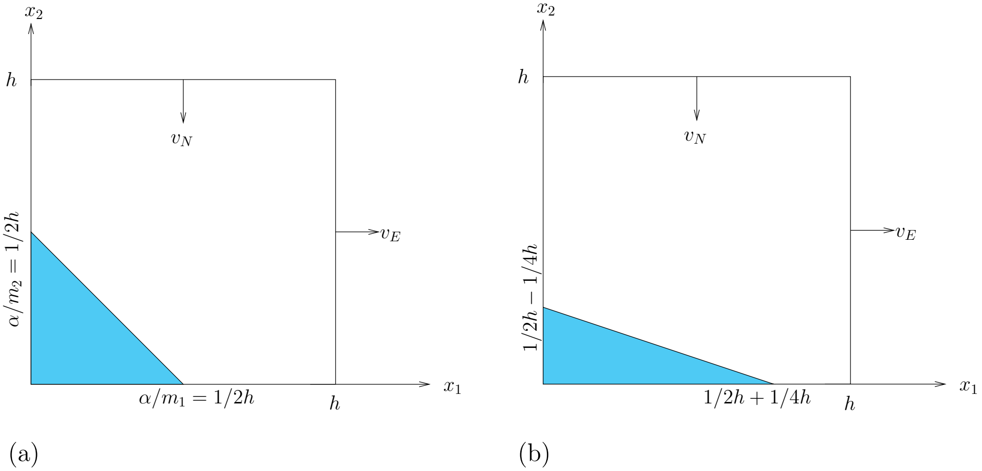


Figure 2.9: Sketch of a simple example showing that the Lagrangian advection algorithm in (Gueyffier et al. 1999) is not volume conserving. (a) before advection, (b) after advection in both the x_1 and the x_2 direction.

modification to the assumption that the velocity of a point with initial position x_1 will have a constant velocity during the advection step. Let us say that we have the situation depicted in Figure 2.8 top and we want to advect the interface in, say, the x_1-direction. If we still assume that the velocity varies linearly from the west to the east wall but now instead say that as a given interface point initially at x_1, moves with the flow it changes velocity due to linear dependence of velocity on position. At any position x_1 the velocity is given by equation 2.4-15 thus one can write:

$$\frac{dx_1}{dt} = v_W\left(1 - \frac{x_1}{h}\right) + v_E\frac{x_1}{h} \tag{2.4-22}$$

Solving this first order ODE with initial condition $x_1(t=0) = x_1$ yields

$$x_1^* = x_1 \exp\left(\frac{v_E - v_W}{h}\Delta t\right) + \frac{hv_W}{v_E - v_W}\left(\exp\left(\frac{v_E - v_W}{h}\Delta t\right) - 1\right) \tag{2.4-23}$$

If now a new time step variable Δt^* is defined as

$$\Delta t^* = \frac{h}{v_E - v_W}\left(\exp\left(\frac{v_E - v_W}{h}\Delta t\right) - 1\right) \tag{2.4-24}$$

then equation 2.4-23 can be written as

$$x_1^* = \left[1 + \frac{v_E - v_W}{h}\Delta t^*\right] x_1 + v_W\Delta t^* \tag{2.4-25}$$

which is identical to equation 2.4-16 but with Δt replaced by Δt^*. Thus we can now use equation 2.4-20 and 2.4-21 but with Δt replaced by Δt^*. However, we note that Δt^* is undefined when $v_E = v_W$ in which case $\Delta t^* = \Delta t$ should be set. If we return to the simple test problem outlined in Figure 2.9 (a) and calculate Δt^* for the x_1 direction we obtain:

$$\Delta t^*_{x_1} = \frac{h}{v_E - v_W}\left(\exp\left(\frac{1}{4}\right) - 1\right)$$

where $v_E \Delta t = 1/4$ has been used. Inserting $\Delta t^*_{x_1}$ in place of Δt in equation 2.4-20 yields

$$m_1^* = \exp(-1/4)$$

while $\alpha^* = \alpha$ since $v_W = 0$. This means that the line segment intercepts the south boundary of the cell at $x_1^* = 1/2h\exp(1/4)$. Similarly Δt^* in the x_2 direction is given by

$$\Delta t^*_{x_2} = \frac{h}{v_N - v_S}\left(\exp\left(-\frac{1}{4}\right) - 1\right)$$

where now $v_N \Delta t = -1/4$ has been used. Thus m_2^* becomes

$$m_2^* = \exp(1/4)$$

which means that the interface will intercept the west boundary of the cell at $x_2^* = 1/2h\exp(-1/4)$. Hence the area enclosed by the line segment after advection is

$$\frac{h}{2}\frac{h}{2}\exp\left(\frac{1}{4}\right)\frac{1}{2}\exp\left(-\frac{1}{4}\right) = \frac{1}{8}h^2$$

which is exactly the initial area. If, in a given advection direction k, the velocity v_k varies linearly with position x_k then Δt^* is constant from element to element since the 1D divergence term $\frac{\partial v_k}{\partial x_k}$ is constant. In this case application of the adjusted time step parameter ensures that the volume is globally conserved. However in a general flow Δt^* will vary from element to element and therefore care has to be taken not to overfill ($F > 1$) elements when adding volumes entering a cell from its neighbours. This is done by scaling the volume entering a cell with the ratio $\Delta t^*_{in}/\Delta t^*_{out}$ where subscripts *in* and *out* refer to the cell into which volume enters and cell from which volume leaves respectively. Therefore we can only expect the advection to be globally volume conserving in linear flow fields - and not in general. Later we will investigate the volume conserving properties of the method with some numerical experiments.

The advection procedure outlined above requires that the velocity is defined on the VOF cell faces and that the velocity is constant on each face. In practice these velocities are obtained by dividing the face fluxes with the face areas, i.e.

$$v_{j,i} = \frac{f_{j,i}}{A_{j,i}}$$

where $f_{j,i}$ is the flux on face i of VOF cell j and $A_{j,i}$ is the face area. In the next section it is explained how the fluxes are calculated. For now we assume that we know the face fluxes and the face areas. Since the interface advection is carried out in the local coordinate system it is necessary to make sure that the amount of fluid fluxed across a given interface in the local coordinate system agrees with flux in the global system. For a given face velocity $v_{j,i}$, the volume which is fluxed across face j, i during time step Δt relative to the total VOF cell volume is given by:

$$VF^{rel}_{j,i} = \frac{v_{j,i}\Delta t A_{j,i}}{V_j}$$

In order to obtain the same relative flux in the local coordinate system a geometry based factor $\beta_{j,i}$ is defined such that

$$\frac{v_{j,i}\Delta t A_{j,i}}{V_j} = \beta_{j,i}\frac{v_{j,i}\Delta t A_{loc}}{V_{loc}}$$

or

$$\beta_{j,i} = \frac{A_{j,i}V_{loc}}{V_j A_{loc}}$$

where A_{loc} and V_{loc} is the face area and cell volume in the local system. Thus if face velocities in the local system are scaled by $\beta_{j,i}$ the volumetric flux per cell volume will be the same in the local and global coordinate systems. In our case we use the standard element ($[-1, 1] \times [-1, 1] \times [-1, 1]$) in the local system and thus:

$$\beta_{j,i} = 2\frac{A_{j,i}}{V_j}$$

To summarize the procedure of calculating the fluxes the following procedure is used:

1. VOF cell face velocities are calculated from the face fluxes.
2. Based on the VOF cell volumes and face areas in the global system, $\beta_{j,i}$ is calculated.
3. Face velocities are transformed from the global to the local system using $\beta_{j,i}$.
4. Interface segments are advected in the local coordinate system using the split operator Lagrangian procedure.

It is noted that since the advection is carried out in the local coordinate system the adjusted time step Δt^* is based on the transformed velocities. Furthermore one has to take into consideration that adjacent cells do not necessarily have the same volume in the global system. This means that the volume VF_{in} entering a cell with total volume V_{in} and volume VF_{out} leaving a cell with total volume V_{out} must be scaled in the local system. Recalling from earlier the VF_{in} also needs to be scaled with the ratio $\Delta t^*_{in}/\Delta t^*_{out}$ in order to avoid overfilling cells. Therefore the volume entering a cell is given by:

$$VF_{in} = VF_{out}\frac{\Delta t^*_{in}V_{out}}{\Delta t^*_{out}V_{in}}$$

where VF_{out} is the volume leaving the adjacent cell due to the Lagrangian advection of the interface segment.

Since the advection procedure is purely lagrangian in nature there is in principle no restriction on the time step size used. Here we define a CFL number (Courant number) as:

$$\text{CFL} = \max\left(\frac{f_{j,i}}{V_j}\right)\Delta t, \quad j = 1, \dots, N_{vof}, \quad i = 1, \dots 6 \tag{2.4-26}$$

The CFL number in (2.4-26) is thus given by the maximum cell face flux relative to the cell volume times the time step size. Therefore if CFL > 1 then the advection routine has to be able to track an interface segment a distance greater than a cell width. This feature has not been implemented in our code and thus the time step should be restricted such that CFL ≤ 1. Furthermore since it is actually Δt^* which is used during the advection it is the CFL number based on Δt^* which should be less than one (CFL$^* < 1$). It is possible to set a criterion on the CFL number such that CFL$^* \leq 1$ always. This is done by noting that for compression (i.e. $\frac{\partial v_k}{\partial x_k} < 0$) $\Delta t^* < \Delta t$ and vice versa for expansion. Thus if we designate the face velocity on the face with maximum flux relative to VOF cell volume v_{max} then the maximum expansion possible in this VOF cell is obtained if the opposite cell face has velocity $-v_{max}$. Then given a cell width h the maximum 1D divergence term becomes $2v_{max}/h$. If we set $\Delta t v_{max}/h$ = CFL and $\Delta t^* v_{max}/h$ = CFL* = 1 then from equation 2.4-24 one obtains:

$$\frac{1}{2CFL}\left(\exp(2CFL) - 1\right) CFL = 1 \Longrightarrow CFL = \ln(3)/2$$

Thus if CFL $\leq \ln(3)/2$ then CFL$^* \leq 1$. It is noted that in practice only interface cells and their nearest neighbours need to be considered during the advection step (since CFL is chosen to be less than one).

2.4.2.4 Calculation of fluxes

In order to carry out the advection in a flow field generated by the Stokes solver the fluxes on the VOF cell faces need to be calculated. This requires that we introduce our 2-level procedure where we use a VOF mesh which is finer than the finite element mesh. The refined VOF mesh is obtained by subdividing each finite element into a number of smaller VOF cells. For this we specify the parameters N_1, N_2 and N_3 such that each finite element corresponds to a $N_1 \times N_2 \times N_3$ block of VOF cells. The subdivision itself is carried out in the local coordinate system such that for a given finite element its parent element is divided into smaller hexahedral elements with side lengths $h_\xi, h_\eta, h_\zeta = 2/N_1, 2/N_2, 2/N_3$. In Figure 2.10 (a) a 2D example is shown for the division of the parent finite element when $N_1 = 3$ and $N_2 = 2$.

The global coordinates of the VOF cells are readily found using the quadratic finite element shape functions since from its definition we know that each VOF cell is completely bounded by a finite element and we know the ξ, η, ζ coordinates of all VOF cells, cf. Figure

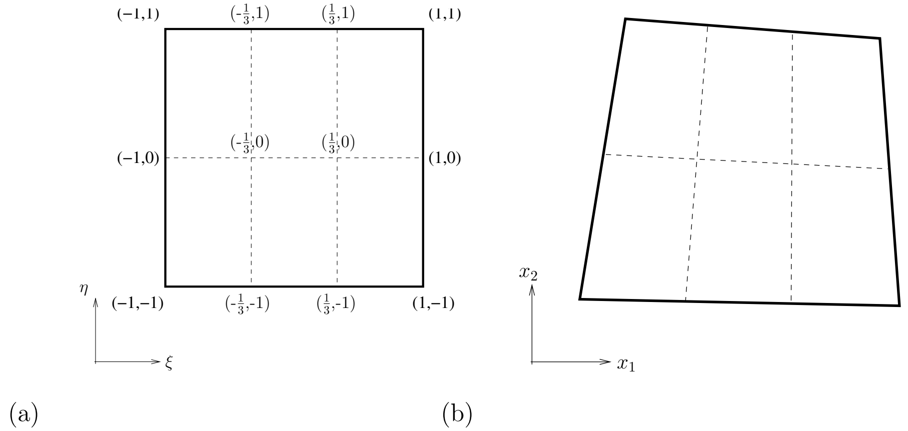


Figure 2.10: 2D sketch illustrating the subdivision of a finite element into VOF cells with $N_1 = 3$ and $N_2 = 2$. (a) Division of the finite element in local coordinates and (b) the VOF cells in the global coordinate system (sketch is not exact). The broken lines indicate VOF cell boundaries while the full lines indicate the finite element boundary.

2.10 (a) and (b). However, this requires that we know which finite element a given VOF cell belongs to and what local position this VOF cell has in the parent finite element. This is done by setting up a table which for each VOF cell points to the corresponding finite element and also specifies the local VOF cell number.

Now we return to the problem of calculating the fluxes on the VOF cell faces which is done by using:

$$f = \int_{\Gamma} \boldsymbol{n} \cdot \boldsymbol{v} dS \tag{2.4-27}$$

where Γ refers to the face boundary of the VOF cell. This boundary integral can be transformed to the local coordinate system by (Zienkiewicz & Taylor 2000):

$$f = \int_{\eta_0}^{\eta_1} \int_{\xi_0}^{\xi_1} \boldsymbol{v} \cdot \left\{ \begin{array}{c} \frac{\partial x_1}{\partial \xi} \\ \frac{\partial x_2}{\partial \xi} \\ \frac{\partial x_3}{\partial \xi} \end{array} \right\} \times \left\{ \begin{array}{c} \frac{\partial x_1}{\partial \eta} \\ \frac{\partial x_2}{\partial \eta} \\ \frac{\partial x_3}{\partial \eta} \end{array} \right\} d\xi d\eta \tag{2.4-28}$$

In this case the boundary under consideration has constant ζ. Similar expressions can be used for the case where either ξ or η is constant. The integral in equation 2.4-28 is calculated numerically using a Gaussian quadrature formula. This means that for each integration point the derivative of the global coordinates with respect to the local coordinates, i.e. terms of the type $\frac{\partial x_k}{\partial \xi}$ and $\frac{\partial x_k}{\partial \eta}$, and also the velocity components need to be evaluated. Since we know that the face boundary in question either is inside or on the boundary of the finite element the velocities can simply be interpolated from the finite element nodes onto the integration points using the quadratic shape functions. In order to evaluate the cross-product in equation 2.4-28, 2D quadratic shape functions N_i^{2D} are defined on the face in question such that e.g.

$$\frac{\partial x_k}{\partial \xi} = \sum_{i=1}^{8} \frac{\partial N_i^{2D}}{\partial \xi} x_{k,i} \quad , k = 1, \ldots, 3 \tag{2.4-29}$$

where $x_{k,i}$ are the global coordinates of the quadratic nodes on the face under consideration. These shape functions are the 2D equivalents of the 3D shape functions used for the Stokes solver. They are given e.g. in (Zienkiewicz & Taylor 2000). Since the subdivision is carried out in the local coordinate system the global coordinates $x_{k,i}$ of these quadratic nodes on the VOF cell faces can be calculated directly from the shape functions associated with the finite element mesh. It is noted that the integration weights, used in the numerical integration, are scaled linearly to match the integration boundaries in equation 2.4-28, i.e. the integration boundary is not from -1 to 1 in the general case. The sign of the flux calculated according to 2.4-28 depends on the node numbering direction used on the element faces. A thorough explanation of this problem is given in e.g. (Henriksen 1991) and will not be discussed further here. However, it is noted that the node numbering is chosen such that positive fluxes indicate volume leaving a cell and vice versa.

2.4.2.5 Flux adjustment

Due to the application of constant discontinuous pressures in the finite element formulation each finite element is divergence free. Thus the fluxes on the boundary of each finite element sum to zero. However, this does not mean that each finite element is point-wise divergence free and thus the flux calculations on the faces of the VOF cells will not necessarily result in volume conservation within each VOF cell (except if $N_1 = N_2 = N_3 = 1$). In order not to loose or accumulate volume of the disperse phase it is therefore necessary to introduce an adjustment to the calculated fluxes such that the VOF cells become divergence free. The correction method implemented here is to some extent based on the method described in (Li & Li 2004). If we for simplicity take an example with a 2D finite element subdivided into 6 VOF cells as shown in Figure 2.11, the procedure is as follows: For a given finite element the fluid fluxes are first calculated on the faces of all the VOF cells using the procedure described in the previous section. With reference to Figure 2.11 we designate these fluxes as $f_1, \ldots, f_{17}$. The sign of these fluxes depends on which VOF cell is considered. Since the finite element is divergence free we know that $\sum_{i=1}^{10} f_i = 0$ and use this information to adjust the internal fluxes $f_{11}, \ldots, f_{17}$ such that all cells become divergence free. Let us start by adjusting f_{12} and f_{13} which is done by first calculating the residual flux

$$\delta f = \sum_{i=1}^{4} f_i + f_{12} + f_{13}$$

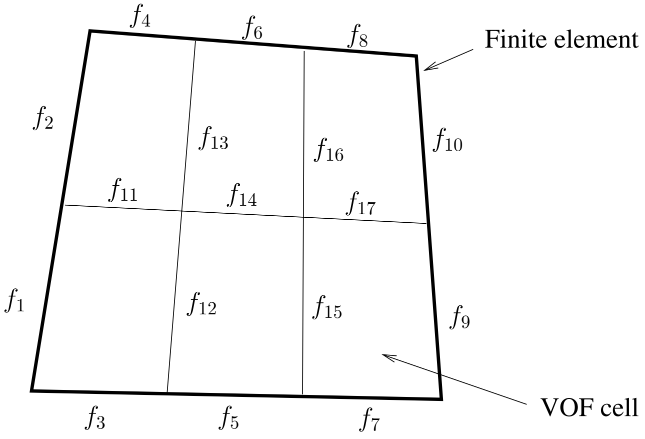


Figure 2.11: Schematic drawing to illustrate the flux adjustment method.

which is then subtracted from f_{12} and f_{13} such that the change in flux pr. area is the same on face 12 and 13, i.e.

$$f_{12}^* = f_{12} - \frac{\delta f}{A_{12} + A_{13}} A_{12}$$

and

$$f_{13}^* = f_{13} - \frac{\delta f}{A_{12} + A_{13}} A_{13}$$

where A_{12} and A_{13} are the areas (lengths in 2D) of face 12 and 13 respectively. It is then straight forward to calculate flux f_{11}. Using f_{12}^* and f_{13}^* the same procedure can be used to adjust the remaining fluxes. It is noted that this method is direction dependent such that the final result will depend on if one starts by adjusting fluxes in the horizontal or in the vertical direction. The procedure can be extended to the 3D case directly, however, in this case there are 6 different combinations of fluxing directions which lead to different final results. In order to reduce any skewness in the final adjusted flux field the adjusted fluxes are calculated using all 6 combinations where after the average is used in further calculations. One of the problems with such an adjustment is that the flow field is altered in a way that is not governed by the equations of motions, thus it is to some degree physically incorrect. In section 3.5 an alternative method for calculating the fluxes on the VOF cell faces is discussed.

2.4.3 Tests of the advection algorithm

In this section the advection procedure is investigated in terms of accuracy with respect to spatial convergence. Furthermore the volume conservation behavior is also investigated in various kinds of flow fields and the volume error is investigated in terms of temporal convergence. It is noted that all advection tests are carried out on a VOF mesh which coincides with the finite element mesh (i.e. 1-level) and that the adjusted time step parameter Δt^* is used unless otherwise stated. The implications of using the 2-level method will be investigated later when interfacial tension is present and the flow is calculated using the finite element Stokes solver. Here we will investigate the advection algorithm in flow fields which are pre-defined on finite element nodes. Fluxes on VOF cell faces are then calculated as explained earlier. The first test problem we will use is a 3D variant of the Zalesak's slotted disk rotation originally introduced in (Zalesak 1979). Using this problem the behavior of the error is analyzed in terms of mesh refinement. Hereafter the volume conserving behavior of the advection scheme is investigated in some simple flows and also in a highly non-linear and time dependent flow field. All tests have been carried out in a mesh with regular hexahedral elements and thus the performance of the reconstruction and advection algorithms has not been tested in a domain with irregular VOF cells. However, if the code is to be used in the future for simulations in irregular domains such tests should be performed.

2.4.3.1 Zalesak's slotted disk rotation

This problem was designed to test the error associated with VOF type advection algorithms in 2D domains. The test consists of translating and rotating a disc with a slot cut into it, c.f. Figure 2.12.

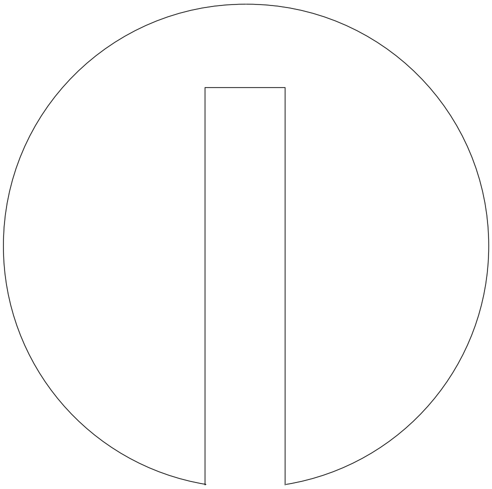


Figure 2.12: Zalesak's slotted disc.

The width of the slot is 1/3 of the disk radius and the distance from the slot end to the disk face is the same as the width. The disk is then rotated and translated in a flow

	mesh 1	mesh 2	mesh 3	mesh 4
Mesh size	$16 \times 16 \times 8$	$32 \times 32 \times 16$	$64 \times 64 \times 32$	$128 \times 128 \times 64$

Table 2.1: Meshes used in the slotted sphere tests. The 1-level method is used (i.e. the FE and VOF mesh are the same).

defined by

$$\begin{aligned} v_1 &= -\dot{\omega} x_2 + \dot{\omega} x_{2,0} \\ v_2 &= \dot{\omega} x_1 - \dot{\omega} x_{1,0} \end{aligned}$$

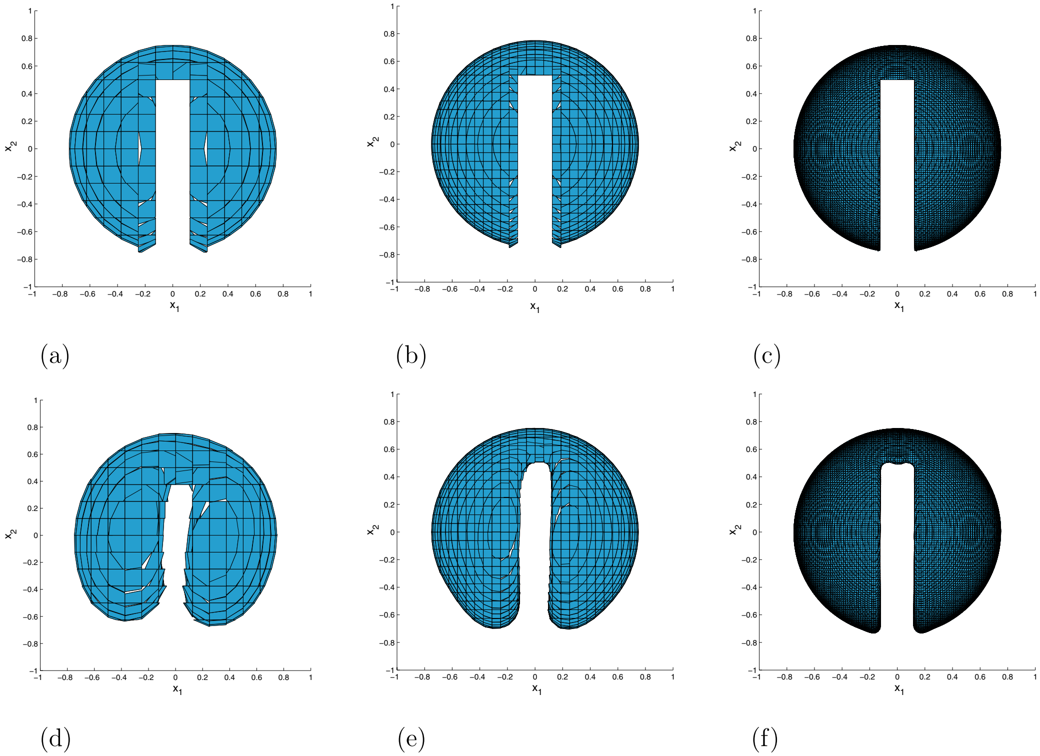
where $\dot{\omega}$ is the angular frequency. In (Zalesak 1979) the centre of the disc is $(x_{1,0}, x_{2,0}) = (0.5, 0.75)$ and the radius is 0.15. Since $\frac{\partial v_1}{\partial x_1} = 0$ and $\frac{\partial v_2}{\partial x_2} = 0$ for this flow field the volume will be conserved exactly during advection. This test it therefore not suited for investigating the volume conserving behavior of the method which will be investigated later using other types of flow fields. The slotted disc problem has been used for testing a number of advection procedures, e.g (Harvie & Fletcher 2000, Harvie & Fletcher 2001, Scardovelli & Zaleski 2003, Pilliod & Puckett 2004). Although the test has been designed for 2D problems we will use it here for our 3D advection procedure. In order to do this a sphere with radius r_0 is used in stead of a disk. A slot with width $1/3r_0$ is then cut all the way through the sphere in the x_3 direction. For simplicity we will place the slotted sphere in the centre of the domain such that only rotation takes place. We will later investigate the advection performance in a complex flow which includes both rotation and translation. It is noted that the geometry of the slotted sphere includes discontinuities along the points where the slot cuts the surface of the sphere and in the groove of the slot. In the PLIC reconstruction these discontinuities are smoothed as a consequence of the smoothing effect associated with the interface normal calculation. The error associated with the rotation is investigated in terms of mesh refinement. This problem is symmetric across the (x_1, x_2)-plane at $x_3 = 0$ and thus only half the domain needs to be considered. The meshes used in this test are shown in Table 2.1.

The time step in all three experiments is given by:

$$\Delta t = \frac{2\pi}{377}$$

while the angular frequency is $\dot{\omega} = 1\text{s}^{-1}$. Hence it takes 377 time steps to complete one rotation of the slotted sphere. The computational domain is given by the box $([-1, 1] \times [-1, 1] \times [-1, 1])$ and the radius of the sphere is $r_0 = 0.75$. Furthermore, the x_3 direction is kept neutral, thus $v_3 = 0$.

In Figure 2.13 (a)-(c) the slotted sphere on mesh 1,2 and 4 before rotation is plotted using the PLIC reconstruction. Here one sees that going from (a) to (c) the accuracy in the description of the discontinuity where the slot cuts the sphere surface increases. It is noted that although the groove of the slot seems to be perfectly described, this is not the case, it is merely a consequence of the way the spheres are plotted. In Figure 2.13 (d) - (f) the



(a) (b) (c)

(d) (e) (f)

Figure 2.13: The PLIC reconstruction of the slotted sphere before and after one rotation. The view is perpendicular to the (x_1, x_2)-plane. (a), (d) mesh 1 before and after rotation; (b), (e) mesh 2 before and after rotation; (c), (f) mesh 4 before and after rotation.

Mesh	Error	Order
1	0.163	
		1.54
2	0.0561	
		1.28
3	0.0231	
		1.07
4	0.0110	

Table 2.2: Error in F after one rotation of the slotted sphere. Also shown is the order of the error decrease going from one mesh to the next.

slotted spheres are shown after one rotation. In all cases it is seen that the discontinuities at the slot edges are smoothed and that the vertical walls of the slot become distorted - especially in Figure 2.13 (d). This deformation behavior of the slot is typical for this type of test and is described e.g. in (Scardovelli & Zaleski 2003, Pilliod & Puckett 2004). The error associated with the rotation is defined as

$$\text{Error} = \frac{\sum_{j=1}^{N_{vof}} F_j - \tilde{F}_j}{\sum_{j=1}^{N_{vof}} F_j}$$

where N_{vof} is the number of VOF cells, F_j is the value of F in cell j before advection and $\tilde{F}_j$ is the value after advection. In table 2.2 the error obtained after one rotation of the slotted sphere can be seen. Also shown is the order at which the error decreases with h. From these results it is seen that the error decreases as the mesh is refined but that the order also decreases. It is not possible to make any conclusions on the order of the advection procedure from the results in Table 2.2. However, it is noted that error is a combination of the error introduced by the PLIC reconstruction and that of the advection procedure. Since the normal estimation is carried out using a method which for a completely regular mesh is comparable to Youngs method, which is first order accurate, we cannot expect the advection to be much better than first order. Therefore in order to investigate the true convergence order of the advection scheme a second order reconstruction method like LVIRA or ELVIRA, see e.g. (Pilliod & Puckett 2004), should be implemented. However this has not been done at the present time because these methods are not directly extensible to non-regular meshes.

2.4.3.2 Advection in pre-defined simple flow fields

Here a sphere is deformed in either simple shear, planar elongation or extensional flow, i.e.:

Simple shear

$$v_1 = x_2; \quad v_2 = 0; \quad v_3 = 0$$

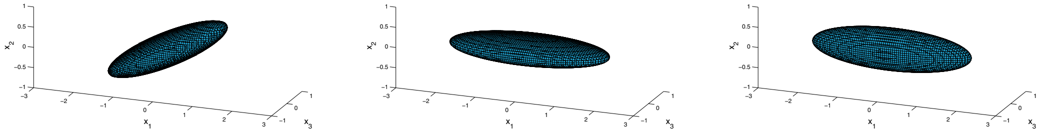
Planar elongation

$$v_1 = x_1; \quad v_2 = -x_2; \quad v_3 = 0$$

Uni-axial extension

$$v_1 = x_1; \quad v_2 = -\frac{1}{2}x_2; \quad v_3 = -\frac{1}{2}x_3$$

The purpose of these tests is to check how well volume is conserved and if the interface is kept sharp. The simulations are carried out using a regular 120 × 40 × 20 mesh where symmetry at $x_3 = 0$ is applied. The domain size is $[-3, 3] \times [-1, 1] \times [-1, 0]$ and the time step size is chosen such that CFL = 1/4. In Figure 2.14 (a)-(c) the deformation from the initial sphere with radius $r_0 = 0.8$ can be seen for each of the three flow fields.



(a) (b) (c)

Figure 2.14: Advection in simple flow fields. (a) simple shear (b) planar elongation and (c) pure extension. Deformation after 101 time steps in all cases.

Although this is not a quantitative test it shows that the interface is kept sharp throughout the advection procedure. In all test cases the relative change in volume of the advected phase after 101 time steps is on the order of 10^{-8} %. Here a tolerance of 10^{-8} is used in the PLIC reconstruction (bisection iteration). If the tolerance is changed to 10^{-12} the relative change in volume is in the order of 10^{-12} % and thus we can conclude that the volume is conserved within the tolerances used in the interface reconstruction for the 3 flow fields tested. It is noted that if the time step is not adjusted according to equation 2.4-24 (i.e. Δt^*) then the relative change in the volume of the advected phase is in the order of 10^{-2} % for planar elongation. Tests have also been performed for various flows which are linear in position, i.e. flows of the type:

$$v_k = a_k x_1 + b_k x_2 + c_k x_3, \quad k = 1, \ldots, 3$$

where $a_1 + a_2 + a_3 = 0$. For brevity the results will not be shown here, however, in all cases the volume was conserved within the tolerances used in the PLIC reconstruction indicating the volume is exactly conserved in linear flow fields.

	mesh 1	mesh 2	mesh 3	mesh 4
Number of Cells	20 × 20 × 6	40 × 40 × 9	80 × 80 × 15	160 × 160 × 27

Table 2.3: Meshes used for the single vortex tests. The 1-level method is used.

2.4.3.3 Advection in the reversed single vortex flow

In order to investigate the performance of the advection algorithm more rigorously, tests have been carried out in a 2D non-uniform vorticity flow given by:

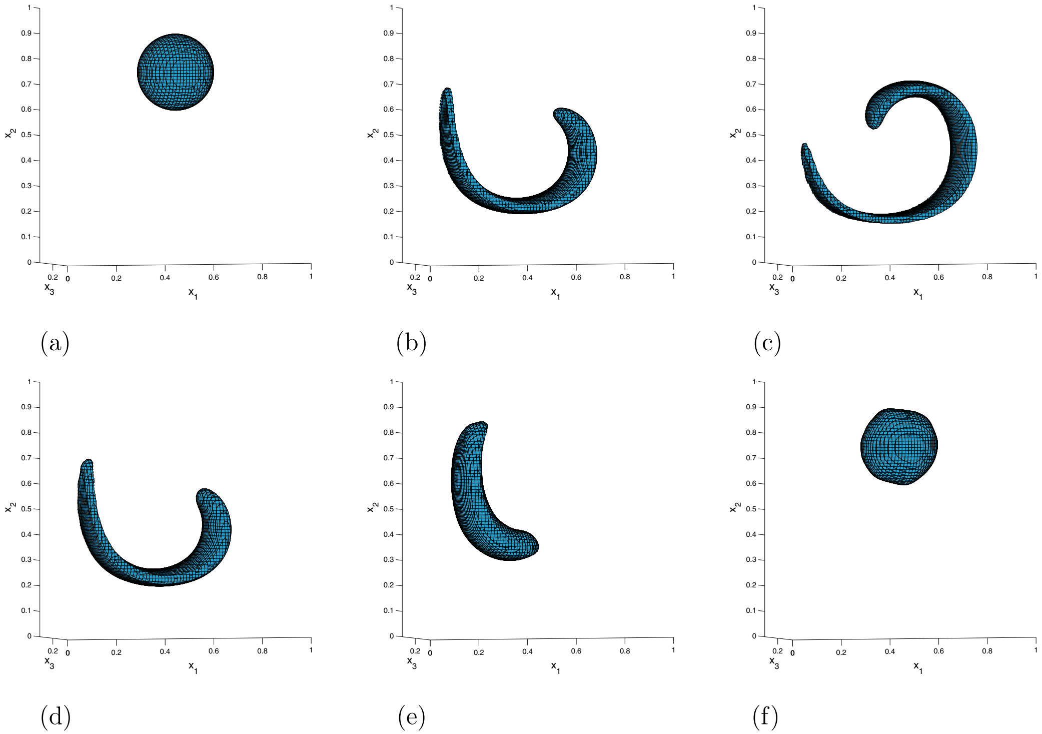
$$v_1 = 2\sin(\pi x_2)\cos(\pi x_2)\sin^2(\pi x_1)\cos\left(\frac{\pi t}{T}\right)$$
$$v_2 = 2\sin(\pi x_1)\cos(\pi x_1)\sin^2(\pi x_2)\cos\left(\frac{\pi t}{T}\right)$$

This test flow was originally used by Rider & Kothe (1998) to investigate the performance of 2D advection algorithms and has later been used by e.g. (Harvie & Fletcher 2000, Scardovelli & Zaleski 2003, Pilliod & Puckett 2004) also to test 2D advection algorithms. The 2D domain is given by the unit square $[0,1] \times [0,1]$ in which the flow is characterized by a single vortex which is rotating and stretching about the centre $(x_1, x_2) = (0.5, 0.5)$ of the domain. In the 2D case a circle is placed at $(x_{1,0}, x_{2,0}) = (0.5, 0.75)$ such that the centre will translate with the flow. The flow is time dependent and changes direction when the time is half the amplitude T, i.e. $t = T/2$. Here the deformation is also at its maximum. At $t = T$ the circle is (should be) back at its original position. In our case we want to test the algorithm in 3D. In order to do this the same flow field is used while the x_3 component of the velocity is zero. Here a computational box with size $[0,1] \times [0,1]$ in the x_1 and x_2 direction is used while the size in the x_3 direction is specified such that the VOF cells have equal side lengths. Furthermore since x_3 is neutral the problem is symmtric at $x_3 = 0.5$. The maximum deformation increases with increasing amplitude T. The initial position if the sphere is the same as that used in (Rider & Kothe 1998) (see also above). First the error is examined with respect to mesh refinement using a constant Δt. For this purpose we use the meshes given in Table 2.3.

We use an amplitude of $T = 4$ in our tests. In Figure 2.15 the reconstructed PLIC segments can be seen at various times during the advection stage for mesh 3 in Table 2.3. Going from (a)-(c) in Figure 2.15 the initial sphere is advected from its initial shape to its maximum deformation in (c). Figure 2.15 (d)-(f) shows the deformation after the flow is reversed which eventually leads to the initial spherical shape (approximately). The spatial error is calculated at $t = T$ and is in this case defined as (in agreement with Rider & Kothe (1998)):

$$\text{Error} = \sum_{j=1}^{N_{vof}} h^3 |F_j - \tilde{F}_j|$$

where h^3 is the VOF cell volume.



(a) (b) (c)

(d) (e) (f)

Figure 2.15: Advection of an initially spherical body in the single reversed vortex flow field. (a) $t = 0$, (b) $t = 1/4T$, (c) $t = 1/2T$, (d) $t = 3/4T$, (e) $t = 7/8T$ and (f) $t = T$. Mesh 3 in Table 2.3 was used in the simulation and the sphere was initially centered at $(x_1, x_2, x_3) = (0.5, 0.75, 0.375)$. The amplitude was $T = 4$.

Mesh	Error	Order	$\mathrm{V_{err}}\%$
1	$4.59\mathrm{e}-3$		$3.26\mathrm{e}-2$
		2.20	
2	$1.00\mathrm{e}-3$		$3.30\mathrm{e}-2$
		1.97	
3	$2.55\mathrm{e}-4$		$3.28\mathrm{e}-2$
		1.64	
4	$8.19\mathrm{e}-5$		$3.28\mathrm{e}-2$

Table 2.4: Error in F and average relative error in volume $\mathrm{V_{err}}$ from advection of a sphere in the reversed single vortex flow with $T = 4$. The error is calculated using the initial F-field and the final F-field at $t = T$.

The time step, which is kept constant for all simulations, is chosen to be

$$\Delta t = \frac{1}{320}$$

This time step corresponds to CFL$\approx 1/2$ at $t = 0$ on mesh 4. In Table 2.4 the error in the F-field at $t = T$ is shown for the various meshes. Furthermore the relative error (or change) in volume of the advected phase is also shown. Here the average (of absolute values) is taken over all time steps. From Table 2.4 we see that the error in F decreases with mesh refinement while the relative error in the total volume is more or less constant. The order in which the error in F decreases is, as was also the case with the slotted disc experiments, decreasing with increased mesh refinement. A similar convergence behavior is described by Rider & Kothe (1998) in their tests on 2D advection algorithms. Unfortunately it was not possible to carry out the advection test on a finer mesh than mesh 4 in Table 2.3 within a reasonable computational time. Therefore it was not possible to investigate if the order of the error stabilized as the mesh is refined.

In order to investigate the effect of the time step size on the performance of the advection algorithm, simulations are carried out using mesh 2 with various CFL numbers. The results can be seen in Table 2.5. The basis time step is $\Delta t_0 = 1/73$ which corresponds approximately to CFL=0.5 based on the flow field at $t = 0$. From Table 2.5 it is seen that the error in the F-distribution increases with decreasing time step size. This behavior is also observed by Scardovelli & Zaleski (2003) using various types of 2D advection algorithms and is explained by the increasing number of reconstructions needed as the time step size is decreased. The average error in the total volume of the advected fluid is seen to increase when going from $\Delta t = 1/2\Delta t_0$ to $\Delta t = 1/4\Delta t_0$. For smaller time steps the volume error decreases with decreasing step size. However, the decrease in the volumetric error with time step size seems to be somewhat oscillating with large decreases at certain time step sizes, e.g. $1/8\Delta t_0$ to $1/16\Delta t_0$. Nevertheless the volumetric error is quite small and does decrease with step size.

In (Scardovelli & Zaleski 2003) it is indicated that one problem with purely Lagrangian advection is that the method leads to a phase error. This means that for the reversed single

Time step	Error in F	$\mathrm{V}_{\mathrm{err}}\%$
Δt_0	$7.98\mathrm{e}-4$	$7.69\mathrm{e}-2$
$1/2\Delta t_0$	$9.27\mathrm{e}-4$	$9.43\mathrm{e}-2$
$1/4\Delta t_0$	$1.00\mathrm{e}-3$	$2.35\mathrm{e}-2$
$1/8\Delta t_0$	$1.02\mathrm{e}-3$	$2.23\mathrm{e}-2$
$1/16\Delta t_0$	$1.03\mathrm{e}-3$	$5.84\mathrm{e}-3$
$1/32\Delta t_0$	$1.04\mathrm{e}-3$	$8.37\mathrm{e}-4$
$1/64\Delta t_0$	$1.04\mathrm{e}-3$	$5.24\mathrm{e}-4$
$1/128\Delta t_0$	$1.05\mathrm{e}-3$	$2.05\mathrm{e}-4$

Table 2.5: Error in F and average relative error in volume $\mathrm{V}_{\mathrm{err}}$ on mesh 2. $\mathrm{V}_{\mathrm{err}}$ is based on the average (of absolute values) of all time steps. The basis time step is $\Delta t_0 = 1/73$.

vertex flow the final position of the circle (2D) is out of phase with the initial position. The reason for this is that the procedure is explicit in all advection directions. However, in the present implementation the explicit nature of the method is adjusted through the time step parameter Δt^*, cf. equation 2.4-24. It is therefore interesting to compare the results obtained using the Lagrangian advection procedure with and without the adjusted time step parameter Δt^*. Therefore the same simulations as those used for studying the influence of the time step size, cf. Table 2.5, are carried out again but without using Δt^*. First we take a look at the qualitative differences obtained when using the two methods. In Figure 2.16 (a) a slice through the symmetry plane of the sphere after one period of advection (i.e. $t = T$) can be seen for advection with no time step adjustment (broken curve) and with time step adjustment (dotted curve). Also shown is the initial position of the sphere (solid circle). From this figure it is clear that the simulation carried out with no time step adjustment seems to be out of phase with the initial position. This problem is not observed when Δt^* is applied. In Figure 2.16 (b) the results from the two methods are shown when the time step size is $\Delta t = 1/8\Delta t_0$. Here it is seen that the simulation carried out without the adjusted time step parameter is no longer out of phase with the original position of the sphere. Furthermore it is seen that the results from the two simulations lie much closer than in Figure 2.16 (a). This indicates that the results obtained from the two methods approach each other as the time step Δt decreases. In Table 2.6 the error in the final F-distribution and the error in volume can be seen for the simulations carried out without the adjusted time step parameter. By comparing the results in Table 2.6 with those in Table 2.5 it is seen that for $\Delta t \geq 1/4\Delta t_0$ the error in the F-distribution is larger when no adjustment is made to Δt. For smaller time steps the error in the final F-distribution is more or less the same for the two methods. However, if we compare the volume error associated with the two methods it is clear that when Δt^* is applied volume conservation is much more rigorous (approximately 2 orders of magnitude).

It is noted that it is not expected that the final F distribution will converge towards the initial F distribution for the numerical experiments shown in Table 2.5 and 2.6 because the error also depends on the mesh density.

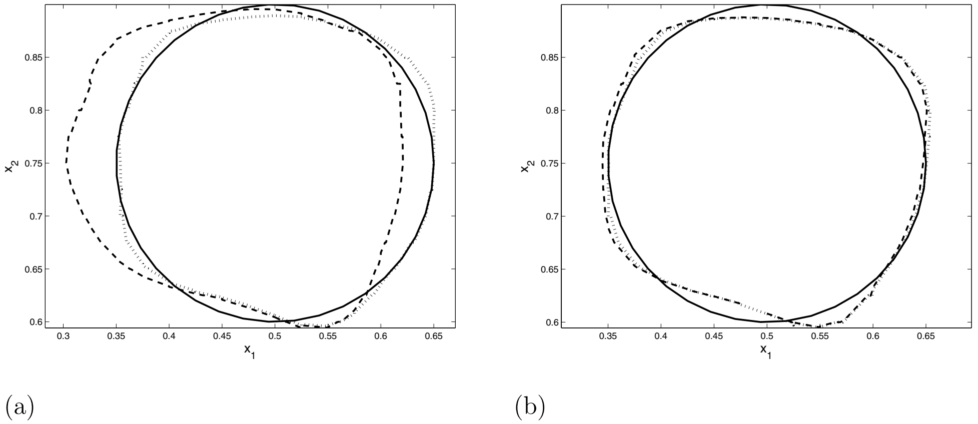


Figure 2.16: (x_1, x_2)-slice at symmetry plane after advection in the reversed single vortex flow at $t = T$ where $T = 4$. The solid curve is the original position of the sphere. The dotted curve is the position after advection using the lagrangian advection scheme *with* the adjusted time step parameter and the dashed curve is the position after advection using lagrangian advection *without* the adjusted parameter Δt^*. (a) $\Delta t = \Delta t_0$, (b) $\Delta t = 1/8\Delta t_0$. In the simulations $\Delta t_0 = 1/73$ was used.

Time step	Error in F	$\mathrm{V_{err}}\%$
Δt_0	2.03e − 3	3.66
$1/2\Delta t_0$	1.22e − 3	1.57
$1/4\Delta t_0$	1.06e − 3	8.26e − 1
$1/8\Delta t_0$	1.03e − 3	4.71e − 1
$1/16\Delta t_0$	1.02e − 3	2.17e − 1
$1/32\Delta t_0$	1.03e − 3	1.12e − 1
$1/64\Delta t_0$	1.04e − 3	5.54e − 2
$1/128\Delta t_0$	1.04e − 3	2.77e − 2

Table 2.6: Error in F after 1 period of advection in the reversed single vortex flow without time step adjustment. Also shown is the average error in volume of the advected fluid when simulations are carried out with Δt. $\Delta t_0 = 1/73$ and $T = 4$ was used in the simulations.

2.4.4 Summary of the interface reconstruction and advection routines

In this work a 3D purely Lagrangian split operator advection scheme has been implemented. The interface is reconstructed using the Piecewise Linear Interface Calculation (PLIC) method. In order to increase accuracy and especially volume conservation behavior during advection the explicit Lagrangian method described in e.g. (Gueyffier et al. 1999) for the 3D case has been modified. This is done by letting the velocity of a point within a VOF cell vary with time. This results in a adjusted time step Δt^* which includes the original time step Δt and the 1D velocity divergence term of the cell under consideration. Therefore the adjusted time step varies from VOF cell to VOF cell. This also requires some extra care not to overfill cells by scaling volumes entering a cell with the ratio of the adjusted time step of the cell which accepts the volume and the cell which donates the volume. In the implementation all calculations are carried out in the local curvilinear coordinate system wherefore fluxes are scaled appropriately before advection is carried out. Furthermore the interface normal is transformed from the global to local coordinates before advection is carried out. A number of advection tests have been carried out on VOF cell meshes with regular hexahedral elements. These tests include rotation of a slotted sphere, advection of a sphere in various linear flow fields and advection of a sphere in the reversed single vortex flow proposed by Rider & Kothe (1998). The tests indicate that the combined interface reconstruction (PLIC) and Lagrangian advection schemes are between first and second order accurate with mesh size h. Comparison between results obtained for advection with and without the adjusted time step parameter Δt^* show that considerably more accurate results are obtained at large to medium time steps when Δt^* is used in place of Δt. Furthermore, volume conservation is much more rigorous for all time step sizes when the adjusted time step parameter is applied. In practice this means that larger time steps can be used in our algorithm compared to the original Lagrangian procedure while still obtaining the same accuracy in both interface position and volume conservation.

Since all advection tests are carried out using a mesh based on regular hexahedral elements the performance of the implemented advection routine in non-regular cell geometries needs to be tested if calculations are to be performed in complex geometries. However, the simulations on drop deformation and break-up behavior carried out in this project are all based on rectangular geometries wherefore advection tests in non-regular domains have not been carried out.

This concludes the section on interface reconstruction and volume advection in predefined flow fields. In the forthcoming sections we will concentrate on the inclusion of interfacial tension.

2.5 Interfacial tension

One implication of the VOF method is that the interface is not sharply defined in the domain and typically has a finite thickness of 1-2 VOF cells. The inclusion of interfacial tension forces therefore requires some special treatment where interfacial tension is not applied onto interfaces but rather as a volumetric force in interface elements (and possibly their neighbours). Two models developed for VOF methods are available. These are the Continuous Surface Force (CSF) model of Brackbill et al. (1992) and the Continuous Surface Stress (CSS) model of (Lafaurie et al. 1994). Both models have been implemented, however, most simulations have been carried out using the CSS model as this only requires calculation of first order derivatives of the F-field while the CSF model requires the calculation of second order derivatives. Therefore we will focus on the CSS model here while details on the CSF model are presented in Appendix B. In the following subsections it is shown how the CSS model is implemented. Hereafter results from simulations with a static drop will be given both for the CSS and CSF model. These simulations are carried out as a preliminary check in order to verify that correct pressure distributions are obtained. Also the magnitude of the spurious (or anomalous) currents which are always present in VOF calculations with interfacial tension will be investigated. The phenomena is seen as an artificial flow field with circulation zones in the vicinity of the surface of a sphere even though no external flow field is applied. The reason for this is related to errors in the approximation of the gradient of the discontinuous function F. Methods are available to reduce these artifacts which will be discussed later. It is though noted that no work has been carried out to overcome or reduce this problem in the present implementation which thus is a possible subject for future work.

2.5.1 The CSS interfacial tension model

If a fluid system consists of two or more immiscible fluid phases a pressure jump will exist across the interfaces separating the various fluids if interfacial tension is present. Let us concentrate on two-fluid systems and denominate the two fluids as phase 1 and phase 2. Then the momentum balance in its finite element formulation given by equation 2.2-26 can be written as the sum:

$$\sum_{i=1}^{2}\left(-\int\limits_{\Gamma_i}\phi^v[\boldsymbol{n}_i\cdot\boldsymbol{\pi}_i]dS+\int\limits_{\Omega_i}[\nabla\phi^v\cdot\boldsymbol{\pi}_i]dV+\int\limits_{\Omega_i}\phi^v\boldsymbol{f}_i dV\right)=\mathbf{0} \tag{2.5-30}$$

The volume integrals in equation 2.5-30 have been discussed earlier and need not be reconsidered here. If we concentrate on the surface integrals instead and note that $\boldsymbol{n}_2 = -\boldsymbol{n}_1$ everywhere on $\Gamma_1 = \Gamma_2 = \Gamma_{12}$ the surface terms can be written as:

$$-\int\limits_{\Gamma_1}\phi^v[\boldsymbol{n}_1\cdot\boldsymbol{\pi}_1]dS-\int\limits_{\Gamma_2}\phi^v[\boldsymbol{n}_2\cdot\boldsymbol{\pi}_2]dS=-\int\limits_{\Gamma_{12}}\phi^v[\boldsymbol{n}_1\cdot(\boldsymbol{\pi}_1-\boldsymbol{\pi}_2)]dS \tag{2.5-31}$$

In equation 2.5-31 the integrand $[\boldsymbol{n}_1 \cdot (\boldsymbol{\pi}_1 - \boldsymbol{\pi}_2)]$ is the difference or jump in the total molecular stress tensor across the interface and thus is directly related to the interfacial tension. In case of constant interfacial tension coefficient σ the stress jump can be written as:

$$[\boldsymbol{n}_1 \cdot (\boldsymbol{\pi}_1 - \boldsymbol{\pi}_2)] = [\kappa\sigma\boldsymbol{n}_1] \tag{2.5-32}$$

where κ is the curvature of the interface. It is noted that if the interfacial tension coefficient is position dependent the gradient of σ needs to be added to the RHS of equation 2.5-32, see e.g. (Szabo 1994). By using equation 2.5-32 the surface integral in equation 2.5-31 is written as:

$$-\int\limits_{\Gamma_{12}} \phi^v[\boldsymbol{n}_1 \cdot (\boldsymbol{\pi}_1 - \boldsymbol{\pi}_2)]dS = -\int\limits_{\Gamma} \phi^v[\kappa\sigma\hat{\boldsymbol{n}}]dS \tag{2.5-33}$$

where for simplicity indices 12 and 1 have been omitted on the RHS and $\hat{\boldsymbol{n}}$ is the unit normal pointing from phase 1 to phase 2. In the discrete case the integral on the RHS of equation 2.5-33 can be readily calculated if the interface position is defined on finite elements faces, e.g. as in Lagrangian finite element methods. However, as mentioned earlier this is not the case when the interface is tracked by the VOF method. Here the interface position is only known implicitly from the F-distribution and even if the interface is reconstructed e.g. by PLIC the interface segments do not coincide with the faces of the finite element mesh. It is therefore necessary to make some approximations to the integral on the RHS of equation 2.5-33. First it is noted that the RHS of equation 2.5-33 can be written as a volume integral by applying a delta function δ_s such that:

$$-\int\limits_{\Gamma} \phi^v[\kappa\sigma\hat{\boldsymbol{n}}]dS = -\int\limits_{\Omega} \phi^v[\kappa\sigma\hat{\boldsymbol{n}}]\delta_s(\hat{\boldsymbol{n}}(\boldsymbol{x} - \boldsymbol{x}_s))dV \tag{2.5-34}$$

Here $\boldsymbol{x}_s$ are interface coordinates. We now denote the term $[\kappa\sigma\hat{\boldsymbol{n}}]\delta_s$ on the RHS of (2.5-34) as the interfacial tension force per unit volume $\boldsymbol{F}_s$. According to the CSS interfacial tension model, $\boldsymbol{F}_s$ can be written as the divergence of a capillary pressure tensor $\boldsymbol{T}$ (Lafaurie et al. 1994):

$$\boldsymbol{F}_s = [\kappa\sigma\hat{\boldsymbol{n}}]\delta_s = -\nabla\boldsymbol{T} \tag{2.5-35}$$

where $\boldsymbol{T}$ is given by:

$$\boldsymbol{T} = -\sigma(\boldsymbol{I} - \hat{\boldsymbol{n}} \otimes \hat{\boldsymbol{n}})\delta_s \tag{2.5-36}$$

Here $\boldsymbol{I}$ is the identity tensor and δ_s is the previously defined delta function. The problem now consists of calculating the unit interface normals $\hat{\boldsymbol{n}}$ and the function δ_s. In order for the discrete interfacial tension calculations to be consistent with the continuum case in (2.5-34) any discrete evaluation of the interfacial tension term should obey:

$$\lim_{h\to 0} \int\limits_{\Omega} [\kappa\sigma\boldsymbol{g}(\boldsymbol{x}, h)]dV = \int\limits_{\Gamma} [\kappa\sigma\hat{\boldsymbol{n}}]dV \tag{2.5-37}$$

where $\boldsymbol{g}(\boldsymbol{x})$ is the discrete evaluation of $[\hat{\boldsymbol{n}}\delta_s]$ and h is the thickness of the interface which is on the order of the mesh size used in the discretization. The limiting property in (2.5-37) is usually achieved by smoothing the F-field by a smoothing kernel prior to the evaluation of the interface normals and defining (Brackbill et al. 1992, Lafaurie et al. 1994):

$$\delta_s = \left|\nabla\tilde{F}\right|$$

where $\tilde{F}$ is the smoothed F-field. The discretely calculated interface unit normal $\hat{\boldsymbol{n}}_s$ is then found as:

$$\hat{\boldsymbol{n}}_s = \frac{\nabla\tilde{F}}{\left|\nabla\tilde{F}\right|}$$

However, as discussed by Brackbill et al. (1992) in their CSF method, the force arising from the interfacial tension calculations should be localized to the proximity of the interface and thus using the non-smoothed F-field for approximating δ_s should be an advantage. This is also supported by Lafaurie et al. (1994) who find that smoothing the F-field does not lead to appreciable better results compared to the non-smoothed case. In our implementation we do not smooth F in connection with the interfacial tension calculations and thus the unit interface normal and the delta function are approximated as:

$$\hat{\boldsymbol{n}}_s = \frac{\nabla F}{|\nabla F|}, \quad \delta_s = |\nabla F|$$

In order to incorporate the CSS model into the flow solver, equation 2.5-35 is inserted into equation 2.5-34 which combined with equation 2.5-33 leads to:

$$-\int\limits_{\Gamma_{12}} \phi^v[\boldsymbol{n}_1 \cdot (\boldsymbol{\pi}_1 - \boldsymbol{\pi}_2)]dS = \int\limits_{\Omega} \phi^v \nabla \cdot \boldsymbol{T} dV \tag{2.5-38}$$

Applying partial integration and the Gauss-Ostrogradskii divergence theorem on the RHS of equation 2.5-38 leads to:

$$\int\limits_{\Omega} \phi^v \nabla \cdot \boldsymbol{T} dV = \int\limits_{\Gamma_{12}} \phi^v[\boldsymbol{n}_1 \cdot (\boldsymbol{T}_1 - \boldsymbol{T}_2)]dS - \int\limits_{\Omega} \nabla\phi^v \boldsymbol{T} dV \tag{2.5-39}$$

Here indices 1 and 2 refer to fluid phase 1 and 2 respectively and $\boldsymbol{n}_1$ is the interface unit normal pointing from fluid 1 into fluid 2. Since $\boldsymbol{n}_1 = -\boldsymbol{n}_2$ and $\boldsymbol{T}_1 = \boldsymbol{T}_2$ the surface integral in equation 2.5-39 is zero and equation 2.5-33 becomes:

$$\int\limits_{\Gamma_{12}} \phi^v[(\boldsymbol{\pi}_1 - \boldsymbol{\pi}_2) \cdot \boldsymbol{n}_1]dS = \int\limits_{\Omega} \nabla\phi^v \boldsymbol{T} dV \tag{2.5-40}$$

Thus when interfacial tension is implemented using the CSS model the momentum balance simply yields:

$$\int_{\Omega} [\nabla \phi^v \cdot (\boldsymbol{\pi} - \boldsymbol{T})] dV + \int_{\Omega} \phi^v \boldsymbol{f} dV = \mathbf{0} \tag{2.5-41}$$

where the capillary pressure tensor is only non-zero in the proximity of the interface. Hence when the CSS model is used only first order derivatives of the F-field need to be calculated. Since the F-field is not smoothed prior to the calculation of interface normals in our implementation the capillary pressure tensor is calculated as:

$$\boldsymbol{T} = -\sigma \left(\boldsymbol{I} \, |\nabla F| - \frac{\nabla F \otimes \nabla F}{|\nabla F|} \right) \tag{2.5-42}$$

In summary the CSS interfacial tension model is included into the momentum balance as follows:

- The surface integral is approximated according to equation 2.5-40.
- The capillary stress tensor $\boldsymbol{T}$ is calculated according to equation 2.5-42.

2.5.2 Discretization of the interfacial tension terms

In this section more details on the discretization of the interfacial tension model is given. Here an explanation is also given on the calculation of interfacial tension when the 2-level method is applied. We will start by going through the discretization procedure for the CSS model in the case where the finite element and VOF mesh coincide.

2.5.2.1 Discretization of the CSS model

Since the CSS model only requires calculation of first order derivatives of the F-field the procedure explained in section 2.4.2 for calculating the interface normal can be used directly. Hence the F-field is first interpolated onto the linear nodes of the VOF cells using the ghost mesh. This leads to a new F-field, say F^*, such that the interface normal at any position inside an element m belonging to the interface region can be calculated as:

$$n_k = \left. \frac{\partial F^*}{\partial x_k} \right|_{\boldsymbol{x}_m} = \sum_{i=1}^{8} \left. \frac{\partial L_i}{\partial x_k} \right|_{\boldsymbol{x}_m(\xi,\eta,\zeta)} F_i^*, \quad k = 1, \dots 3 \tag{2.5-43}$$

If we denote the discretely calculated capillary stress tensor $\boldsymbol{T}_s$ then the discrete equivalent of the RHS of equation 2.5-40 becomes:

$$\sum_{n=1}^{N_e} \sum_{k=1}^{3} \left(\sum_{k_1=1}^{3} \int_{\Omega_n} \frac{\partial N^v_{i(I,n)}}{\partial x_{k_1}} \boldsymbol{T}^n_{s,k_1 k} dV \right) \boldsymbol{\delta}_k, \quad k = 1, \dots 3 \tag{2.5-44}$$

Then equation I in the global equation system will get a contribution from interfacial tension if node I is connected to elements which have one or more vertex nodes with $0 < F < 1$.

Equation 2.5-44 is calculated using numerical integration as explained in section 2.2.6. It is noted that since the interfacial tension calculations are based on nodal values of the F-field, i.e. F^*, a smoothed F-field can be applied directly without any changes to the code simply by applying the smoothed F-field onto the nodes of the VOF cells. In fact the interpolation procedure based on the ghost mesh does have a smoothing effect because the resultant field F^* is less discontinuous across the interface than the original F-field.

2.5.2.2 Interfacial tension calculations and the 2-level method

When the 2-level algorithm is applied each finite element is subdivided into $N_1N_2N_3$ number of smaller VOF cells, cf. section 2.4.2.4. This means that in finite elements where a part of the interface resides the F-field will in general not be constant as the value of F varies from VOF cell to VOF cell. In order to use the information on the VOF cell level in our finite element calculations, contributions from each VOF cell in a given finite element are summed together. For a given finite element n, equation 2.5-44 becomes:

$$\sum_{k=1}^{3}\left(\sum_{k_1=1}^{3}\int_{\Omega_n}\frac{\partial N^v_{i(I,n)}}{\partial x_{k_1}}\boldsymbol{T}^n_{s,k_1k}dV\right)\boldsymbol{\delta}_k=\sum_{k=1}^{3}\left(\sum_{k_1=1}^{3}\sum_{m=1}^{N_1N_2N_3}\int_{\Omega_m}\frac{\partial N^v_{i(I,n)}}{\partial x_{k_1}}\hat{T}^m_{s,k_1k}dV\right)\boldsymbol{\delta}_k,$$

$$I=1,\dots NN,\quad \sum_{m=1}^{N_1N_2N_3}\Omega_m=\Omega_n \qquad (2.5\text{-}45)$$

Hence when the 2-level algorithm is applied equation I in the global system of equations will get a contribution from interfacial tension if it is connected to one or more finite elements with an interior VOF cell which can be regarded as part of the interface region. When the integral in equation 2.5-45 is evaluated numerically appropriate integration points and weights need to be defined. This is done by scaling the integration points and weights, used for the standard square hexahedron ($[-1,1]\times[-1,1]\times[-1,1]$), such that they apply for a rectangular element with side lengths $(2/N_1, 2/N_2, 2/N_3)$ positioned somewhere inside the standard element. The values of the quadratic finite element shape functions and the values of the gradients of F are then evaluated at the integration points defined in the interior of the VOF cells. It is noted that Rudman (1998) and Li et al. (2000) have also used a 2-level approach in connection with VOF calculations. However, in both these methods the VOF mesh is restricted to being twice as fine as the calculational mesh. Furthermore the main purpose of these 2-level algorithms is to increase accuracy during advection.

2.6 Viscosity and density

For a given finite element n the viscosity and density are scaled linearly with the volume fraction F_n of the 'disperse' phase:

$$\mu_n = F_n\mu_2 + (1 - F_n)\mu_1$$

and

$$\rho_n = F_n\rho_2 + (1 - F_n)\rho_1$$

where index 2 refers to fluid 2 which is defined as the fluid tracked by the F-function and index 1 refers to the other fluid. F_n is in this case the value of F in finite element n which is calculated as:

$$F_n = \frac{\sum_{m=1}^{N_1 N_2 N_3} F_m V_m}{V_n}$$

where F_m is the value of F in 'local' VOF cell m and V_m is the total volume of VOF cell m. V_n is the total volume of finite element n.

This means that the viscosity and density are calculated as being constant in each finite element also when the 2-level approach is used.

It is noted that for interface finite elements one could base the calculations of the coefficients in the momentum and continuity equations on summations over the VOF sub volumes and thereby letting μ and ρ vary within each interface finite element. This is discussed further in section 3.3.2.

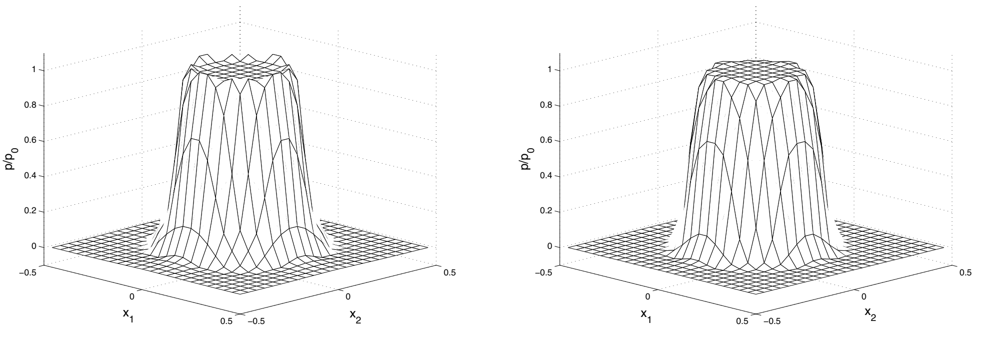
2.7 Verification of interfacial tension: Internal pressure of a static drop

For a 3 dimensional stagnant drop with radius r_0 and surface tension σ the difference between the internal and external pressure is given by Laplace's law, i.e.:

$$p = \frac{2\sigma}{r_0}$$

This simple problem will be used here as a first validation of the Stokes flow calculations with the presence of interfacial tension. Due to symmetry only 1/8 of the total geometry needs to be considered in this problem. On the domain boundaries the normal components of the velocities are set equal to zero and the domain size is $[0, 0.5]$ in all directions while the drop radius is $r_0 = 0.25$. The interfacial tension is $\sigma = 1{}^{\mathrm{N}}\!/\!{}_{\mathrm{m}}$ and the viscosity of the drop and the surroundings is $\mu = 1$Pas. Tests are made with both the CSS model and the CSF model. In both cases meshes with three different resolutions are tested. The mesh elements have side lengths $h = 1/12$, $h = 1/24$ and $h = 1/48$ respectively. Furthermore a test is made of the 2-level method. In this test the medium calculational mesh ($h = 1/24$) is used. Calculations are carried out using VOF meshes which are two, four and eight times as fine as the calculational mesh.

In Figure 2.17 (a) and (b) the pressure distributions for the drop on the (x_1, x_2)-symmetry plane obtained using the CSS and CSF methods can be seen (external pressure specified to zero). Here the medium mesh is used. The pressures in the figure are normalized by the Laplace pressure $p_0 = 8\text{Pa}$. In both Figure 2.17 (a) and (b) the interior drop pressure is approximately constant with $p = p_0$. However, when comparing the results from the CSS and CSF model, differences in the pressure distribution along the interface region are observed. Especially one sees considerably larger fluctuations in the pressure along the drop interface when the CSS model is used. Hence for this problem the CSF model yields a more uniform internal pressure distribution than the CSS model. In Table 2.7 the normalized interior drop pressure can be seen on all three meshes for both methods. These pressures are calculated as the average of the pressures in the interior elements of the sphere, i.e. elements which are not part the interface region. Comparison of the results from the CSS and CSF model does not reveal any large differences in the average internal drop pressure between the two methods. It is noted that one can obtain smoother pressure distributions by smoothing the F-field before calculating the interfacial tension forces, see e.g. (Brackbill et al. 1992). From these tests we can, however, conclude that the implementation of both interfacial tension methods gives the correct drop (bubble) pressure when the 1-level method is used.



(a)

(b)

Figure 2.17: Pressure distribution on the (x_1, x_2)-symmetry plane of a stagnant drop. (a) pressure distribution obtained using the CSS interfacial tension model and (b) Pressure distribution obtained with the CSF model. The pressures are normalized with the Laplace pressure $p_0 = 8\text{Pa}$ for a sphere with $r_0 = 0.25$ and $\sigma = 1\,{}^{\text{N}}\!/_{\text{m}}$ (medium mesh).

The next step is to investigate how the 2-level method affects the pressure distribution. Here the medium mesh is used for the finite element discretization. In Figure 2.18 the pressure distributions obtained using the CSS (a) and CSF (b) models with the 2-level method

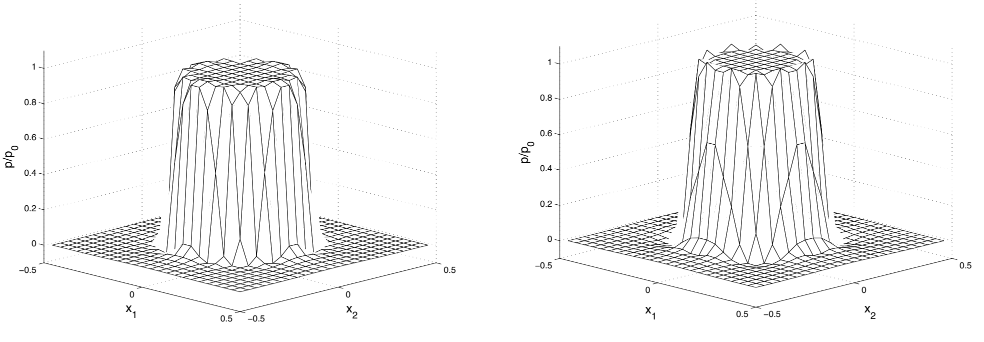
Element size h	p/p_0 CSS	p/p_0 CSF
1/12	1.079	1.096
1/24	0.992	0.999
1/48	0.975	0.977

Table 2.7: Interior drop pressure normalized by the Laplace pressure p_0 obtained using the CSS and CSF interfacial tension models.

h_{FE}/h_{VOF}	p/p_0 CSS	p/p_0 CSF
2	0.972	0.979
4	0.970	0.992
8	0.970	1.020

Table 2.8: Interior drop pressure normalized by the Laplace pressure p_0 obtained using the CSS and CSF interfacial tension models with various VOF cell sizes h_{VOF}. The medium mesh ($h_{FE} = 1/24$) was used for the FE discretization.

can be seen. Here the VOF mesh was eight times as fine as the FE mesh. Comparing the results in Figure 2.17 (a) and 2.18 (a) which are based on the same FE mesh but different VOF meshes indicates that the pressure near the drop interface is more uniform when the CSS model is used with the 2-level method. Furthermore the pressure transition region seems to be smaller in the 2-level case. In order to give a better illustration of this the pressures from Figure 2.17 (a) and Figure 2.18 (a) along the $(x_1, 0, 0)$-line are plotted in Figure 2.19. This figure clearly shows a change in the pressure distribution even though the FE meshes are the same for both simulations. The most striking difference is that, in the 2-level case, the interface becomes more sharp in terms of the pressure distribution. On the other hand comparison of 2.17 (b) and 2.18 (b) shows the opposite behavior when the CSF method is used. However, in all cases the internal pressure is constant and $p \simeq p_0$. In Table 2.8 the internal drop pressure relative to the Laplace pressure can be seen from simulations with the two interfacial tension models and various 2-level meshes. Here it is seen that when the CSS model is used, the relative pressure is more or less constant at $p/p_0 = 0.97$, which is close to the pressure obtained using the 1-level method and the finest mesh, c.f. Table 2.7. When the CSF model is used together with the 2-level method the pressure increases as the VOF mesh is refined. It is noticed that the results from the 2-level simulation where the VOF mesh is twice as fine as the FE mesh yields an average internal pressure which is close to that obtained on the finest mesh with the 1-level method (the VOF mesh resolution is the same for those 2 cases). Hence the tests carried out here show that the implemented interfacial tension models lead to pressure distributions which are in good agreement with the theoretical result also when the 2-level approach is used.



(a) (b)

Figure 2.18: Pressure distribution on the (x_1, x_2)-symmetry plane of a stagnant drop when the 2-level method is applied. In the simulations the VOF mesh was eight times finer than the medium FE mesh. (a) pressure distribution obtained using the CSS interfacial tension model and (b) Pressure distribution obtained with the CSF model. The pressures are normalized with the Laplace pressure $p_0 = 8\text{Pa}$ for a sphere with $r_0 = 0.25$ and $\sigma = 1^{\mathrm{N}}/_{\mathrm{m}}$.

2.8 Spurious currents

The spurious currents associated with the CSS and CSF models will be explored here. Results are presented both for the 1-level and 2-level procedures. It is noted that all results presented in this subsection were obtained using the simulations in the previous subsection and that the results are based on the first time step (i.e. $t = 0\text{s}$).

In Figure 2.20 (a)-(c) the spurious currents on the (x_1, x_2)-symmetry plane obtained using the CSS method on the three different meshes are shown. Figure 2.20 (d)-(f) shows the corresponding results obtained using the CSF method. When going from (a) to (c) and (d) to (f) one sees that the flow becomes more localized to the vicinity of the interface. Furthermore the maximum magnitude of the velocity vector seems to decrease somewhat with mesh refinement. If results obtained using the CSS and CSF method are compared the most distinct difference is that the CSS method seems to produce a flow which is somewhat more uniform around the interface than the CSF method. Typically the magnitude of the spurious currents is quantified by calculating the maximum (or infinity) norm, e.g. (Brackbill et al. 1992, Lafaurie et al. 1994, Renardy, Cristini & Li 2002), although the 1-norm and 2-norm are also used. Here we will use the maximum norm. In Table 2.9 the maximum norm of the velocity can be seen for the 1-level results. These results show that the CSF method leads to a smaller maximum velocity than the CSS method on all three meshes. Furthermore one observes that the maximum velocity decreases as the mesh

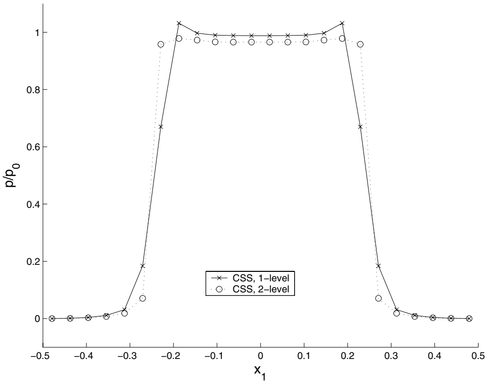


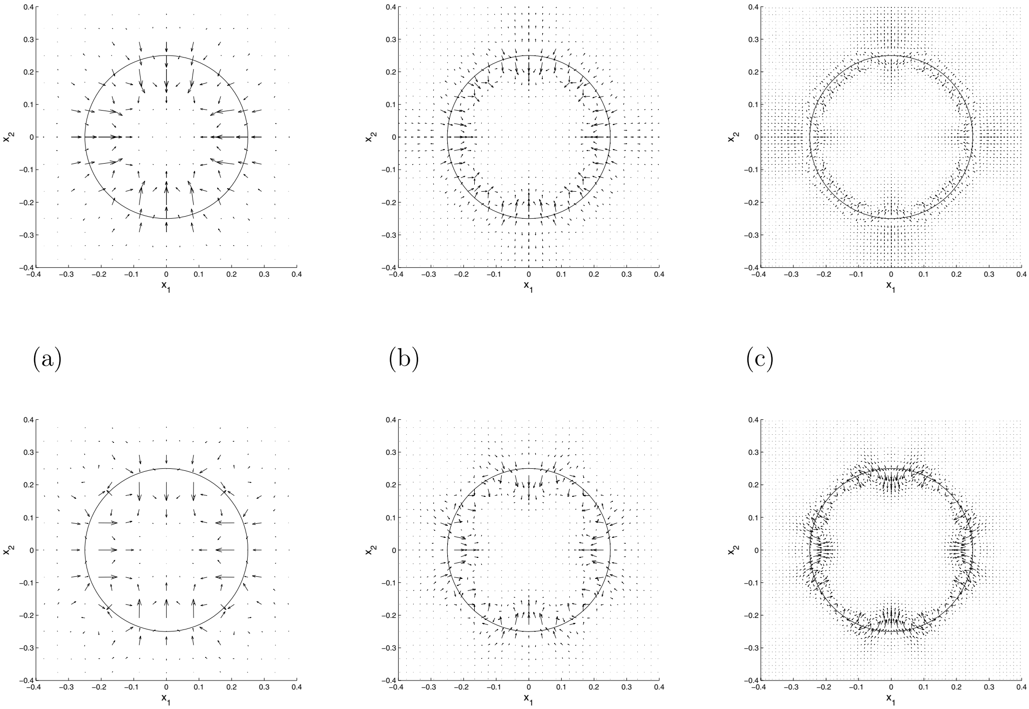
Figure 2.19: Pressure variation along the $(x_1, 0, 0)$-line when the CSS model is used with the medium mesh. Shown are pressures obtained using the medium mesh with 1-level calculations and 2-level calculations with $h_{FE}/h_{VOF} = 8$.

Element size h	$\|\|\boldsymbol{v}\|\|_\infty$ CSS	$\|\|\boldsymbol{v}\|\|_\infty$ CSF
1/12	0.0536	0.0288
1/24	0.0352	0.0181
1/48	0.0279	0.0148

Table 2.9: Maximum norm of the spurious current velocities obtained using the CSS and CSF model with the 1-level method.

is refined. It is noted here that usually the magnitude of the spurious currents does not converge towards zero with mesh refinement unless some special methods are applied. One such method is the PROST method of Renardy, Cristini & Li (2002) where the spurious currents are at least two orders of magnitude smaller than those obtained with the CSS and CSF methods. For the CSS and CSF models, numerical experiments carried out by e.g. Lafaurie et al. (1994), Scardovelli & Zaleski (1999) and Renardy, Cristini & Li (2002), show that the maximum velocity of the spurious currents is on the order of $0.01\sigma/\mu$. In our simulations $\sigma/\mu = 1$ and thus the results in Table 2.9 agree with the scaling found in the literature.

In Figure 2.21 velocity vector plots of the spurious currents from the 2-level calculations with the CSS and CSF models can be seen. Figure 2.21 (a)-(c) shows the CSS results where the VOF mesh is two times, four times and eight times as fine as the FE mesh (medium mesh). Figure 2.21 (d)-(f) shows the corresponding CSF results. The velocity plots from the 2-level CSS calculations indicate that the flow becomes more localized to the true interface position as the VOF mesh is refined. This is especially evident by comparing the 1-level result in Figure 2.20 (b) and the 2-level result in Figure 2.21 (c). The velocity plots in Figure 2.21 (d) and (e) show a similar behavior when compared to the 1-level result

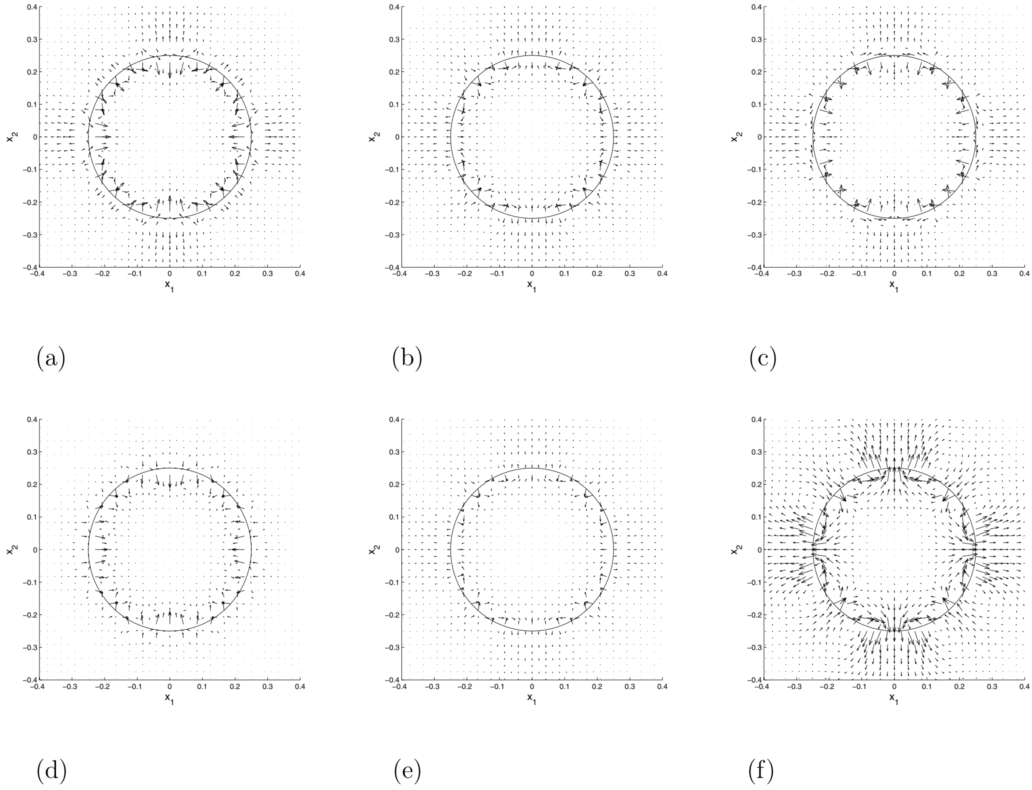


(a) (b) (c)

(d) (e) (f)

Figure 2.20: Velocity vector plots showing spurious currents on the (x_1, x_2)-symmetry plane of the drop. (a)-(c) results obtained using the CSS model on the coarse, medium and fine mesh respectively. (d)-(f) results obtained using the CSF model. The 1-level method was used in all simulations.

in 2.20 (e). On the other hand the result in Figure 2.21 (f) shows a large increase in the magnitude of the velocities when the finest VOF mesh is used. This behavior indicates that increasing the h_{FE}/h_{VOF} ratio does not necessarily lead to better results. In Table 2.10 the maximum norms of the velocities can be seen from the 2-level experiments. Here we see that the maximum velocity does not change with VOF mesh refinement when the CSS method is used. Furthermore the maximum velocity is comparable to that obtained from the 1-level calculation on the finest mesh, cf. Table 2.9. The results from the CSF calculations in Table 2.10 show that the smallest maximum velocity is obtained when the VOF mesh is four times finer than the FE mesh which Figure 2.21 (e) also indicates. Again, as Figure 2.21 (f) suggests, the maximum velocity is largest on the finest VOF mesh.



(a) (b) (c)

(d) (e) (f)

Figure 2.21: Velocity vector plots showing spurious currents on the (x_1, x_2)-symmetry plane of the drop when the 2-level approach is used. (a)-(c) results obtained using the CSS model with $h_{FE}/h_{VOF} = 2$, 4 and 8 respectively. (d)-(f) results obtained using the CSF model.

h_{FE}/h_{VOF}	$\|\|\boldsymbol{v}\|\|_{\infty}$ CSS	$\|\|\boldsymbol{v}\|\|_{\infty}$ CSF
2	0.0244	0.0202
4	0.0244	0.0167
8	0.0244	0.0247

Table 2.10: Maximum norm of the spurious current velocities obtained using the CSS and CSF model with the 2-level method. The medium mesh was used in the FE discretization.

The implication of the spurious currents is that the interface topology of an otherwise stagnant drop will change with time. One can relate the viscous and interfacial tension forces through the Ohnesorge number:

$$Oh = \frac{\mu}{\sqrt{\sigma \rho r_0}}$$

When the Ohnesorge number is combined with the scaling of the spurious currents with interfacial tension and viscosity the order of the Reynolds number based on spurious currents becomes:

$$Re = \frac{0.01}{Oh^2}$$

and thus for small Ohnesorge numbers the spurious currents will be large. In practice this means that for a drop undergoing deformation in an externally applied flow the spurious currents can lead to flow instabilities if the Ohnesorge number is small and the applied flow is weak, cf. also (Renardy, Cristini & Li 2002).

In summary this investigation of the spurious currents associated with the implemented CSS and CSF models agrees well with numerical experiments carried out by other researchers, e.g. (Brackbill et al. 1992, Lafaurie et al. 1994, Scardovelli & Zaleski 1999, Renardy, Cristini & Li 2002). Furthermore when the CSS model is used some decrease in the magnitude of the spurious currents is observed for the 2-level results compared to the 1-level results. This behavior is also seen for the CSF model for some FE-VOF mesh combinations. However, an increase in the velocity field is seen when the CSF method is used together with the finest VOF mesh and thus care has to be taken when using the 2-level method with highly refined VOF meshes.

Chapter 3

Simulations of single drop deformations

In order to investigate the validity of the implemented FE-VOF algorithms a number of simulation tests are carried out on single drops suspended in a second immiscible liquid on which a flow is applied. These tests consist of checking boundary wall effects (subsections 3.2.1) effects of mesh type (subsection 3.2.2), temporal convergence (subsection 3.2.3), spatial convergence of 1-level and 2-level meshes (subsection 3.2.4), simple shear simulations with drop break-up and merging (section 3.3) and simulations in planar elongation (section 3.4). In all simulations the CSS interfacial tension model is used and the adjusted time step parameter Δt^* is used during the advection step - unless otherwise stated. Before we proceed to the results from these tests an introduction to the computational domain, boundary conditions and time step size needs to be given.

3.1 The computational domain, BC's, and time step size

All simulations carried out in this project are based on a simple hexahedral computational domain (a computational box). Figure 3.1 shows an example of a finite element mesh used for the discretization of the domain. In order to save necessary computation time the mesh consists of elements with variable side lengths. Elements are then concentrated in the region where the drop is expected to be during the simulations, typically in the centre, while the element density decreases as one approaches the walls of the box. The computational box thus consists of a fine region with regular hexahedral elements in the center and a coarse region with hexahedral elements extending from the fine region towards the walls. The dimensions of the domain are $([-L_1, L_1] \times [-L_2, L_2] \times [-L_3, L_3])$ where L_1, L_2 and L_3 are the half lengths of the box, cf. Figure 3.1. In principle one can generate many types of flow fields in the computational box by applying appropriate boundary conditions. However, here we will focus on planar flow fields of the type:

$$
\begin{aligned}
v_1 &= \alpha G x_1 + (1 - |\alpha|) G x_2 \\
v_2 &= -\alpha G x_2 \\
v_3 &= 0
\end{aligned}
\tag{3.1-1}
$$

where $G = |\dot{\epsilon}| + |\dot{\gamma}|$ and $\alpha = \dot{\epsilon}/G$. Here $\dot{\epsilon}$ is an elongation rate and $\dot{\gamma}$ is a shear rate. If $\dot{\gamma} = 0$ the flow corresponds to planar elongation and if $\dot{\epsilon} = 0$ the flow corresponds to pure shear. If both the shear and elongation rates are larger than zero the flow is a mixture of both. It is noted that due to the definition of the flow field the shearing direction does not depend on the sign of $\dot{\gamma}$. However if $\dot{\epsilon}$ is negative the elongational contribution to the flow changes direction. For any combination of $\dot{\epsilon}$ and $\dot{\gamma}$ the flow will have a stagnation point in the centre $(x_1, x_2, x_3) = (0, 0, 0)$ of the box. In all single drop simulations the drop is initially placed in the centre of the domain which guaranties that the drop will not translate out of the box. It is noted that a symmetry plane exists at $x_3 = 0$ which halves the required number of elements.

In order to generate this flow by the Stokes solver the velocity is specified according to (3.1-1) on boundary nodes at $x_1 = -L_1$, $x_1 = L_1$, $x_2 = -L_2$ and $x_2 = L_2$. On the symmetry plane and the wall at $x_3 = -L_3$, $v_3 = 0$ is specified while the remaining velocity components are unspecified. The motivation for this choice of BC's is that one can easily control the flow field ranging from pure shear to pure planar elongation just by changing the shear and elongation rate. This also means that one can specify time dependent shear and elongation rates which we will use later to simulate more complex flow situations. Unless otherwise stated a variable time step is used during the simulations. Since, only interface cells and their nearest neighbours, i.e. the active cells, need to be considered during a given advection step a routine searches through all the active cells to find the CFL number:

$$
\mathrm{CFL} = \max\left(\frac{|f_{j,i}|}{V_j}\right)\Delta t, \quad j = 1, \ldots, N_{active}
$$

where N_{active} is the number of active cells, V_j are VOF cell volumes and $f_{j,i}$ are VOF cell face fluxes. The user then specifies a priori what the CFL should be. Thus for, say time step number k, the time step size is:

$$
\Delta t^k = \frac{\mathrm{CFL}}{\max\left(\frac{|f_{j,i}|}{V_j}\right)^k}
$$

where CFL is a user input. It is noted that if the flow at some point ceases, e.g. due to reversal in the flow direction, the time step becomes infinite. Therefore a maximum time step-size needs to be specified.

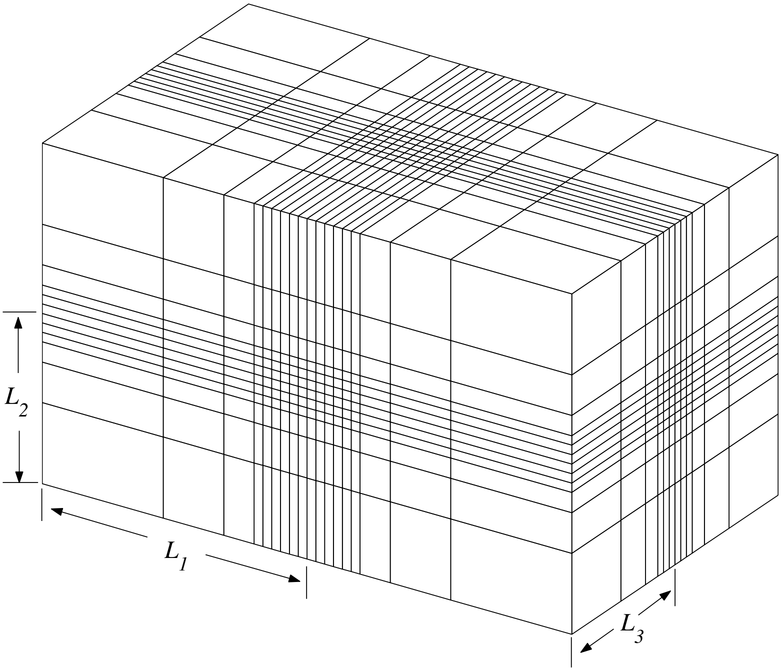


Figure 3.1: Structured mesh with variable mesh size. The dimensions of the rectangular domain are given by half side lengths L_1, L_2, L_3. The mesh shown has a central fine region with constant element size and a variable coarse part which extends out to the domain boundaries.

3.2 Convergence of the FE-VOF code

In order to check the convergence behavior of the FE-VOF code a standard numerical experiment (benchmark problem) is constructed. This benchmark is based on pure shear of a drop with initial radius $r_0 = 0.25$ and viscosity $\mu_d = 1$Pas suspended in a second liquid with viscosity $\mu_c = \mu_d$. The interfacial tension coefficient is $\sigma = 0.625$N/m and the shear rate is $\dot{\gamma} = 1\text{s}^{-1}$. Furthermore the drop is neutrally buoyant. These parameters correspond to $Ca = 0.40$ and $\lambda = 1$ which is approximately 10% below the critical point for a drop sheared in an infinite medium, i.e. no wall effects. The reason for the choice of these test conditions is as follows:

1. Flow field: Both experimental and numerical results on single drop deformation behavior in simple shear are readily available in the literature.

2. Ca number: In order to quantify the deformation of the drop it is advantageous that the drop reaches a steady-state and that the deformation is as large as possible

but still quantifiable by the Taylor deformation parameter. This should ensure good accuracy in the deformation calculations.

3. Viscosity ratio: As the coefficient matrix is constant for $\lambda = 1$ factorization needs only to be carried out once outside the time integration loop which speeds up computation considerably.

A characteristic drop time is defined as $\hat{t} = r_0\mu_c/\sigma$ which is used for defining a dimensionless time as $t^* = t/\hat{t}$. The dimensionless time will be used when transients are studied.

3.2.1 Wall effects

The mesh used in these tests is as follows: The size of the fine mesh region is $[-24h_{fine}, 24h_{fine}] \times [-8h_{fine}, 8h_{fine}] \times [-8h_{fine}, 0])$ where $h_{fine} = r_0/7$. The coarse part of the mesh consists of 4 elements extending from the fine mesh region towards each wall. The coarse mesh region is set up by halving the element side lengths when going from the walls to the fine region. The side lengths of the elements closest to the boundaries are then chosen in such a way that the size of the coarse mesh region agrees with the size of the region extending from the fine mesh to the walls. Hence, as the wall separation is changed the fine mesh region is unaltered while the sizes of the elements in the coarse region are scaled. In these tests the variable time step size is chosen such that CFL $= 1/4$.

In Figure 3.2 an example of the results obtained during the shear experiment can be seen. The figure shows both the 3D drop shape (PLIC faces), figure (a)-(c), and the (x_1, x_2)-slice through the symmetry plane of the drop, figure (d)-(f), for three dimensionless times during the deformation. Also shown is the flow field on the symmetry plane. In this example the size of the computational domain is $(L_1, L_2, L_3) = (24r_0, 24r_0, 24r_0)$. From Figure 3.2 it is seen that as the drop deforms from its initial spherical shape the flow field starts to circulate in the interior of the drop. This circulation of the fluid eventually results in a steady state where the drop shape no longer changes (c) and (f). By measuring the length L and breadth B of the drop on the symmetry plane at every time step one can obtain the transient deformation parameter D. Two series of experiments have been carried out. In the first series the total domain length in the x_1 direction is kept constant at $24r_0$ while the length in the x_2 and x_3 direction is varied from $2r_0$ to $24r_0$. In the second experimental series the length in the x_1 direction is varied while the x_2 and x_3 lengths are kept constant at $24r_0$.

In Figure 3.3 the deformation as a function of the dimensionless time can be seen for the simulations carried out with varying L_2 and L_3. The data in the figure show that when $(L_2, L_3) = (2r_0, 2r_0)$ and $(L_2, L_3) = (4r_0, 4r_0)$ the deformation does not reach a limiting value and the drop will at some point break up (break up occurs at around $t^* = 20$ and $t^* = 130$ for the two cases). For the remaining geometries the drop does reach a limiting deformation D_s. In these cases the deformation curve is seen to increase fast in the initial part of the experiments ($t^* < 5$). Hereafter the curves start leveling out and at around

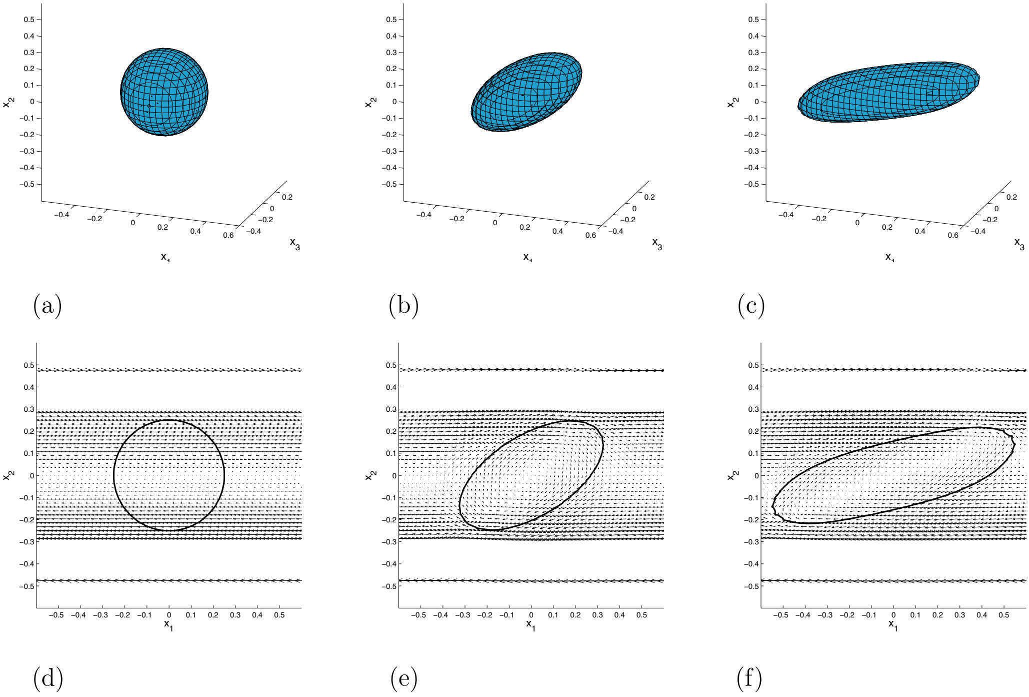


Figure 3.2: Deformation of a drop in simple shear flow at dimensionless times $t^* = 0$, $t^* = 3$ and $t^* = 70$. The dimensions of the computational box were $(L_1, L_2, L_3) = (24r_0, 24r_0, 24r_0)$. $Ca = 40$ and $\lambda = 1$. (a)-(c) PLIC face reconstruction of the the interface. (d)-(f) cross-sectional slice through symmetry plane of the drop, also shown is a vector plot of the velocities. It is noted that only a part of the domain is shown.

$t^* = 40$ a steady state is reached. From these data it is clear that the limiting value of D decreases as the wall separation is increased, however, for $(L_2, L_3) = (16r_0, 16r_0)$ and $(L_2, L_3) = (24r_0, 24r_0)$ the curves almost overlap, indicating that wall effects are becoming negligible. In Table 3.1 the steady-state deformation values (D_s) taken at the end of the numerical experiments are given. From the table it is seen that we obtain $D_s = 0.5803$ when $(L_1, L_2, L_3) = (24r_0, 24r_0, 24r_0)$.

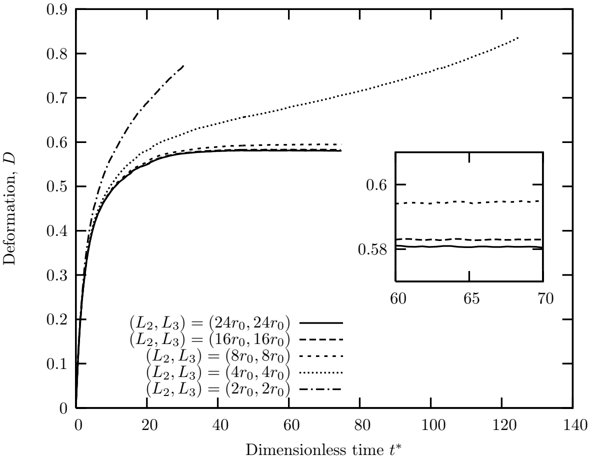


Figure 3.3: Influence of wall separation in the x_2 and $_3$ direction. The length of half of the domain is $L_1 = 24r_0$ in all simulation.

(L_2, L_3)	$(24r_0, 24r_0)$	$(16r_0, 16r_0)$	$(8r_0, 8r_0)$	$(4r_0, 4r_0)$	$(2r_0, 2r_0)$
D_s	0.5803	0.5825	0.59463	unstable	unstable

Table 3.1: Steady-state values of D for various domain sizes in the x_2 and x_3 direction. L_2 and L_3 correspond to half the total height and width of the domain. The length L_1 of the domain is $24r_0$ in all cases.

In Figure 3.4 the transient deformation can be seen for the second experimental series, i.e. where (L_2, L_3) are kept constant at $24r_0$. The steady-state deformations from these simulations can be seen in Table 3.2. By comparing Figure 3.4 with Figure 3.3 it is seen that the results obtained in the second experimental series are similar to those obtained for the first series. The drop is stable in all cases except when $L_1 = 4r_0$. Furthermore we again see that the curves more or less overlap when $L_1 = 16r_0$ and $L_1 = 24r_0$.

These numerical experiments thus suggest that a wall separation corresponding to $(L_1, L_2, L_3) = (24r_0, 24r_0, 24r_0)$ leads to negligible wall effects. In the VOF computations of Li et al. (2000) the steady state deformation at $Ca = 0.40$ and $\lambda = 1$ is found to be 0.541, however, in their calculations the F-field is smoothed prior to interfacial tension

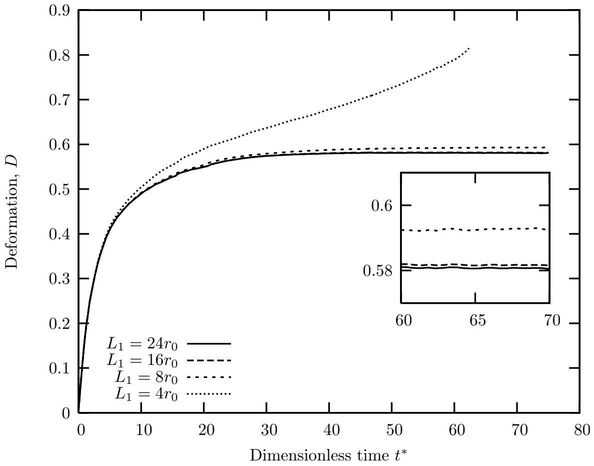


Figure 3.4: Influence of boundary effects in the x_1 direction. The half height and width of the domain is $(L_y, L_z) = (24r_0, 24r_0)$ in all simulations.

L_x	$16r_0$	$8r_0$	$4r_0$
Steady-state D	0.5812	0.5927	unstable

Table 3.2: Steady-state values of D for various domain lengths in the x_1 direction. L_1 corresponds to half the total length of the domain. The half-height L_2 and half-width L_3 of the domain are $24r_0$ in all simulations.

calculations. It is noted that Li et al. (2000) find that a domain size of $(8r_0 \times 16r_0 \times 4r_0)$ is sufficient to neglect wall effects. However, in their simulations a regular finite difference grid is used and thus all elements have equal size. Furthermore they apply periodic BC's in the x_1 and x_3 direction which of course changes the way the walls affect the flow. In the current implementation it is also possible to apply periodic BC's on these walls, however, here we are not interested in that because we want to be able to carry out simulations in mixed flow fields.

3.2.2 Refinement of the coarse part of the mesh

In order to investigate how the mesh concentration in the coarse mesh region affects the flow field, simulations are carried out on three different meshes. In these simulations the fine mesh region is the same as that used in the previous subsection while the coarse region consists of either 2, 4 or 6 elements extending from the fine region out to domain boundary. These meshes are denoted $N_{coarse} = 2$, $N_{coarse} = 4$ and $N_{coarse} = 6$ and a (x_1, x_2)-slice of the lower left part of the meshes can be seen in Figure 3.5.

The dimensions of the domain are $(L_1, L_2, L_3) = (24r_0, 24r_0, 24r_0)$ and the variable time

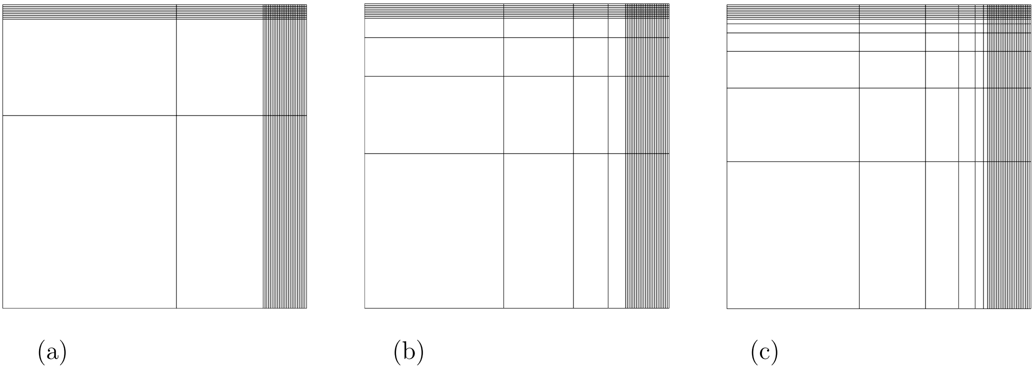


Figure 3.5: Bottom left slice in the (x_1, x_2)-plane of the meshes used for studying the effect of refinement on the coarse mesh region. (a) $N_{coarse} = 2$, (b) $N_{coarse} = 4$, (c) $N_{coarse} = 6$.

step is specified by setting CFL = 1/4. In Figure 3.6 the transient deformation obtained from the simulations can be seen. The results show that when $N_{coarse} = 2$ a steady state is not obtained and the drop will therefore eventually break-up. The simulation with $N_{coarse} = 4$ is the same as the one carried out in the previous section with $(L_1, L_2, L_3) = (24r_0, 24r_0, 24r_0)$ and thus reaches a steady state. By comparing the transient deformation data for $N_{coarse} = 4$ and $N_{coarse} = 6$ it is seen that the difference is very small which together with the results in the previous section indicates that a mesh with $N_{coarse} = 4$ and $(L_1, L_2, L_3) = (24r_0, 24r_0, 24r_0)$ is adequate for carrying out the shear experiment with negligible wall effects.

3.2.3 Influence of time-step size

The influence of the time step size Δt is investigated by carrying out simulations with CFL ranging from 1/2 to 1/8. The mesh used in these simulations is $N_{coarse} = 4$ and $(L_1, L_2, L_3) = (24r_0, 24r_0, 24r_0)$ with the fine region being the same as that described in subsection 3.2.1. The results can be seen in Figure 3.7. Here it is evident that for the mesh and parameters used there is practically no difference in the results when going from the largest to smallest time step tested. The relative loss in volume of the advected fluid is $9.3 \cdot 10^{-2}$ % for largest time step and $6.5 \cdot 10^{-3}$ % for the smallest time step. In (Li et al. 2000) a loss of $2.6 \cdot 10^{-3}$ % is reported. However, our largest time step is approximately 2 orders of magnitude larger than that used in (Li et al. 2000). The results obtained here therefore suggest that CFL = 1/2, which is close to the maximum time step allowed, is adequate. However, it is noted that if a mesh with a coarser fine region is used care must be taken because the actual time step will be larger than that used for generating the results in Figure 3.7, i.e. the CFL number depends on the mesh element size.

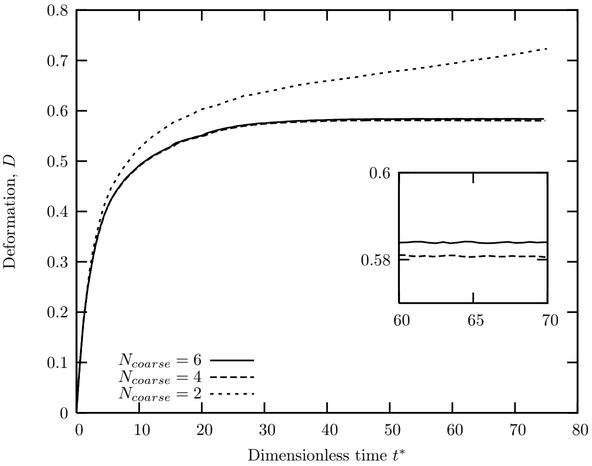


Figure 3.6: Influence of element refinement on the coarse part of the mesh. Each simulation is carried out with the same number of coarse elements in all principal directions, i.e. $N_{coarse,1} = N_{coarse,2} = N_{coarse,3} = N_{coarse}$.

mesh	h_{fine}	CFL	DOF
1	$r_0/3$	1/16	8815
2	$r_0/6$	2/16	32809
3	$r_0/9$	3/16	80203
4	$r_0/12$	4/16	158773

Table 3.3: Element side lengths of the fine mesh region for the four meshes used for testing spatial convergence of the 1-level method. Also shown are the CFL numbers specified for each mesh and the total number of DOF associated with each mesh.

3.2.4 The fine mesh region and the 2-level algorithm

Now the convergence with respect to mesh refinement of the fine mesh region is investigated. These tests include both the 1-level and 2-level methods. In the 1-level tests, simulations are carried out on four meshes with different mesh concentration in the fine region. The dimensions of the fine region are those given in subsection 3.2.1 while the mesh element side lengths (h_{fine}) are varied from $r_0/3$ to $r_0/12$, cf. Table 3.3. All meshes used have $N_{coarse} = 4$ and $(L_1, L_2, L_3) = (24r_0, 24r_0, 24r_0)$. In order to obtain time step sizes, which are approximately the same in all simulations, mesh dependent CFL numbers are specified, these are given in Table 3.3.

In Figure 3.8 the transient deformation can be seen for the simulations carried out using the 1-level method. The simulations show that the drop is unstable when the coarsest mesh is applied. For the remaining meshes the drop is stable and the results show that the steady-

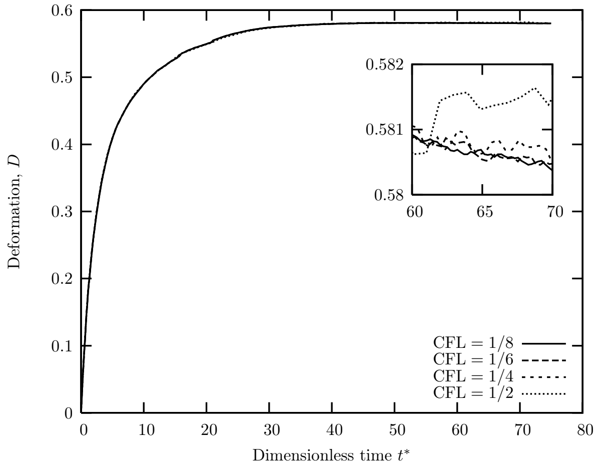


Figure 3.7: Influence of time-step size on droplet deformation behavior.

mesh	D_s
1	Unstable
2	0.598
3	0.560
4	0.550

Table 3.4: Steady-state deformation parameters obtained from simulations with the meshes in Table 3.3.

state deformation parameter decreases as the mesh is refined, cf. also Table 3.4. The data in Figure 3.8 also indicate that the transient drop shape converges with mesh refinement. In the last column of Table 3.3 the DOF associated with each mesh are also shown. The total simulation time used for mesh 1 in Table 3.3 was approximately 2 hours while it was on the order of 3 days for mesh 4 (using 1 processor). And judging from the results in Figure 3.8 and Table 3.4 the mesh needs to be at least as fine as mesh 4 if simulations are to be carried out in the converged region. This means that in order to carry out realistic simulations using the 1-level method computation times become exceedingly large. It is noted that the simulations presented until now are based on relatively small drop deformations. If simulations are carried out which involve large deformations leading to break-up the fine mesh region needs to be extended in the deformation direction which requires a larger number of elements. Furthermore if the two phases have different viscosities each time step also requires a re-factorization of the coefficient matrix which is very time consuming. Thus in summary the range of numerical experiments which we can carry out using the 1-level method is heavily limited by the computational costs.

This brings us to the 2-level method because here one can choose a relatively coarse

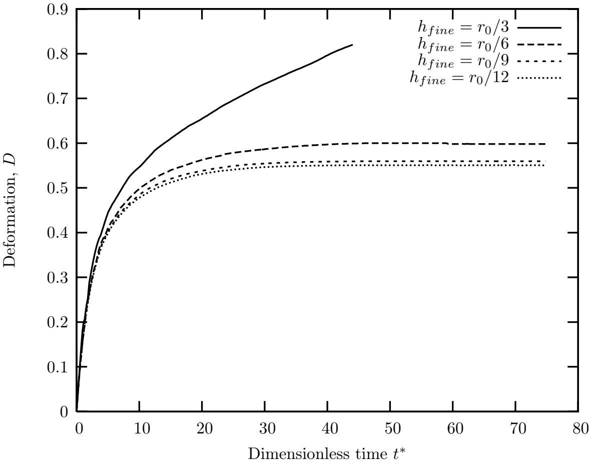


Figure 3.8: Transient deformation obtained on meshes with different element concentrations in the fine region (1-level).

computational mesh while still having a fine resolution on the VOF mesh level. Since, for a given time step, all the VOF calculations are limited to interface cells and their neighbours these calculations are generally very fast relative to the flow field calculations (usually less than 5% of the total computation time). This though of course depends on how fine the VOF mesh is relative to the FE mesh. It is noted that the interface normal and interface advection calculations are localized to a given VOF cell and thus it should be possible to parallelize this part of the code efficiently - this though has not been carried out here. In order to investigate the accuracy of the 2-level method compared to the 1-level method a number of simulations have been carried on various combinations of FE and VOF meshes. In all cases each finite element is divided into equally many VOF cells in each direction, i.e. $N_1 = N_2 = N_3 = N_v$. The combinations of meshes used are given in Table 3.5. The numbers in the table are used for identifying the various meshes while minuses indicate combinations which have not been used. It is noted that the 1-level meshes (mesh 1, 5, 9 and 12 in Table 3.5) are the same as those in Table 3.3. By comparing results obtained on 2-level meshes with the corresponding 1-level mesh it is possible to get a measure of the accuracy of the 2-level method. In all simulations the size of the fine mesh region is the same as in the previous tests and the initial drop radius is also the same, i.e. $r_0 = 0.25$. For the finest FE-mesh with $N_v = 2$ the variable time step is chosen such that CFL=1/2. The time steps used for the other meshes are then scaled relative to this mesh such that the total number of time steps is approximately the same for all simulations.

The results from the 2-level simulations are presented in 3 different ways. First the interface of the drops after steady state is reached will be investigated on a qualitative level. This is carried out in order to investigate how the subdivision of the FE mesh into a

$h_{fine,fem}$	$N_v = 1$	$N_v = 2$	$N_v = 3$	$N_v = 4$
$r_0/3$	1	2	3	4
$r_0/6$	5	6	7	8
$r_0/9$	9	10	11	-
$r_0/12$	12	13	-	-

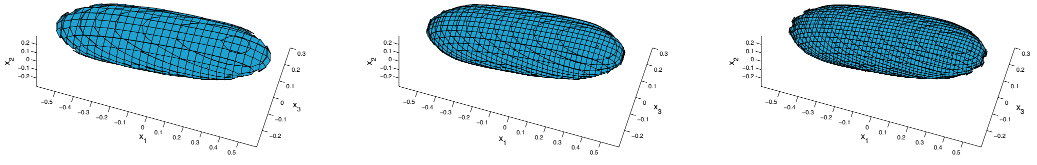
Table 3.5: Combinations of FE and VOF meshes used for investigating the 2-level method. N_v is the number of VOF cells per finite element in each direction. The numbers in the table are those used when referring to a given FE-VOF mesh combination. A minus means that no simulation has been carried out.

finer VOF mesh influences the interface quality compared to the 1-level method. Then the convergence of the transient drop deformation behavior is investigated in terms of VOF mesh resolution. Finally we will take a look at the steady-state deformation parameter D_s for the various mesh combinations tested. Here particular emphasis is on whether or not results obtained on a 2-level mesh are comparable to those obtained on a 1-level mesh with the same VOF mesh resolution.

In Figure 3.9 (a)-(c) the drop shape at steady state can be seen for the simulations carried out using mesh 2-4 in Table 3.5. From the previous results on the 1-level method it was observed that the drop is not stable when mesh 1 is used. This means that the drop is stabilized by the increased resolution of the VOF mesh (i.e. the deformation is limited). By inspection of Figure 3.9 one sees that the interface is sharply defined in all 3 cases, however, as the VOF mesh gets increasingly refined the smoothness of the interface at the drop ends seems to decrease. This is especially evident in Figure 3.9 (c) where each finite element is subdivided into 64 VOF cells. In this case the ends have a somewhat 'wobbly' structure. In Figure 3.10 the steady state results from mesh 5, 6 and 8 can be seen. Here the VOF mesh used for generating the result in Figure 3.10 (a) (1-level) is the same as that used for generating Figure 3.9 (a) (2-level with $N_v = 2$). Comparison between these two figures reveals that the 1-level method leads to a steady state drop with a smoother interface than the 2-level method. However, the agreement between the simulations is acceptable. Figure 3.10 (b) and Figure 3.9 (c) are also based on the same VOF mesh but in the former a 2-level mesh with $N_v = 2$ is used while the latter is based on $N_v = 4$. In this case the interface is considerably smoother when $N_v = 2$ compared to $N_v = 4$. Especially the 'wobbly' ends in Figure 3.9 (c) are much less pronounced in Figure 3.10 (b). The drop interface seen in Figure 3.10 (c), where $N_v = 4$, is quite smooth with much less pronounced end 'wobbling' compared to the coarser 2-level mesh with $N_v = 4$ in Figure 3.9 (c).

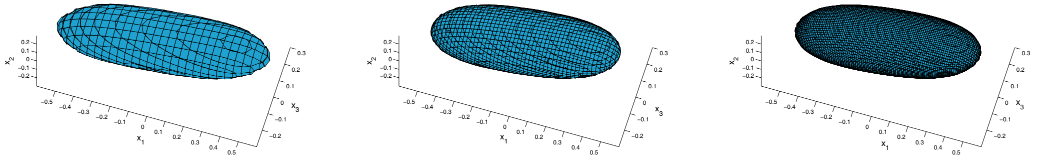
In Figure 3.11 the result obtained from mesh 9-11 can be seen. Here the VOF mesh used in Figure 3.11 (a) (1-level) is the same as the one used in Figure 3.9 (b). Again the agreement is acceptable but with the 1-level mesh leading to a smoother interface. In Figure 3.11 (b) and (c) the drop interfaces are very smooth.

The results from the simulations carried out on the finest calculational mesh (mesh 12 and 13) can be seen in Figure 3.12 (a) and (b). Here direct comparison can be made



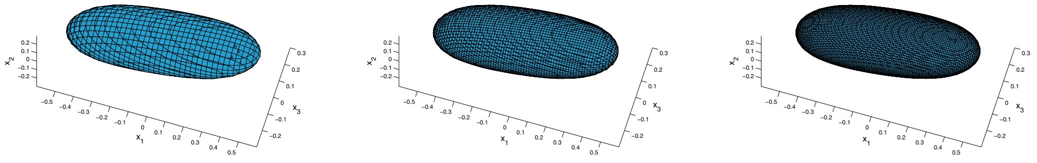
(a) (b) (c)

Figure 3.9: Drop shape after steady state is reached (dimensionless time $t^* \approx 70$) for mesh 2-4, (a)-(b). Shown is the PLIC reconstruction.



(a) (b) (c)

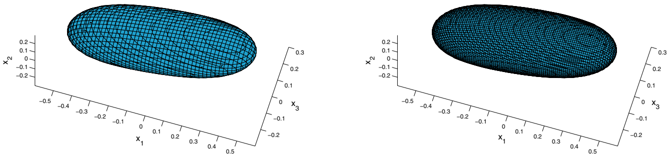
Figure 3.10: Drop shape after steady state is reached (dimensionless time $t^* \approx 70$) for mesh 5,6 and 8, (a)-(b). Shown is the PLIC reconstruction.



(a) (b) (c)

Figure 3.11: Drop shape after steady state is reached (dimensionless time $t^* \approx 70$) for mesh 9-11, (a)-(b). Shown is the PLIC reconstruction.

between Figure 3.12 (a), Figure 3.10 (b) and Figure 3.9 (c). Comparison between the 1-level result in Figure 3.12 (a) and the 2-level result in Figure 3.10 (b) ($N_v = 2$) shows good agreement. However, differences in the smoothness of the interface become apparent if one zooms in on drop end regions of the figures (not shown). Figure 3.12 (b) which is based on the 2-level method with $N_v = 2$ shows a very smooth interface. This result can be compared to Figure 3.10 (c) based on the 2-level simulation with $N_v = 4$. The agreement between the results obtained on these two 2-level meshes is good.



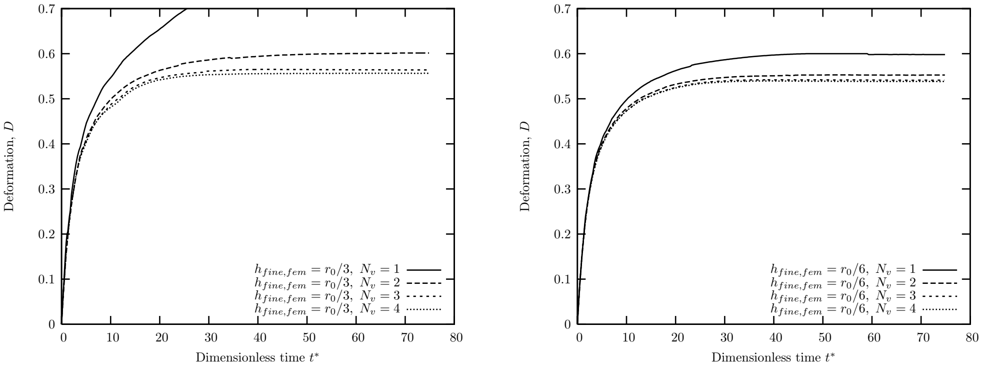
(a) (b)

Figure 3.12: Drop shape after steady state is reached (dimensionless time $t^* \approx 70$) for mesh 12-13, (a)-(b). Shown is the PLIC reconstruction.

From these qualitative observations of steady state drop shapes obtained on the various FE-VOF mesh combinations it is evident that increasing the resolution of the VOF mesh while keeping the FE mesh constant stabilizes the drop on the coarsest FE mesh. However, when the 2-level method is used together with the coarsest FE mesh the interface is not very smooth especially when the number of VOF cells per finite element is increased. This problem is considerably reduced when finer FE meshes are used.

The results from mesh 1-4 and 5-8 are used for investigating how the transient drop deformation behaves as the VOF mesh is refined on a given FE mesh. In Figure 3.13 (a) the drop deformations obtained on mesh 1-4 can be seen (here the coarsest FE mesh is used). This figure indicates that the transient deformation curves converge as the VOF mesh is refined. In Figure 3.13 (b) the results from mesh 5-8 can be seen. Here the FE mesh is twice as fine as that in figure (a). Again we see a convergence of the deformation curves. Comparing the results in Figure 3.13 (a) and (b) with the results from the 1-level method in Figure 3.8 shows that the general behavior of the deformation curves with VOF mesh refinement is consistent. Furthermore the steady-state deformation parameter D_s for a given VOF mesh does not seem to be very dependent on the FE mesh used.

In Figure 3.14, D_s obtained at the end of each simulation can be seen as a function r_0/h_{vof} where h_{vof} is the VOF cell side lengths (the deformation parameters are also given in table 3.6). Furthermore the degree of finite element sub-division (i.e. N_v) is indicated for each data point. From the figure it is seen that for constant VOF mesh resolution, increasing the degree of sub-division (i.e. N_v) leads to a larger D_s when $r_0/h_{vof} < 18$. On the other hand the data points for $r_0/h_{vof} \geq 18$ show very little dependence of D_s on N_v. It is noted that the 2-level data points at $r_0/h_{vof} = 6$, 9 and 12 with $N_v = 2$, 3 and 4 are based on the coarsest FE mesh. From the qualitative analysis of the interface smoothness above it was observed that these simulations resulted in a somewhat 'wobbly' interface at the drop ends. This of course will have an influence on the accuracy of the interfacial tension calculations and might explain a part of the deviation observed between the 2-level and 1-level results (circles in Figure 3.14). Hence one can argue that the accuracy in the finite element discretization is not sufficient when the coarsest mesh is used. Nevertheless the results in Figure 3.14 give a clear indication that one can improve the performance of the program by using the 2-level algorithm and that D_s converges as the VOF mesh is refined. From Table 3.6 the limiting deformation parameter is seen to be around 0.54 when the VOF resolution is in the converged region which is obtained for $r_0/h_{vof} \geq 18$. In these numerical experiments only one type of flow field and set of parameters are used. Therefore the analysis carried out here is somewhat restricted. However, our result $D_s \approx 0.54$ agrees very well with the results of Li et al. (2000) where $D_s = 0.541$ was obtained with the same parameters (though not the same mesh resolution). Boundary integral calculations by Rallison (1984) and by Kwak & Pozrikidis (1998) yield a limiting deformation parameter of 0.46 and 0.5 respectively and thus the discrepancy between our result and their result is somewhat larger.



(a) (b)

Figure 3.13: Transient drop deformations for (a) mesh 1-4 in Table 3.5 and (b) mesh 5-8.

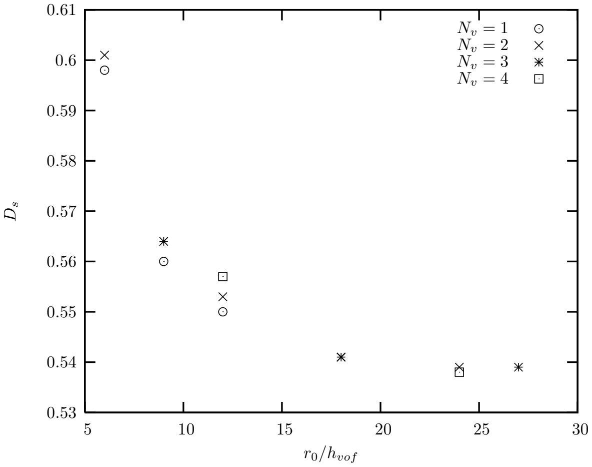


Figure 3.14: The steady-state deformation parameter D_s obtained using the meshes in Table 3.5 as a function of the VOF cell side lengths h_{vof} (in the fine mesh region).

$h_{fine,fem}$	$N_v = 1$	$N_v = 2$	$N_v = 3$	$N_v = 4$
$r_0/3$	unstable	0.601	0.564	0.556
$r_0/6$	0.598	0.553	0.541	0.538
$r_0/9$	0.560	0.541	0.539	-
$r_0/12$	0.550	0539	-	-

Table 3.6: Steady-state deformation parameters D_s obtained on the various combinations of FE and VOF meshes used for testing the 2-level algorithm.

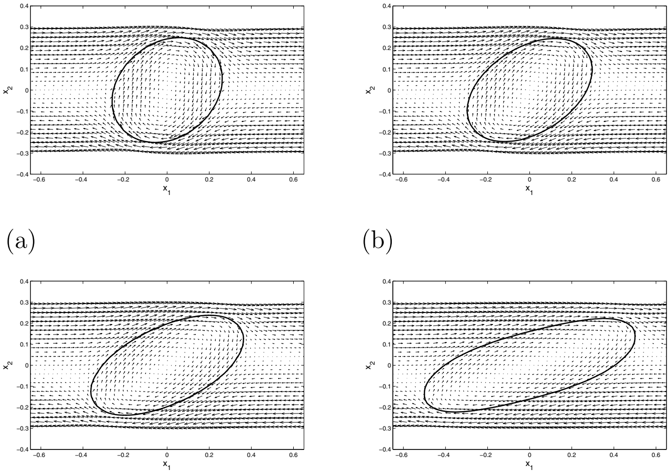
3.3 Simulations in simple shear flow

In order to test the FE-VOF model further a number of simulations have been carried out in simple shear with various values of the capillary number and viscosity ratio. These simulations include drop break-up, merging of 2 drops and near collision of 2 drops. However, we will begin with results obtained from simulations with $\lambda = 1$ and Ca varying from 0.1 up to just above the critical value. The results from our simulations are then compared to results reported in the literature.

3.3.1 Simple shear, $\lambda = 1$

All simulations carried out here are based on a finite element mesh with $h_{fine} = r_0/6$ and the 2-level method with $N_v = 3$ (i.e. $h_{vof} = r_0/18$) - unless otherwise stated. This corresponds to the converged region of Figure 3.14. Furthermore the time-step size is based on the setting CFL=1/2. The dimensions of the fine mesh domain depend on how much the drop is expected to deform, but in all simulations the coarse mesh region consists of 4 elements spanning from the fine region to the domain boundaries. The dimension of the calculational domain is $(L_1, L_2, L_3) = (24r_0, 24r_0, 24r_0)$.

In Figure 3.15 (a)-(d) the steady velocity field and drop shape on the (x_1, x_2)-symmetry plane can be seen for simulations with Ca = 0.1, 0.2, 0.3 and 0.4. This figure shows how the interfacial tension perturbs the applied shear flow and results in a rotating flow field in the interiour of the drop. The dimensions of the fine region is $[5r_0, -5r_0] \times [-7/6r_0, 7/6r_0] \times [-7/6r_0, 0]$ in all simulations. Besides the obvious increase in the degree of deformation with increasing Ca number, Figure 3.15 (a)-(d) also shows that the angle between the major drop axis and the $x_2 = 0$ plane decreases with increasing capillary number as expected by theory, e.g. (Cox 1969). One can compare Figure 3.15 (a)-(d) with Figure 5 in (Li et al. 2000) where the same types of plots are given. From a qualitative inspection our results are in good agreement with the results of Li et al. (2000). In order to make a more quantitative comparison the steady state deformation parameter D_s and the steady state angle of the drop relative to the x_1-axis are presented in Figure 3.16 (a) and (b) together with literature data. It is noted that similar figures are presented in (Kwak & Pozrikidis 1998) and (Li et al. 2000). For small capillary numbers ($Ca < 0.2$) the steady state deformation data in Figure 3.16 (a) agree well with the small deformation asymptote. However as the capillary number increases beyond 0.2 the spread in the values of D_s increases. Especially one sees that both BI methods lead to somewhat smaller deformations than the VOF methods. On the other hand it is seen that our 2-level VOF method is in excellent agreement with the VOF calculations of Li et al. (2000) up to $Ca = 0.4$. It is noted that because the mesh used in our simulations corresponds to the converged region in Figure 3.14 increasing the resolution does not change the final result significantly. From this we can conclude that our steady deformation results agree with the VOF calculations of Li et al. (2000) for all capillary numbers up to 0.4 while the agreement with BI methods is somewhat worse for high capillary numbers. Although only three data points are present, the experimental results of Rumscheidt & Mason (1961), also plotted in Figure 3.16 (a), can be seen to deviate from the small deformation limit of Cox (1969) with increasing Ca in the same manner as the numerical calculations. In Figure 3.16 (b) the drop angle at steady state can be seen for the same data points as in (a). The figure shows $45^\circ - \theta$ as a function of Ca where θ is the angle between the drop major axis and the x_1 coordinate axis. From the figure some spread in the data is seen for all capillary numbers. Furthermore the deviation between the small deformation asymptote and the data points is considerable for all values of Ca. It is though noted that for small deformations it is difficult to obtain accurate values of the drop angle as the major axis of the drop is difficult to identify precisely. This is, however, not a problem for large deformations (Ca>0.1). The results on the steady state angle from our VOF calculations are in good agreement with those of Li et al. (2000) for $Ca > 0.1$. The discrepancy at $Ca = 0.1$ is probably due to inaccuracy in determining the major drop axis.

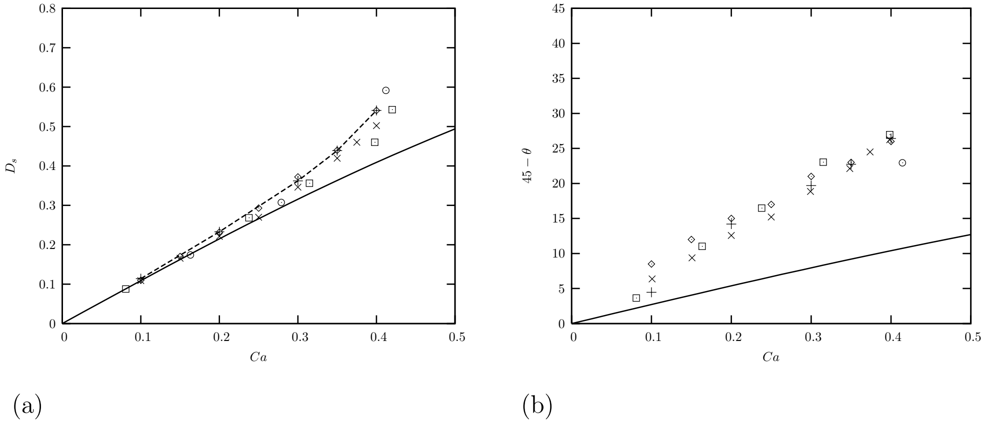


(a) (b)

(c) (d)

Figure 3.15: Velocity vector plots and drop shapes at steady state on the (x_1, x_2)-symmetry plane for simulations with (a) $Ca = 0.1$, (b) $Ca = 0.2$, $Ca = 0.3$ and $Ca = 0.4$. $\lambda = 1$ and $N_v = 3$ was used in all simulations.

In Table 3.7 some values of the critical capillary number reported in the numerical literature for $\lambda = 1$ are given. Here it is seen that the there is some spread in the literature data, however, the 'true' critical capillary number is usually taken to be around 0.42-0.43, e.g. (Kwak & Pozrikidis 1998). As shown by Renardy, Cristini & Li (2002) (VOF) the resolution of the mesh has a large impact on whether or not a drop will break up for capillary numbers in the vicinity of Ca_{crit}. They have carried out simulations with $Ca = 0.44$ and various mesh resolutions and show that for a mesh with element side lengths



(a)

(b)

Figure 3.16: Drop deformation in simple shear with $\lambda = 1$ and varying capillary number. Comparison of the steady drop deformation parameter (a) and steady drop angle (b) with values reported in the literature. $+$ 2-level VOF with $N_v = 3$ (also indicated with a broken curve in (a)); $\diamond$ VOF calculations of Li et al. (2000); $\times$ BI calculations of Kwak & Pozrikidis (1998); $\boxdot$ BI calculations of Rallison (1984); $\circ$ experiments by Rumscheidt & Mason (1961); solid curve is the asymptotic result by Cox (1969) for small deformation.

Numerical method	Ca_{crit}	Reference
BI	0.42	(Rallison 1984)
BI	0.37	(Kennedy et al. 1994)
BI	0.42	(Kwak & Pozrikidis 1998)
BI	0.43	(Cristini et al. 2003)
VOF	0.42	(Li et al. 2000)

Table 3.7: Values of the critical capillary number for $\lambda = 1$ reported in the literature.

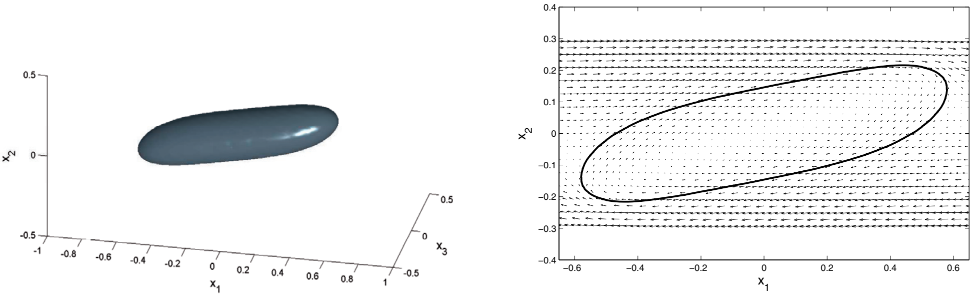
$h = r_0/8$ the drop is stable whereas for $h = 3r_0/32$ the drop breaks-up. Furthermore they show that refining the mesh further results in a faster break-up process. Results from BI calculations are also quite sensitive to the resolution of the boundary mesh as noted in (Cristini et al. 2003). Thus the variation among the reported critical capillary numbers can probably be explained, at least partially, by differences in mesh resolutions used in the simulations.

It is noted that from the Taylor small deformation theory the predicted critical capillary number is 0.457 when $\lambda = 1$, cf. (Taylor 1932).

In order to investigate how our code handles drop break-up and what the predicted critical capillary number is, simulations have been carried out for $Ca = 0.42$, 0.44 and 0.46. In these simulations the dimension of the fine mesh region is $[-8r_0, 8r_0] \times [-7/6r_0, 7/6r_0] \times [7/6r_0, 0])$. Since we now operate in the critical region the Taylor deformation parameter is not useful for quantifying the deformation and instead the half length of the drop relative to the initial radius is used as a measure of the deformation. The shear rate is $\dot{\gamma} = 1\text{s}^{-1}$ in all simulations. For transient events the time is made dimensionless using the shear rate, i.e. $t^+ = t\dot{\gamma}$, and not the characterisitc drop time as earlier which is more suitable for small deformations.

The simulation carried out with $Ca = 0.42$ did not lead to drop break-up. The final drop shape can be seen in in Figure 3.17. It is noted that when plotting 3D drop shapes the iso-surface based on the 0.5 contour of the F-field will be used from now on instead of the PLIC faces used earlier. This is done in order to enhance the 3D view of the drop.

The evolution of the drop shape for the simulation carried out with $Ca = 0.44$ can be seen in Figure 3.18 (a) through (f). Here it is seen that as the drop deforms a neck starts to form as evident in Figure 3.18 (c) ($t^+ = 50$). As time proceeds this neck increases which leads to a thread connecting the two major drops. Figure 3.18 (e) shows the drop shape just before the main drops pinch off the thread. After pinch-off the thread starts to retract which leads to a single drop in the centre and some debris between the central drop and the two main drops. It is not possible, given the resolution of the VOF mesh, to determine if the debris formed is an artifact stemming from the numerical method or if it corresponds to sub-cell sized droplets. The final drop size distribution can be seen in Figure 3.18 (f). Except from the debris this behavior is typically observed in simple shear when $\lambda = 1$ and the capillary number is close to the critical value, see e.g. (Li et al. 2000, Renardy, Cristini & Li 2002).



(a) (b)

Figure 3.17: Steady state drop shape in simple shear with $Ca = 0.42$ and $\lambda = 1$. (a) 3D drop shape and (b) flow field and drop shape on the symmetry plane.

In Figure 3.19 (a) through (d) the drop shape at times $t^+ = 10$, 30, 50 and 60 is shown for the simulation with $Ca = 0.46$. Here it is seen that the break-up process proceeds considerably faster compared to $Ca = 0.44$ (cf. Figure 3.18). Furthermore the drop now becomes more extended before pinch-off of the main drops, see Figure 3.19 (c). At $t^+ = 60$ the break-up process has completed leading to the final drop distribution shown in Figure 3.19 (d). Thus for $Ca = 0.46$ we obtain 5 drops which can be resolved by the VOF mesh, i.e. two main drops, a smaller central drop with two satellite droplets on each side. Hence increasing the Ca-number from 0.44 to 0.46 leads to different drop size distribution.

In Figure 3.20 the evolution of the half drop length L/r_0 with dimensionless time can be seen for the simulations shown in Figure 3.17, 3.18 and 3.19. From Figure 3.20 it is seen that initially ($t^+ < 2$) the rate of deformation is the same for all three simulations. For $t^+ > 2$ the deformation curves start to diverge with $Ca = 0.42$ leading to a stationary drop shape while $Ca = 0.44$ and $Ca = 0.46$ lead to drop break-up. It is also seen that the deformation is faster for $Ca = 0.46$ compared to $Ca = 0.44$ which as discussed above leads to a more extended drop when pinch off occurs.

In order to investigate if the size of the fine mesh region and the BC's applied at the end walls ($x_1 = -L_1$ and $x_1 = L_1$) have any influence on the critical capillary number two more simulations have been carried out with $Ca = 0.42$. In one simulation the size of the fine region is increased in the x_2 and x_3 direction such that the total size is $[-8r_0, 8r_0] \times [-10/6r_0, 10/6r_0] \times [-10/6r_0, 0]$. In the second simulation periodic BC's are imposed on the end walls. The dimensionless drop length L/r_0 as a function of the dimensionless time t^+ for these simulations can be seen in Figure 3.21. Also shown in this figure is the result from Figure 3.20 for $Ca = 0.42$. The figure shows that increasing the size of the fine mesh region or changing the end wall BC's to periodic does not result in any significant change in the drop shape evolution history. This indicates that the critical capillary number in our case is larger than 0.42 (with $\lambda = 1$). In fact the slow drop shape evolution obtained

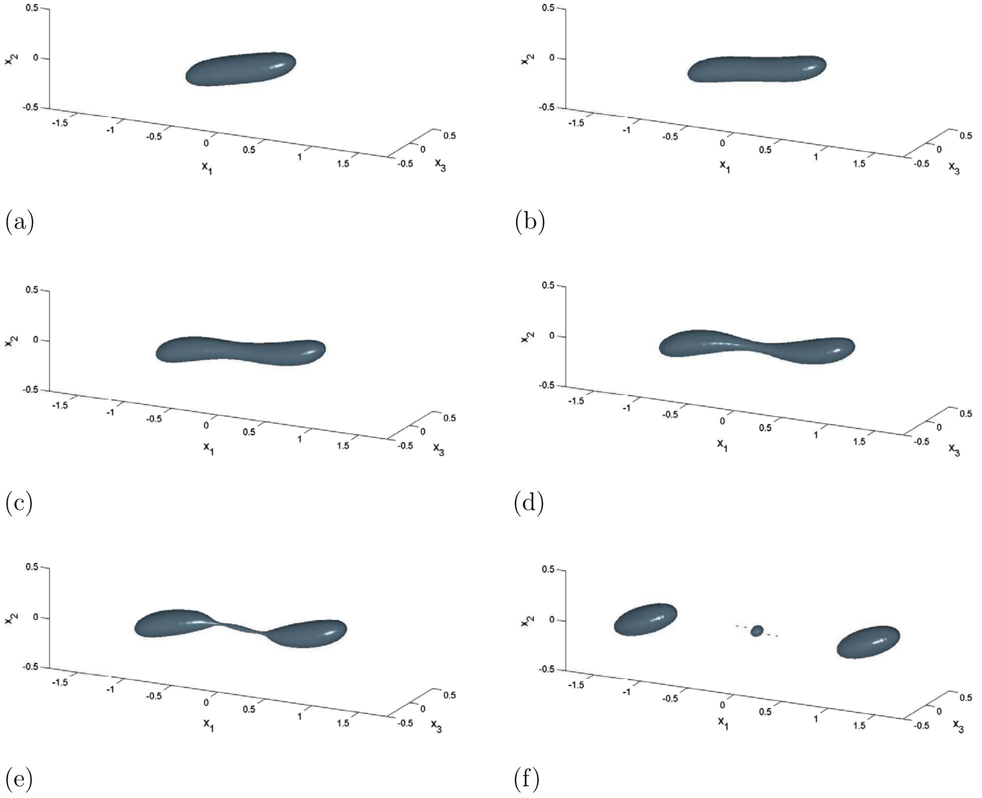
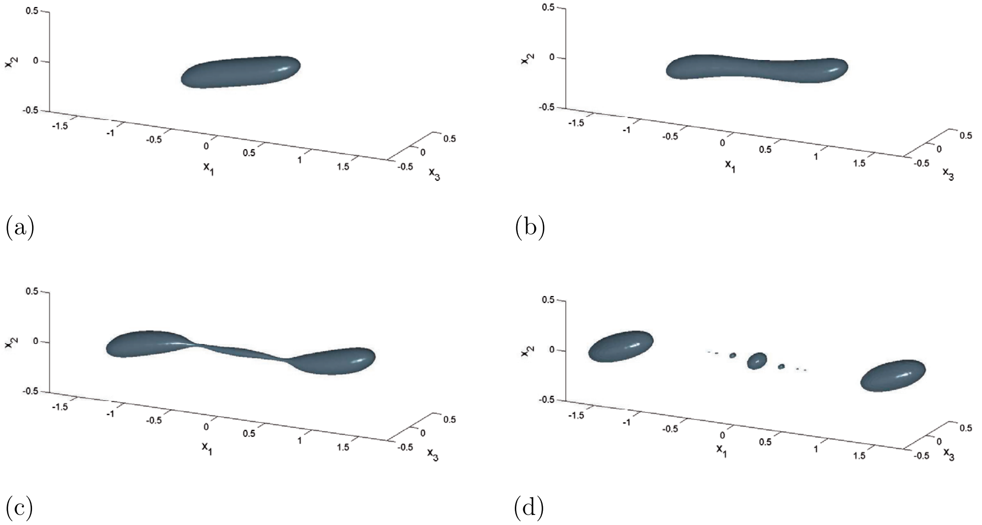


Figure 3.18: Simple shear simulation with $Ca = 0.44$ and $\lambda = 1$. Figures (a) to (f) show the drop at dimensionless times $t^* = 10$, 30, 50, 60, 65 and 76.



(a)

(b)

(c)

(d)

Figure 3.19: Simple shear simulation with $Ca = 0.46$ and $\lambda = 1$. Figures (a) to (d) show the drop at dimensionless times $t^+ = 10$, 30, 50 and 60.

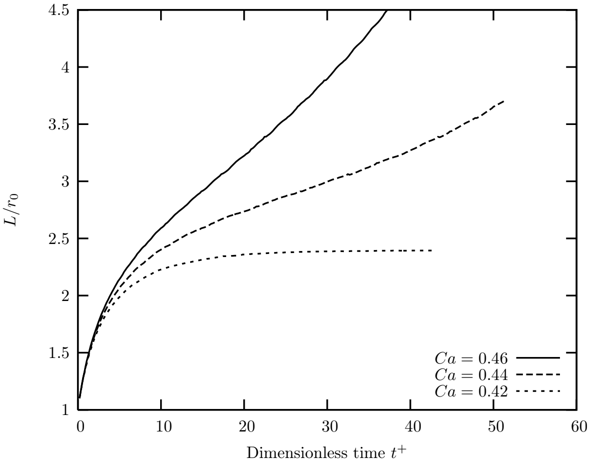


Figure 3.20: Evolution of the drop half length L/r_0 with dimensionless time t^+ for simulations with $Ca = 0.42$, 0.44 and 0.46 and $\lambda = 1$.

for $Ca = 0.44$ suggests that $Ca_{crit} \approx 0.44$ in our case.

Since the computations become considerably more time consuming when the viscosity ratio is different from one (due to factorization at each time step) the tests carried out for λ different from one are somewhat less extensive. Here we will start by presenting results from simulations with $\lambda = 0.1$.

3.3.2 Simple shear $\lambda = 0.1$

In Figure 3.22 (a) the steady state drop shape and velocity field on the symmetry plane can be seen for a simulation with $Ca = 0.40$ and $\lambda = 0.1$. The resolution of the finite element mesh and the VOF mesh used in this simulation is the same as in the previous section, i.e. $h_{fine} = r_0/6$ and $N_v = 3$ ($h_{vof} = r_0/18$). The obtained Taylor deformation parameter at steady state is $D_s = 0.51$. Comparison of the steady state drop shape in Figure 3.22 (a) with that obtained for $\lambda = 1$ in Figure 3.15 (d) reveals that the drop ends become somewhat more pointed in the case where $\lambda = 0.1$. Furthermore it is evident that the rotational motion of the flow inside the drop is stronger when $\lambda = 0.1$. This is expected because the drop is much less viscous than the continuous phase when $\lambda = 0.1$. Comparison of our result with results in (Li et al. 2000) (Figure 10) shows good qualitative agreement with regards to flow field behavior and drop shape. However, Li et al. (2000) report $D_s = 0.44$ which is considerably lower than our result. In order to investigate how the resolution of the finite element mesh influences the results when λ is different from one, the simulation shown in Figure 3.22 (a) has been carried out on a finite element mesh with a higher resolution. In this simulation the side lengths of the finite elements in the fine

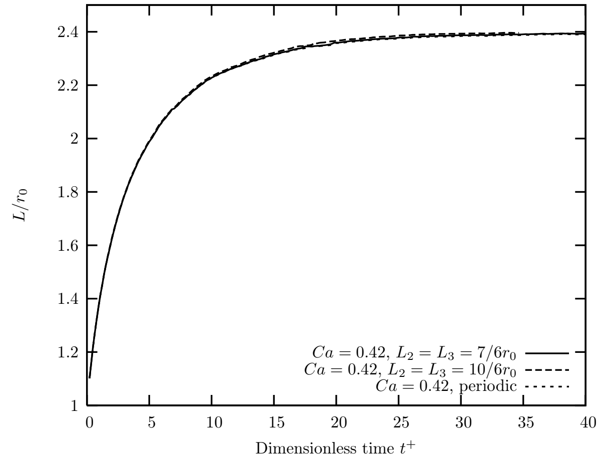


Figure 3.21: Evolution of the drop half length L/r_0 with dimensionless time t^+ for simulations with $Ca = 0.42$ and $\lambda = 1$. The curves correspond to simulations carried out with the tight fine mesh region $(L_{fine,1}, L_{fine,2}, L_{fine,3}) = (8r_0, 7/6r_0, 7/6r_0)$, the 'loose' fine mesh region $(L_{fine,1}, L_{fine,2}, L_{fine,3}) = (8r_0, 10/6r_0, 10/6r_0)$ and periodic BC's on the end walls.

mesh region were decreased to $r_0/8$. The 2-level method with $N_v = 3$ was again used and thus the resolution of the VOF mesh is also increased compared to the simulation in Figure 3.22 (a). In Figure 3.22 (b) the drop shape obtained on the finer mesh is compared to the drop shape obtained on the coarser mesh. Here it is clear that the finer mesh leads to a less deformed drop. The steady state deformation parameter on the fine mesh is $D_s = 0.48$ which is closer to the results of Li et al. (2000). These results thus indicate that when λ is different from one the resolution of the finite element mesh needs to be higher compared to when $\lambda = 1$. Due to time limitations a full convergence test has not been carried out for λ different from 1. It is recalled that for interface finite elements the viscosity (and density) are calculated simply from average values of the F-field. In order to increase the accuracy of the calculations one could instead use a procedure similar to the one used for the 2-level interfacial tension calculations (section 2.5.2.2). Thus for each finite element interface cell, the momentum balance terms should be based on a summation of VOF cell contributions. Although this procedure would increase the necessary computation time for setting up the coefficient matrix, the reduction in necessary factorization and back substitution time could reduce the total computational cost.

In his experimental work Grace (1982) found that for simple shear the critical capillary number is at its minimum somewhere around $\lambda = 0.6$. Furthermore his data suggest that the critical capillary number at $\lambda = 0.1$ is approximately the same as when $\lambda = 1$. The BI simulations of (Cristini et al. 2003) show a minimum around $\lambda = 0.5$, however, in their simulations the critical capillary number for $\lambda = 0.1$ is $Ca_{crit} \approx 0.53$ which is somewhat

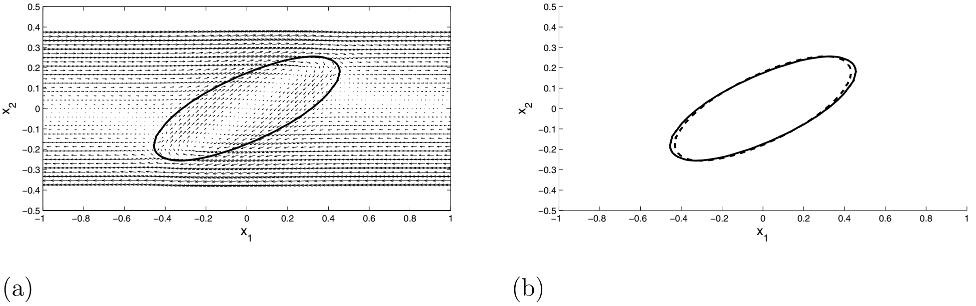


Figure 3.22: Steady state drop shape for $Ca = 0.40$ and $\lambda = 0.1$. (a) cross-sectional slice at the (x_1, x_2)-symmetry plane showing interface location and velocity field at steady state when the coarse finite element mesh is used. (b) Comparison of steady state drop shape when the coarse finite element mesh (full curve) and fine finite element mesh (broken curve) are used.

higher than for $\lambda = 1$ ($Ca_{crit} \approx 0.43$).

In order to investigate how reducing λ from 1 to 0.1 affects the drop deformation when $Ca \approx Ca_{crit}$ a simulation has been carried out with $\lambda = 0.1$ and $Ca = 0.46$. In this simulation the fine mesh from the $Ca = 0.4$ simulation above is used. The time step size is determined from setting CFL=1/2. The results from this simulation can be seen in Figure 3.23 which shows that the drop does break up. Figure 3.23 (a) which shows the drop shape just before break-up indicates that the drop is elongated less before break-up when $\lambda = 0.1$ compared to $\lambda = 1$, cf. Figure 3.19. However, the final drop size distribution for $\lambda = 0.1$ resembles the one obtained for $\lambda = 1$, i.e. two major drops with a smaller central drop with satellite drops on each side.

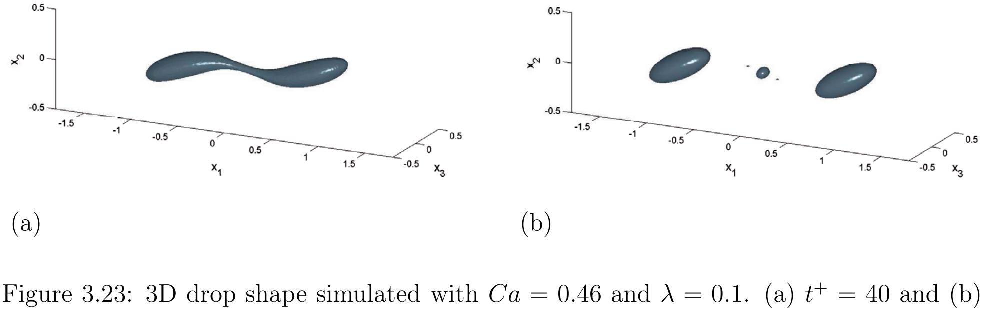


Figure 3.23: 3D drop shape simulated with $Ca = 0.46$ and $\lambda = 0.1$. (a) $t^+ = 40$ and (b) $t^+ = 47$.

In Figure 3.24 the drop shape on the (x_1, x_2)-symmetry plane, just before break-up, is compared for $\lambda = 0.1$ and $\lambda = 1$. Here the differences in elongation is very clear. It is, however, noted that the mesh used in the $\lambda = 0.1$ simulation has a higher resolution than the mesh used for $\lambda = 1$. The differences in the drop shape can therefore be partly due to mesh resolution differences. Nevertheless the simulation with $\lambda = 0.1$ shows that the model is able to handle drop break-up for viscosity ratios smaller than 1.

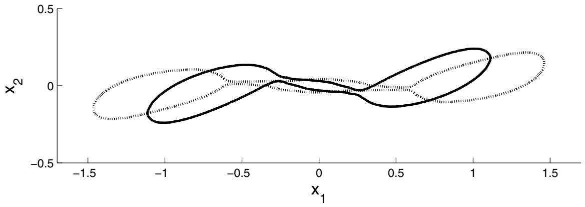


Figure 3.24: Comparison of drop shape just before break-up in simple shear with $Ca = 0.46$. Broken curve shows the drop shape when $\lambda = 1$ while the full curve is obtained for $\lambda = 0.1$.

3.3.3 Simple shear $\lambda = 25$

In order to test the model for $\lambda > 1$ a simulation is carried out with $Ca = 1.52$ and $\lambda = 25$. These parameters correspond to one of the datasets from the experimental series carried out by Torza et al. (1972). Furthermore this dataset has also been used by Rallison (1980) in connection with validation of theoretical work. It is noted that for simple shear flow and $\lambda > 4$ it is not possible to induce drop break-up regardless of the flow field strength, e.g. (Grace 1982). The transient deformation behavior for reaching the steady state is on the other hand more complicated than for small values of λ. The high viscosity of the drop results in an oscillating behavior of the drop angle and the deformation parameter. The frequency, amplitude and damping depend on the viscosity ratio and capillary number. Theoretical expressions for the drop angle and deformation as a function of time for large viscosity ratios are e.g. given by Cox (1969) and Rallison (1980). These are based on the small deformation assumption. Here we will compare the results from our simulations with the experimental data by Torza et al. (1972) and the theoretical prediction by Rallison (1980). In the VOF simulation the resolution of the fine finite element mesh region corresponds to the fine mesh used for the $\lambda = 0.1$ simulation. Again the 2-level procedure with $N_v = 3$ is applied for the VOF mesh.

In Figure 3.25 the results from the simulation can be seen. Furthermore the theoretical results of Rallison (1980) and the experimental results of Torza et al. (1972) are also shown. In the top figure the angle between the drop major axis and the x_2-coordinate axis can be seen while the bottom figure shows the Taylor deformation parameter. The results in Figure 3.25 show that the experimental results lead to larger amplitude and smaller frequency in the deformation and angle oscillations compared to the VOF simulation and the theoretical prediction. However, comparison between our VOF simulation and the small deformation prediction is quite good both with respect to amplitude and frequency. It is though noted that the oscillation frequency is slightly higher in the VOF simulation. Better agreement might be obtained if a finer finite element mesh is used. Nevertheless the agreement between the simulated transient drop shape behavior and that predicted by theory is satisfactory.

3.3.4 Collision of two drops in simple shear flow

An extensive numerical study of droplets en route to collision in simple shear flow has been carried out by Loewenberg & Hinch (1997). In this study the near contact region of two drops flowing past each other is investigated in terms of the relationship between drop distance and the capillary number and viscosity ratio. However, none of their numerical simulations results in coalescence of the two drops. Here we will show the results from two simulations where two drops of equal size approach each other in simple shear. In the first simulation (simulation 1) the drop interfaces do not come in contact with each other while in the second simulation (simulation 2) the drop merge to form a stable larger drop. The motivation for the simulations is to test how the model handles drop-drop interactions and drop merging. In Figure 3.26 (a) the initial position of the two drops is sketched for simulation 1. The drops are initially separated by $5r_0$ and the drop centres are offset from the x_1 axis by the distance $1/2r_0$ and $-1/2r_0$ respectively. Figure 3.26 (b) shows the initial drop positions for simulation 2. Here the initial drop distance is $3r_0$, and the offset from the x_1 axis is $1/3r_0$ and $-1/3r_0$. Thus in simulation 2 the drops have to travel further in the vertical (x_2) direction within in a shorter distance in the horizontal (x_1) direction in order to avoid collision. In both simulations $\lambda = 1$ and the capillary number of each drop is $Ca = 0.3$. The finite element side lengths are $h_{fine} = r_0/6$ and the 2-level method with $N_v = 3$ is used. The time step size is determined from the setting CFL=1/2.

Figure 3.27 left shows the 3D drop shapes at times $t^+ = 2$, 4, 5 and 6 for simulation 1. The right figure shows the corresponding flow field and the drop shape on the symmetry plane. From the plots in Figure 3.27 it is seen that initially the interfacial tension induces a circulation zone close to the lower tip of the left drop and upper tip of the right drop (top figure). As the drops approach each other the flow starts to circulate in the region between the drops which results in a 'flattening' of the interfaces in the near contact region (the two middle plots). The result is that the drops slide past each other due to the interface 'flattening' and due to a vertical diffusion of the drops. Hence the drops do not come into contact with each other.

In Figure 3.28 the results are shown for simulation 2. Now the two drops are placed closer to each other initially and the offset between the x_2-axis and the drop centres is smaller. Here we also see circulation zones close to the drop tips and a net flow in the vertical direction. However, the small initial separation between the drops prevents them from sliding past each other. At around $t^+ = 4$ the interfaces of the two drops start to merge. The symmetry plane plot reveals that as the interfaces merge some of the continuous phase liquid gets trapped in the interior of the drops. This is probably a consequence of the VOF method which cannot resolve two interface segments lying within the same VOF

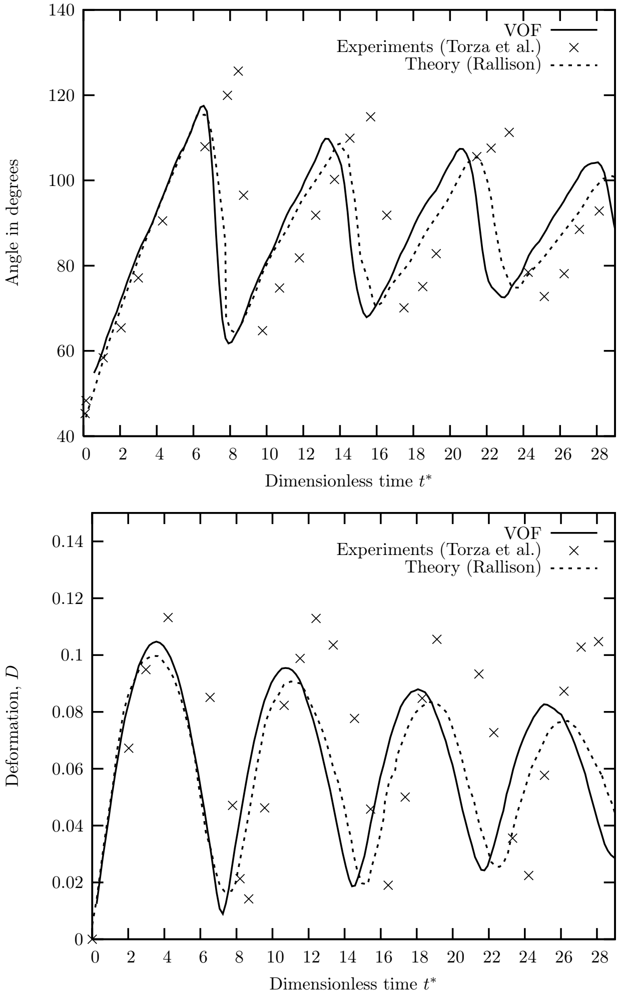


Figure 3.25: The drop angle (with the x_2-coordinate axis) and deformation parameter as a function of time for $\lambda = 25$ and $Ca = 1.52$. Solid curve is our VOF result, broken curve is the theoretical result of Rallison (1980) and × is the experimental result of Torza et al. (1972).

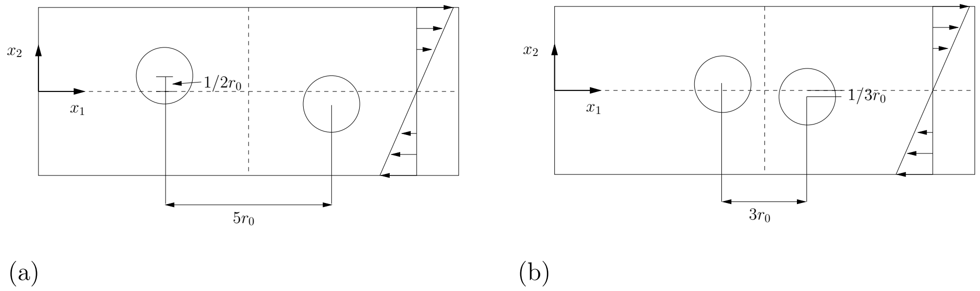
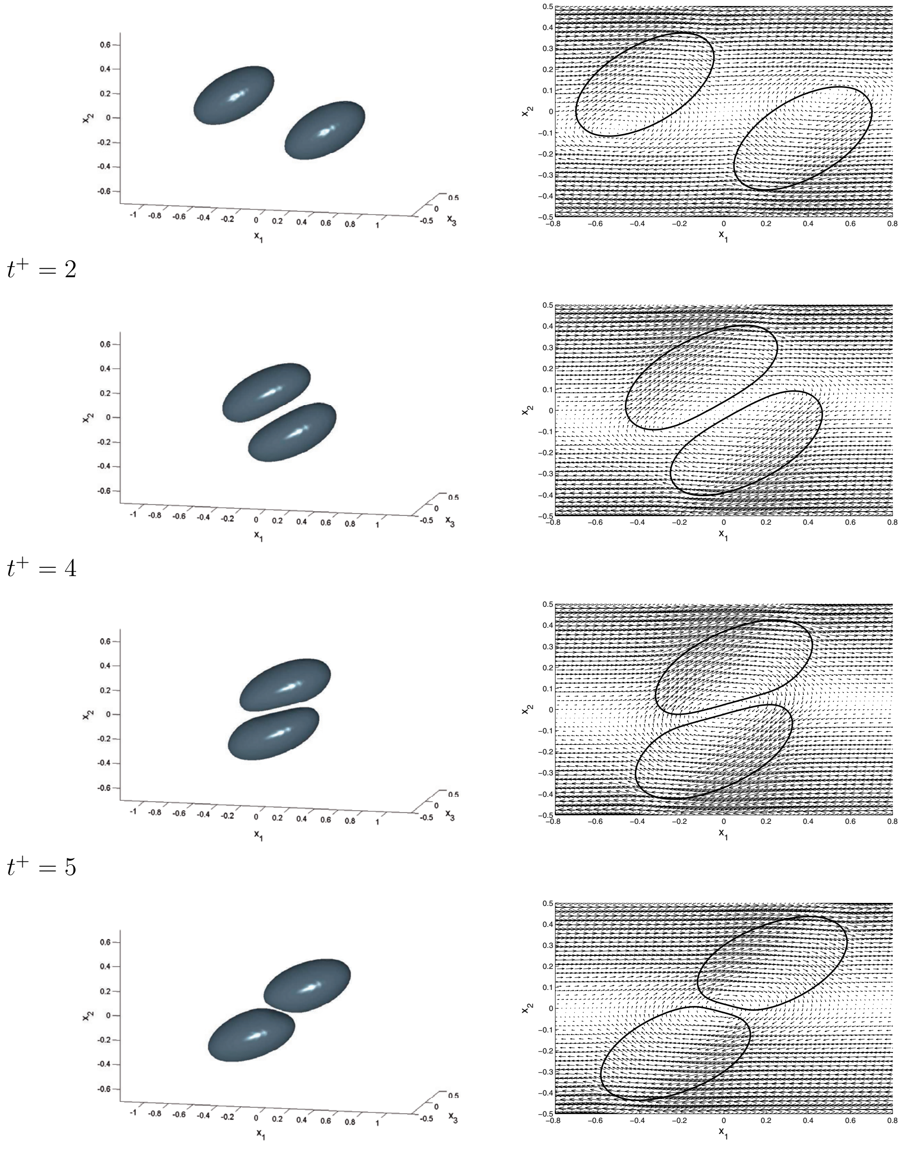


Figure 3.26: Sketch showing the initial position of the two spherical drops. (a) the distance between the centres of the two drops is $5r_0$ while the offset between the drop centres and the x_1 axis is $1/2r_0$ and $-1/2r_0$ respectively. (b) drop distance is $3r_0$ and offset is $1/3r_0$ and $-1/3r_0$ respectively.

element. Thus when the distance between the interfaces is below the VOF cell size they are automatically merged and whatever void volume is left in the VOF cell remains in the interior of the drops. At $t^+ = 4.5$ the two drops are almost fully merged and the flow shows a strong tendency to smooth out the large curvatures seen along the region where the drops are merging. At $t^+ = 6$ the two drops have merged completely into one drop and we see two small drops of the continuous phase within the disperse phase. Although not shown here these two drops are spiralling towards each other and eventually form a single drop in the centre of the disperse phase. It is noted that the total volume of the merged drop is twice that of the initial drop (if we neglect the inclusion of the continuous phase). This means that the capillary number of the final drop is 0.38 which is below the critical value and thus the formed drop is stable in the flow field. The simulations carried out here show that the model is able maintain the integrity of drop interfaces during close drop-drop interactions and during merging of two drops.

3.4 Simulations in planar elongation

In general, flow fields based on planar elongation are more efficient in bursting drops than simple shear flow fields. This is reflected in the critical capillary number vs. viscosity ratio (see the Grace curve in Figure 1.4) where Ca_{crit} is always lower in elongation than in shear. In this section results are presented from simulations carried out in planar elongation flows. The simulations are carried out with Ca-numbers ranging from 0.02 to 0.13 and $\lambda = 1$. The critical capillary number for planar elongation and $\lambda = 1$ is reported to be in the vicinity of 0.11, e.g. (Grace 1982). Thus our simulation with $Ca = 0.13$ is above the critical point and break-up is expected. It is noted that the applied elongation flow is obtained by setting $\alpha = 1$ and $G = 1$ in equation 3.1-1. Thus the applied elongation rate is $\dot{\epsilon} = 1\text{s}^{-1}$. Although



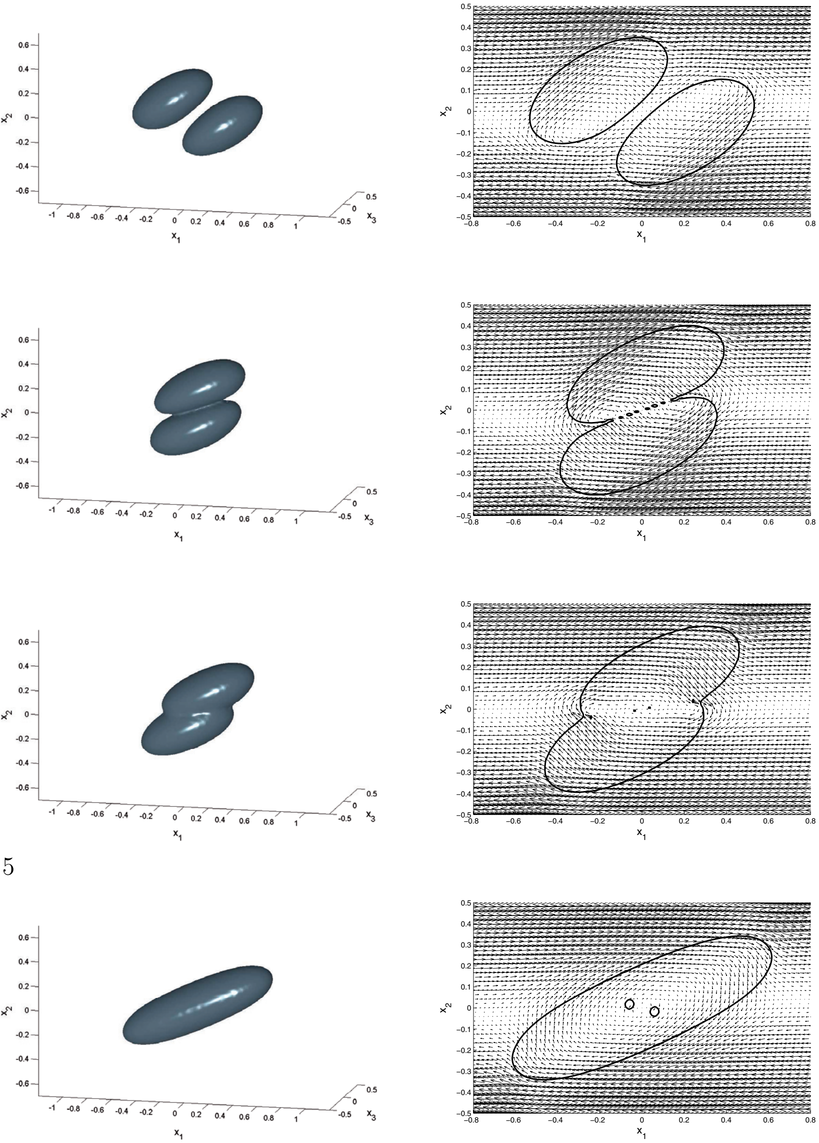
$t^+ = 2$

$t^+ = 4$

$t^+ = 5$

$t^+ = 6$

Figure 3.27: Simulation of two drops approaching and flowing past each other in simple shear. On the left figure the 3D drop shapes are shown while on the right figure the flow field on the x_1, x_2-symmetry plane is shown for dimensionless time $t^+ = 2$, 4, 5 and 6. Initially the drops are spherical and separated by $5r_0$ between their centres. Furthermore the drop centres are off the plane at $x_2 = 0$ by $1/2r_0$ and $-1/2r_0$ respectively.



$t^+ = 2$

$t^+ = 4$

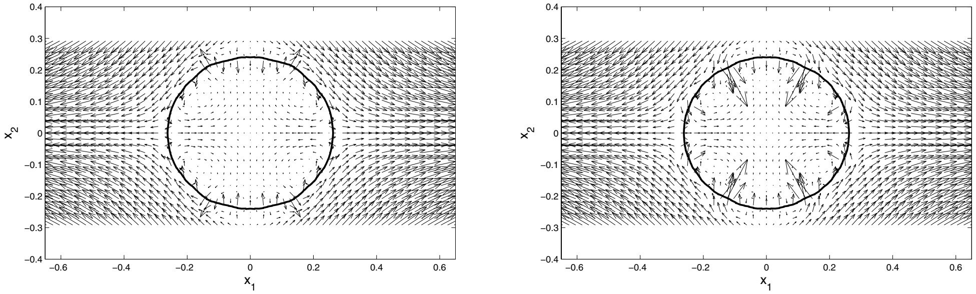
$t^+ = 4.5$

$t^+ = 6$

Figure 3.28: Simulation of two drops approaching and merging with each other in simple shear. On the left figure the 3D drop shapes are shown while on the right figure the flow field on the (x_1, x_2)-symmetry plane is shown for dimensionless time $t^+ = 2$, 4, 4.5 and 6. Initially the drops are spherical and separated by $3r_0$ between their centres. Furthermore the drop centres are off the plane at $x_2 = 0$ by $1/3r_0$ and $-1/3r_0$ respectively.

there are three planes of symmetry, i.e. at $x_1 = 0$, $x_2 = 0$ and $x_3 = 0$, in this problem, symmetry is only applied on the $x_3 = 0$ plane. The reason for this is that later we want to use the model in a time-dependent planar flow field where α varies between 0 and 1 and thus here want to carry out tests which are not dependent on the type of applied BC's. All simulations carried out with Ca below the critical value are based on a finite element mesh where the element length in the fine region is $h_{fine} = r_0/6$ and the size of the fine region is $[-8r_0, 8r_0] \times [-7/6r_0, 7/6r_0] \times [-7/6r_0, 0]$. The total size of the computational domain is the same as that used in the previous sections. Furthermore the 2-level method with $N_v = 3$ is used for the VOF calculations. When the capillary number is above the critical value the drop deforms into a very thin thread and thus a very long computational domain with a very fine mesh is necessary in order to resolve the drop. In order to reduce the necessary computation time a coarser mesh is used for the $Ca = 0.13$ simulation with $h_{fine} = r_0/4$, however, $N_v = 4$ is used in order to increase the resolution of the VOF mesh. Furthermore the applied flow is stopped when the drop has been deformed approximately 16 times its initial diameter and the drop shape is studied as it begins to retract. In all simulations CFL=1/2 is specified for calculation of the variable time step size.

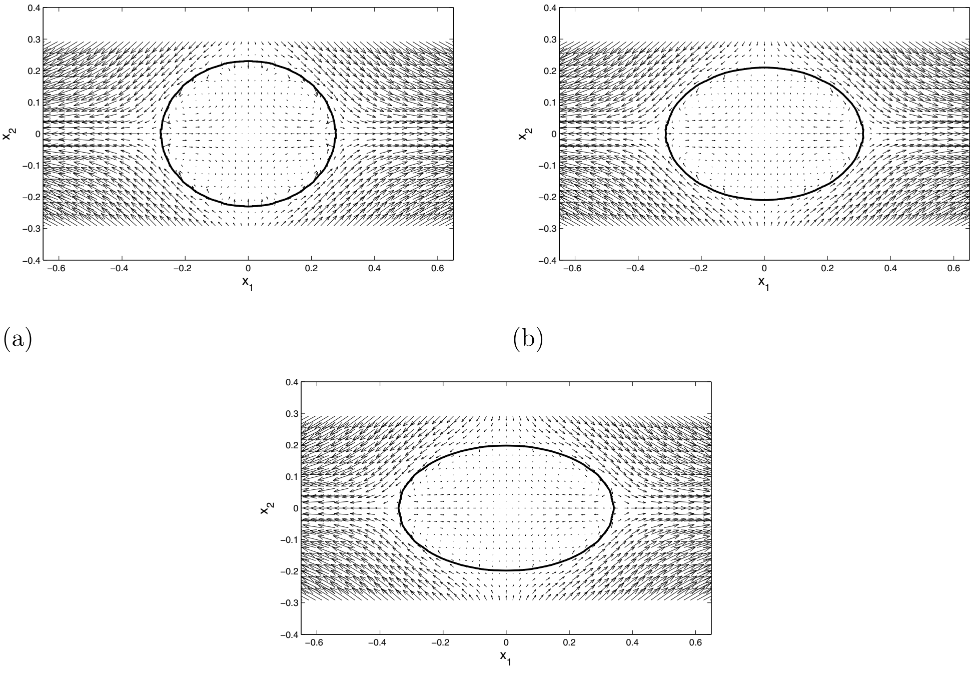
Results from the simulation with $Ca = 0.02$ are presented first because in this case the spurious currents (cf. section 2.8) arising from the CSS interfacial tension calculations are dominating the flow close to the interface. In Figure 3.29 the flow field and drop shape on the (x_1, x_2)-symmetry plane can be seen for two simulation times. From these figures it is clear that the spurious currents are dominating the flow and thus leading to a result which is physically incorrect. Especially one sees that the direction of the velocity field oscillates with time which results in a 'wobbling' interface.



$t^+ = 0.59$ $t^+ = 0.67$

Figure 3.29: Simulation in planar elongation with $\lambda = 1$ and $Ca = 0.02$. The flow field and drop shape on the (x_1, x_2)-symmetry plane is shown at dimensionless times $t^+ = 0.59$ and $t^+ = 0.67$.

Figure 3.30 (a)-(c) shows the steady state drop shape and flow field on the symmetry plane from the simulations with $Ca = 0.04$, 0.08 and 0.1. Here it is seen that the spurious currents are no longer dominating the flow and that a smooth drop interface is obtained at steady state. Furthermore the figures show that the flow inside the drops is very weak compared to the external flow and that 4 vortices are formed inside the drops. This is consistent with the theoretical prediction based on the small deformation theory, see (Bartok & Mason 1958).



(a)

(b)

(c)

Figure 3.30: Simulation in planar elongation with $\lambda = 1$ and (a) $Ca = 0.04$; (b) $Ca = 0.08$; (c) $Ca = 0.1$. The flow field and drop shape on the (x_1, x_2)-symmetry plane is shown after steady state is reached.

In Figure 3.31 the steady state deformation parameters obtained at the various subcritical capillary numbers can be seen from our VOF simulations. Also shown in the figure is the Taylor small deformation asymptote (Taylor 1934). The results indicate that for small capillary numbers the VOF simulations converge with the asymptotic result of Taylor. From our simulations we find that $Ca = 0.11$ results in a steady drop shape while $Ca = 0.13$ does not lead to a steady drop. Thus the VOF model leads to a critical capillary number somewhere between 0.11 and 0.13 in agreement with the literature, e.g. (Grace 1982).

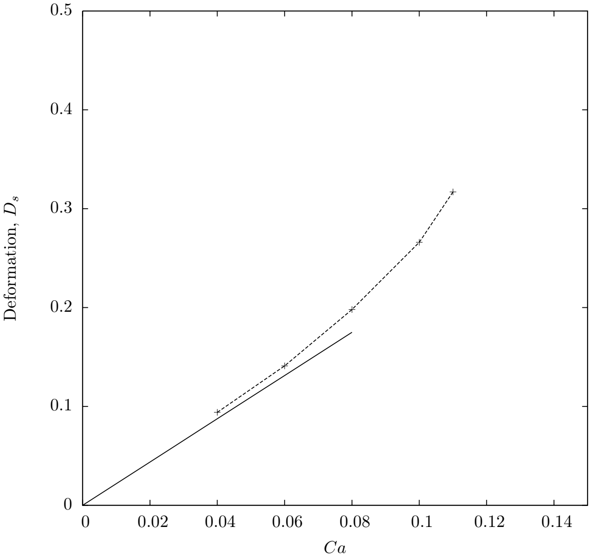
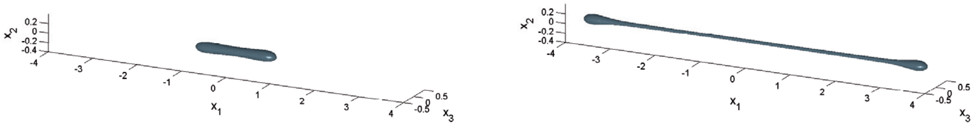


Figure 3.31: Steady state deformation parameter as a function of the capillary number for planar elongation and $\lambda = 1$. + and broken curve is the result obtained from our VOF simulations while the line is Taylor's the small deformation asymptote (Taylor 1934).

In the simulation with $Ca = 0.13$ the applied flow field is, as mentioned earlier, stopped after the drop has been elongated approximately 16 times its initial diameter. In Figure 3.32 the 3D drop shape is shown during the elongation process at dimensionless times $t^+ = 7$ (a) and $t^+ = 9$ (b). Here it is seen that the drop shape evolves into a thread like shape with bulby ends. At $t^+ = 9$ the applied flow ceases and the drop starts to retract.

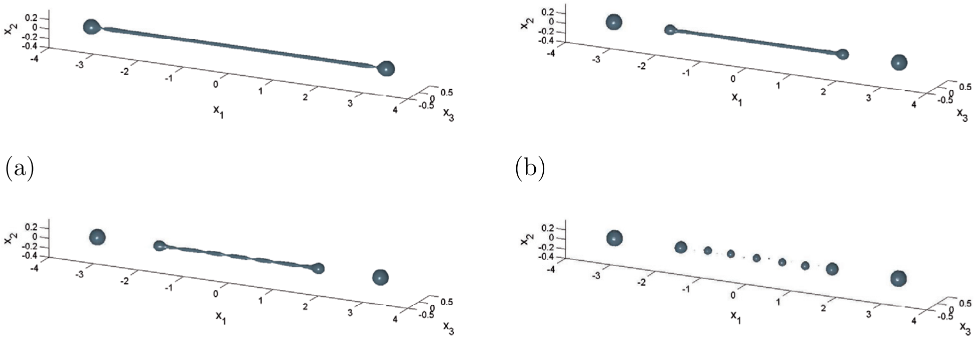
The development of the drop shape after cessation of the applied flow can be seen in Figure 3.33. First the drop starts to retract with the formation of spherical drops on the thread ends (a). After the end drops have pinched off the thread retracts further and forms new drops at the ends (b). The newly formed drops then pinch off and the remaining thread starts to break up into 5 equally sized drops (c). In Figure 3.33 (d) the final drop size distribution can be seen which also includes some sub cell-sized debris in between the drops.

The results obtained from the simulations carried out in the planar elongation flow indicate that the implemented model is able to predict the critical capillary number, at least when $\lambda = 1$. Furthermore the steady state deformation as a function of the capillary number, cf. Figure 3.31, agrees with Taylor's small deformation asymptote. It is noted though that if the model is to be used specifically for studying droplet deformations in planar elongation flows, symmetry conditions should be employed on all three symmetry planes which would reduce the required computational time significantly (thereby enabling



(a) (b)

Figure 3.32: Simulation in planar elongation with $\lambda = 1$ and $Ca = 0.13$. (a) $t^+ = 7$ and (b) $t^+ = 9$, the applied flow is stopped at $t^+ = 9$.



(a) (b)

(c) (d)

Figure 3.33: Drop shape evolution after cessation of the applied elongation flow ($\lambda = 1$). (a) $t^+ = 2$, (b) $t^+ = 7$, (c) $t^+ = 8$ and (d) $t^+ = 11$.

high resolution simulations).

3.5 Discussion and conclusions

In order to test the ability of the implemented FE-VOF model to handle drop deformation, drop break-up and drop merging, simulations have been carried out in simple flow fields. Since we want to simulate flow induced drop deformations in infinite mediums the effects of wall separation are first investigated. Here it is found that when the distance from the center of the domain to the walls is $24r_0$ wall effects are negligible. However, this also means that the domain is large compared to the drop size and thus a mesh with variable mesh size is applied. The mesh used has a fine central region with constant element size and a coarser mesh spanning from the fine region to the walls. The tests show that using four elements in each direction from the fine region to the walls is adequate. The coarse region is set up by halving the element length when going from the walls to the fine region.

Temporal convergence tests based on a mesh with $h_{fine} = r_0/7$ and $N_v = 1$ show that varying CFL from 1/2 to 1/8 results in the same deformation history when a drop with $Ca = 0.4$ and $\lambda = 1$ is sheared. This indicates that CFL=1/2 is sufficient when $h_{fine} = r_0/7$ at least when the drop deformation is limited.

Simulations have also been carried out in order to test the 2-level algorithm. In these tests 1-level and 2-level results are compared both qualitatively and quantitatively. From the qualitative analysis it is seen that the drop surface becomes somewhat 'wobbly' at the drop ends when the 2-level method is used. This is especially evident if the finite element mesh is coarse and the VOF mesh very fine, i.e. when N_v is high and h_{fine} is large. The 'wobbly' ends are not seen when the 1-level method is used. However, the tests carried out with the 2-level method show that the steady-state drop deformation, D_s, converges with VOF mesh refinement (while the resolution of the FE mesh is kept constant). Specifically the tests show that when the VOF element side lengths in the fine mesh region are $h_{vof} \leqq r_0/18$ the obtained D_s is more or less independent of mesh resolution.

More thorough tests of the code are carried out in simple shear. Simulations with varying Ca-numbers and $\lambda = 1$ result in steady-state deformations which are in good agreement with results reported in the literature. This includes the critical capillary number which we find to be around 0.44 compared to literature values of 0.42-0.43. However, when $\lambda = 0.1$ the 2-level method seems to lead to a D_s which is too high, at least compared to the results of Li et al. (2000). This is probably due to the method used to evaluate the viscosity in interface finite elements which is quite coarse. This is also supported by comparing simulations carried out with $h_{fine} = r_0/6$ and $h_{fine} = r_0/8$ which suggest that the results obtained with $\lambda = 0.1$ have not converged with respect to FE mesh refinement. A simulation is also carried out with a high viscosity ratio ($\lambda = 25$) which leads to an oscillating drop shape. By comparing the simulated results with the small deformation results of Rallison (1980) good agreement is found.

In order to test how the implemented model handles drop-drop interactions, simulations have been carried out where two drops are approaching each other in simple shear. In one case the drops flow past each other. In the second case the drops are positioned closer to each other and closer to the x_1-axis which results in coalescence. During the process of merging some of the continuous phase liquid gets trapped inside the disperse phase.

Simulations have also been carried out for $\lambda = 1$ in planar elongation. Here spurious currents are seen to be dominant when the capillary number is low ($Ca = 0.02$) which leads to non-physical results. For larger capillary numbers the effect of the spurious currents diminishes and the simulations lead to smooth interfaces. The steady-state deformation parameters obtained in the simulations are seen to converge towards Taylor's small deformation asymptote as Ca is decreased. From the simulations a steady drop shape was obtained for $Ca = 0.11$ while for $Ca = 0.13$ the deformation was unbounded. Thus the critical capillary number is between 0.11 and 0.13 which is in agreement with the experimental results of Grace (1982). In the simulations carried out with $Ca = 0.13$ the applied flow is stopped when the drop has elongated 16 times its initial diameter. At this point the drop shape consists of a cylindrical thread with bulby ends. After cessation of the applied flow the drop starts to retract which leads to pinch-off at the thread ends leading to the production of two isolated drops. The remaining thread retracts further with one more pinch-off sequence. Finally the rest of the thread breaks up into 5 equally sized drops with some debris in between.

It is noted that in all simulations carried out the difference between the initial and final volume of the disperse phase is below 10^{-2} %.

As mentioned above the 2-level method sometimes results in an interface which is somewhat 'wobbly' at the drop ends. Here a method, which has not been implemented, is proposed which might improve the accuracy in the 2-level calculations. If we recall from section 2.4.2.5 a correction is made to the fluid flux calculations in order to render the VOF cells divergence free. However, this correction method does not have a physical basis and thus the flow field used for updating the position of the interface (advection) does not in general obey the momentum equations. The irregularities of the interface observed when N_v is large may therefore be related to the interpolated flow field. In order to improve the accuracy of the flow field on the VOF mesh one could use a post processing procedure where, after the flow field has been solved, each interface finite element is treated as a subdomain discretized by VOF cells. Then the VOF cells belonging to a given interface finite element are set up with quadratic velocity nodes and constant discontinuous pressure nodes for which the momentum balance equations can be set up. The BC's for this subsystem are obtained by interpolating the velocities from nodes on the finite element in question and onto the nodes of the VOF cells which lie on the surfaces of the finite element. Since we know that the finite element is divergence free these boundary condition velocities also satisfy global mass conservation (of the sub-system). The interfacial tension calculations on the sub-system are straight forward since the F-field has already been calculated on the VOF cell vertices. Thus interfacial tension is included as if the 1-level method is applied. Solving this sub-system for the velocities should then result in a flow field on the VOF cell level which both obeys the momentum equations and the continuity equation. Fluid fluxes can then be calculated directly from the VOF mesh flow field. Another advantage of this method is that the variation of the viscosity within the interface finite elements can be taken directly into account, which would probably improve the accuracy of the calculations when λ is different from one. Although this method could improve the accuracy of the flow field on the VOF mesh level it would also require more extensive computation times because a large number of sub-system would need to be solved at each time step (each finite element belonging to the interface region would need to be considered). On the other hand, it is noted that each sub-system is independent and thus the method is amenable for parallelization.

Chapter 4

Drop deformation experiments in a rotor-stator device

As discussed in the introduction to this thesis much work has been devoted to the understanding of flow induced droplet deformation and break-up since the seminal work of Taylor in the 1930's (Taylor 1932, Taylor 1934). However, the vast majority of results reported in the literature are based on simple model flow fields (e.g. simple shear and planar elongation). Some studies include mixed flows where both shear and elongation are present, e.g. (Bentley & Leal 1986*b*, Stone et al. 1986), however, these flows are still homogeneous and well defined.

Here we want to study the deformation behavior of dispersed droplets in a flow field which is of comparable complexity to the flow fields present in industrial dispersing devices. More specifically an experimental apparatus which is based on a rotor-stator device consisting of two concentric cylinders with teethed walls is used for generating the flow. This device is situated at the 'Laboratory of Food Process Engineering' at ETH, Zürich, and is described in detail in (Kaufmann 2002). This experimental work was carried out during an external research stay at ETH in the fall of 2005/spring 2006 under supervision of Dr. Peter Fischer. A large part of the work carried out involves algorithm development for analyzing the raw experimental data which consists of digital movies. In this chapter the experimental setup is presented in section 4.1 while the material used and the experimental procedure are explained in sections 4.2 and 4.3. Since the data analysis procedure (automated image analysis) is a key part of the work a detailed description of the method used for movie analysis is given in section 4.4. In section 4.5 the experimental results are presented. Here the results in subsection 4.5.1 are based on time averaged values of the drop deformation parameter and a geometry based apparent shear rate. Thus the exact flow field generated in the device is not used here (this is the topic of the next chapter). The results presented in subsection 4.5.2 are concerned with drop break-up in the rotor-stator device where among other things the influence of viscosity ratio is investigated. Finally conclusions are given in section 4.6.

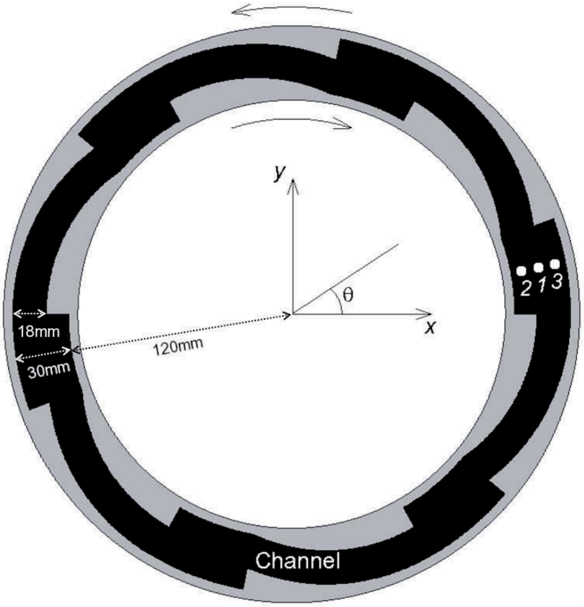


Figure 4.1: Sketch of the teethed concentric cylinders of the rotor-stator device.

4.1 Experimental setup

The rotor-stator device consists of two concentric cylinders having 6 teeth on each cylinder wall. A sketch of the device is shown in Figure 4.1. The channel in between the cylinders is filled with a liquid (the continuous phase) and, as the cylinders are rotated, a complex flow field will form. In Figure 4.2 a photograph of the setup is presented. At the bottom of the image the teethed cylinders can be seen. The channel depth is approximately 3cm. The rotation of the cylinders is powered by two independent computer controlled electrical motors. In the centre of the cylinders a vertical bar is placed which can be rotated manually and independently from the rotation of the cylinders.

Two digital cameras (Sony DFW V500, Japan) are mounted onto the central vertical bar. In order to track the drop position in the device it is necessary to have a camera which records images that include both the drop and parts of the cylinder walls, however, since the cylinder gap is much larger than the drop, the drop itself is poorly resolved in these images. In order to make accurate drop deformation calculations it is therefore necessary with another camera which records images with a magnification of the drop.

The camera used for monitoring the drop deformation (camera 1) is equipped with an unspecified zoom lens while the camera used for monitoring the position of the drop (camera 2) is equipped with a Nikon Micro-Nikkor 55mm f/2.8 lens. The vertical position of the cameras is adjusted such that the desired view angle is obtained. Both cameras

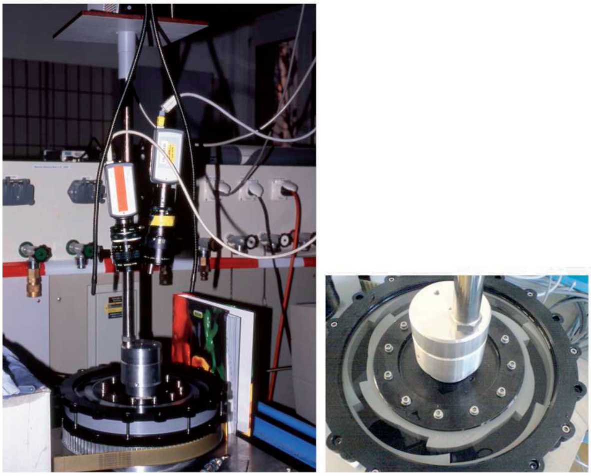


Figure 4.2: Photograph of the experimental setup.

are connected to computers whereby it is possible to store the camera outputs as movie streams. On top of the vertical bar a plate is mounted which holds an antenna light source. Proper positioning of the light source is important for providing adequate contrast between the drop and the surrounding liquid.

4.2 Materials

All experiments are carried out using Polyglycol B11/700 (HOECHST AG) as continuous phase while silicone oils (polydimethylsiloxane, Wacker) with different viscosities are used as drop phase. In Table 4.1 the density (ρ) and viscosity (μ) for the polyglycol and the silicone oils are given. Due to differences in density between the continuous phase liquid and drop phase liquid, TiO_2 is added to the drop phase to adjust the density. Furthermore adding TiO_2 makes the drop opaque which is necessary in order to distinguish the drop from the continuous phase. Due to the small amounts of TiO_2 added the viscosity of the drop phase did not change significantly from that of the pure silicone oils ($\lesssim 5\%$ see also later).

The interfacial tension coefficient (σ) between the continuous phase and the drop phase is measured at ambient temperature using the pendent drop method on a Dataphysics OCA 15 plus. Measurements are carried out on the pure polyglycol/silicone oil system. Since addition of TiO_2 does not add surface active agents to the system, TiO_2 should not affect the interfacial tension, see also (Feigl et al. 2003, Feigl, Megias-Alguacil, Fischer & Windhab 2007). The measured interfacial tension coefficients are listed in Table 4.2.

Name	ρ [g/cm^3]	μ[mPa $\cdot$ s]	λ
Polyglycol	1.048	$3.3{\cdot}10^{3\dagger}$	-
AK50	0.961	50	0.015
AK100	0.963	100	0.030
AK500	0.972	500	0.152
AK2000	0.974	2000	0.606
AK5000	0.974	5000	1.515

†Measured at ambient temperature.

Table 4.1: Physical data for the continuous phase liquid (polyglycol) and drop phase liquids (AK50-AK5000) used in the experiments. Density measured at $20°C$. Viscosity for the silicone oils at $23°C$ supplied by manufacturer. Also shown is the viscosity ratio λ for the polyglycol/AK systems.

System	σ [mN/m]
AK50/polyglycol	5.1
AK100/polyglycol	5.2
AK500/polyglycol	5.3
AK2000/polyglycol	5.6
AK5000/polyglycol	5.4

Table 4.2: Interfacial tension coefficients (σ) between droplet phase and continuous phase measured at ambient temperature.

4.3 Experimental procedure

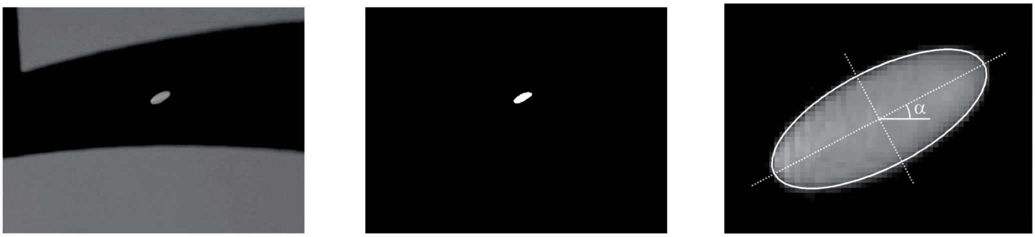
A single drop is added to the continuous phase using a syringe. In order to avoid boundary effects from the free surface and the bottom of the cylinder gap when the walls are rotating the drop is placed as close as possible to the channel centre in the axial direction. A typical drop size is on the order of 1mm. As indicated in Figure 4.1 the outer cylinder rotates counter-clockwise while the inner cylinder rotates clockwise. Each experiment is carried out using approximately the same angular velocity for both cylinders in order to minimize the movement of the drop which is important for capturing the drop images with the cameras. The initial position of the drop will determine the particle track it followed in the device. The time dependent shear and elongational stresses experienced by the drop will therefore be a function of the initial position as well as the angular velocities of the rotating cylinders. For each polyglycol/silicone oil system, experiments are carried out using three different initial (approximate) drop positions while the initial configuration of the cylinders is the same in all experiments. The three initial drop positions are indicated in Figure 4.1.

During an experiment the vertical bar holding the cameras is rotated manually in order to keep the drop within the view angle of both cameras while the output from the cameras is recorded for subsequent data analysis. Hence the frame of reference for the cameras varies as the cameras are rotated. The time scale of an experiment varies depending on the angular velocity of the cylinders; however, if a drop breaks up the experiment is stopped. The image data collected during each experiment is subsequently subjected to automated image analysis.

4.4 Image analysis

Due to the complexity of the flow field generated in the rotor-stator device, the drop deformation dependence on time will be highly complex. In order to extract deformation data as a function of time for a given experiment the movie sequence obtained from camera 1 is analysed. Due to the large amount of frames recorded during a single experiment the image analysis is automated using algorithms based on intrinsic Matlab functions. The first step in the image analysis is to load a given frame and identify the drop in the frame. This is done by filtering out objects in the image having pixel values below a pre-set threshold value. This leads to a binary image containing the drop and possibly some dust particles and/or light reflections on the surface of the continuous phase fluid. For each independent object in the binary image, the centre of mass, area, length and breadth are found. The length and breadth are found by fitting an ellipse to the object. In order to identify the drop in the image the lengths and breadths of the fitted ellipses are used for calculating the volumes of the corresponding spheroids which are compared to the volume of the original non-deformed drop. Furthermore an estimation of how closely an object can be represented by an ellipse is made by comparing the area of the object to the area of the fitted ellipse. Thus the drop is identified as the object which is best represented by an ellipse and whose spheroid volume is closest to the volume of the original non-deformed drop. The original image is then subjected to a fine filtering where the filter threshold value is set to a low value only for sharpening the edge of the drop. Since we know the approximate position of the centre of mass of the drop, the drop can easily be identified in the finely filtered image. The deformation parameter D is then determined by fitting an ellipse to the drop in the finely filtered image. In Figure 4.3 (a)-(c) sample images from the deformation analysis are shown. Figure 4.3 (a) shows the original image from camera 1. In Figure 4.3 (b) the image has been filtered using a high threshold value while in image (c) the original drop is seen together with the fitted ellipse. The error in the drop deformation obtained using an ellipse to fit the drop shape will increase as the drop deformation increases and therefore we only expect the measured drop deformation at large deformation to be useful for qualitative analysis.

In order to relate the drop deformation to the flow field it is necessary to know the drop position in the device at a given time. Figure 4.4 (a) shows a sample image from camera 2 where the drop and part of the teethed walls of the inner and outer cylinders can be seen. The main problem in determining the drop position in the device consists of relating the drop position in the image to the global position of the drop in the device. However, since we know the initial configuration of the system and the angular velocities of the cylinders we can calculate the configuration of the system at all times. Thus at a given time we use

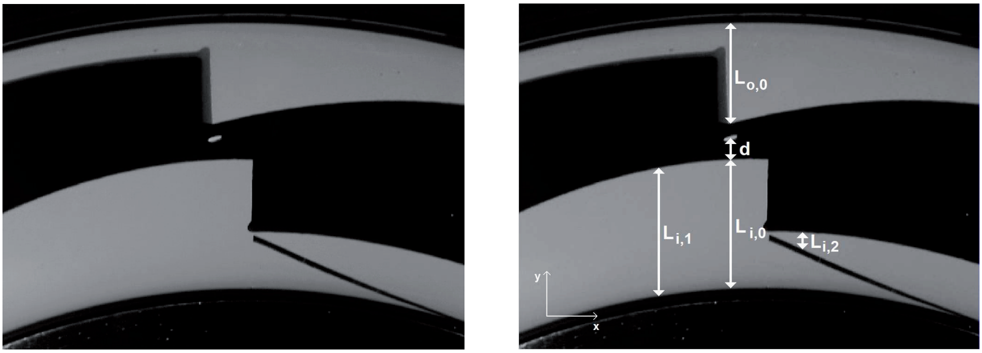


(a) (b) (c)

Figure 4.3: Example images from drop deformation analysis: (a) Original image from camera 1 showing the drop and part of the teethed walls. (b) Image after filtering out objects having pixel values lower than the high threshold value. (c) Crop of the drop after the ellipse fit (Ellipse is fitted to the drop after image (a) has been filtered using the low threshold value).

information on the thickness of the inner cylinder wall obtained from the corresponding movie frame to calculate the global coordinates of the drop. The procedure for locating the drop in the image is the same as described above for the deformation analysis. When the drop has been identified in the image the original image is filtered using a low threshold value in order to decrease interference by dust particles. After filtering, the distances in the $y-$direction, cf. Figure 4.4 (b), from the drop to the inner and outer cylinder walls together with the vertical thicknesses of the cylinder walls, $L_{i,0}$ and $L_{o,0}$, are found. Furthermore, in order to increase the accuracy of the position determination, the lengths $L_{i,1}$ and $L_{i,2}$ of the vertical cross-sections of the inner cylinder are also determined. The horizontal positions of these 'extra' lengths are at $x_0 - \beta$ and $x_0 + \beta$ respectively, where β is a predefined distance in the x-direction and x_0 is the $x-$position of the drop. From a given image we thus have the vertical thickness of the inner cylinder at three $x-$positions. It is seen from Figure 4.4 (a) and (b) that on part of the inner wall a line has been drawn. The reason is that in some parts of the device the wall thickness is constant. However, in order to determine the drop position uniquely at a given time it is necessary for the wall thickness to be unique in the proximity of the drop. The measured wall thickness thus corresponds to the distance in the $y-$direction spanning from the line to the beginning of the gap. When the image analysis has been performed for the entire movie sequence a profile of the gap width and drop position as a function of time is obtained. The next step is to calculate the global drop position in the device for the time steps from which image data has been extracted.

Calculating the radial position of the drop is straightforward using the lengths $L_{i,0}$ and d from the image analysis (cf. Figure 4.4). Calculation of the angular position θ is carried out using a minimization procedure where at time t the three lengths $L_{i,0}(t)$, $L_{i,1}(t)$ and $L_{i,2}(t)$ are compared to the corresponding lengths (L_i^*) calculated from the known configuration of the cylinders. At initial time $t_0 = 0$ the approximate angular position



(a) (b)

Figure 4.4: Example images from drop position analysis: (a) Original image from camera 2 showing the drop and the teethed walls. (b) Various lengths used for calculating the drop position in the device.

$\theta^*(t_{-1})$ of the drop is known from the initial positioning of the drop during the experiment. This position is then corrected by searching in the vicinity of $\theta^*(t_{-1})$ for the minimum

$$\min \sqrt{\sum_{k=0}^{2} \left(L_{i,k}(t_j) - L^*_{i,k}(t_j, \theta)\right)^2}, \quad \theta \in [\theta^*(t_{j-1}) - \epsilon; \theta^*(t_{j-1}) + \epsilon] \tag{4.4-1}$$

where $j = 0$ for at the initial time and ϵ is a pre-set angle which determines the search interval. The initial drop position is thus located at $\theta = \theta^*(t_0)$ where $\theta^*(t_0)$ is found from (4.4-1).

The drop position at the next time step t_j, $j = 1$ is then found by first calculating the new configuration of the cylinders using the known angular velocities. Hereafter the angular position of the drop is updated by searching and determining the minimum from (4.4-1) in the vicinity of the drop position from the previous time step t_{j-1}. Thus repeating this procedure for each time step results in the particle track for the drop. It is noted that any error in the experimentally measured angular velocities of the cylinders will result in accumulation of the error between the actual configuration and the calculated configuration of the system over time. Therefore after the particle track for the drop has been determined using the inner wall only, the vertical thickness of the outer wall $L_{o,0}(t)$ is compared to the corresponding calculated thickness of the outer wall $L^*_o(t)$. If there is a discrepancy between $L_{o,0}(t)$ and $L^*_o(t)$ the angular velocity of the outer wall is corrected until agreement is obtained. In most cases a small correction below 3% is necessary.

Figure 4.5 illustrates a deformation-vs.-time plot for an experiment. Furthermore the configuration of the cylinders and the drop position in the channel are shown at three

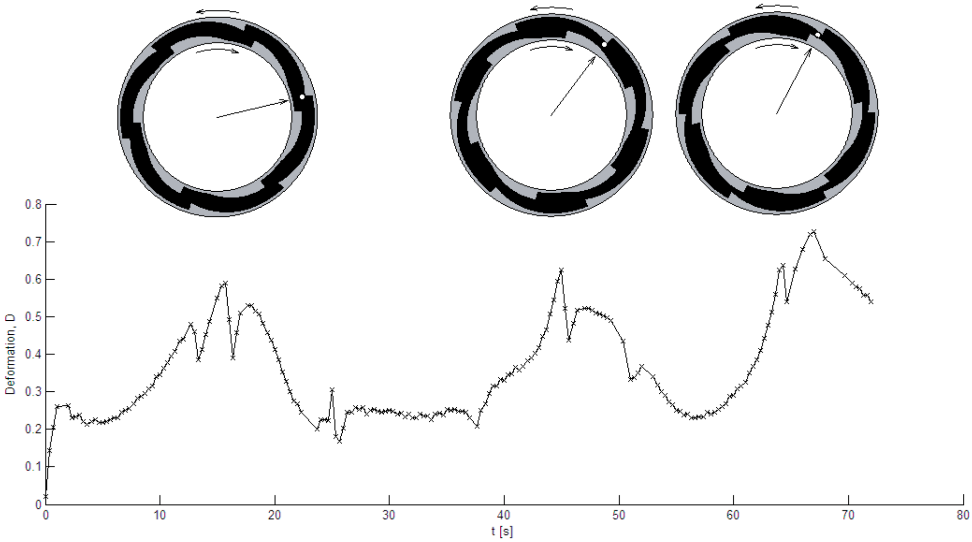


Figure 4.5: Example plot of drop deformation as a function of time. Also shown is the configuration of the system and the drop position (white dot) at three experimental times. The arrow originating from the centre of the cylinders indicates the location of the drop.

experimental times. From this plot it is seen that maxima in D occur when there is a local minimum in the gap width and that the maximum deformation (last peak) is obtained when the gap width is close to a global minimum.

4.5 Results and discussion

The presentation of the results is divided into two main parts. The first part is based on time-averaged values of the experimental data. Here the influence of the various parameters on the drop deformation is discussed, e.g. the effects of the viscosity ratio and the angular velocities of the cylinders. In the second part drop break-up in investigated in terms of the influence of viscosity ratio and the cylinder configurations.

4.5.1 Time averaged deformation

In order to compare experiments carried out using different experimental parameters, e.g. speed settings of the rotating cylinders, initial drop radius and viscosity ratio λ between drop phase and continuous phase, we base the analysis here on time averaged values.

In Table 4.3 an overview of the experiments which have been carried out can be seen. Included in this table are the angular frequencies of the inner ($\dot{\omega}_i$) and outer ($\dot{\omega}_o$) cylinders

No	System	$-\dot{\omega}_i$ [1/s]	$\dot{\omega}_o$ [1/s]	pos	$\langle\dot{\gamma}_{app}\rangle$ [1/s]	$\langle\dot{\gamma}_{app}\rangle_{calc}$ [1/s]
1	AK50	0.0414	0.0382	1	0.52	0.49
2	AK50	0.0621	0.0588	1	-	0.75
3	AK100	0.0414	0.0382	1	0.50	0.49
4	AK100	0.0414	0.0382	2	0.53	0.49
5	AK100	0.0414	0.0378	3	0.47	0.49
6	AK100	0.0621	0.0588	1	-	0.75
7	AK500	0.0414	0.0382	1	0.50	0.49
8	AK500	0.0414	0.0382	2	0.48	0.49
9	AK500	0.0414	0.0382	3	0.54	0.49
10	AK500	0.0828	0.0740	1	-	0.97
11	AK2000	0.0414	0.0382	1	0.50	0.49
12	AK2000	0.0414	0.0382	2	0.47	0.49
13	AK2000	0.0414	0.0382	3	0.53	049
14	AK2000	0.0621	0.0594	1	0.73	0.75
15	AK2000	0.0621	0.0588	2	0.72	0.75
16	AK2000	0.0621	0.0588	3	0.83	0.75
17	AK2000	0.0828	0.0700	1	-	0.95
18	AK5000	0.0414	0.0382	1	0.50	0.49
19	AK5000	0.0414	0.0382	2	0.47	0.49
20	AK5000	0.0414	0.0382	3	0.53	0.49
21	AK5000	0.0621	0.0588	1	0.76	0.75
22	AK5000	0.0621	0.0588	2	0.81	0.75
23	AK5000	0.0621	0.0594	3	0.74	0.75
24	AK5000	0.0828	0.0800	1	1.06	1.01
25	AK5000	0.1035	0.0975	1	1.27	1.25
26	AK5000	0.1242	0.1170	1	-	1.50

Table 4.3: Overview of experiments. The tabulation covers the silicone oil used as drop phase (System), angular frequency of inner cylinder $\dot{\omega}_i$, angular frequency of outer cylinder $\dot{\omega}_o$, the initial position of the drop (pos), cf. Figure 4.1, time averaged apparent shear rate $\langle\dot{\gamma}_{app}\rangle$ at the end of an experiment (values for experiments where drop break-up occurs are omitted) and calculated time averaged apparent shear rate $\langle\dot{\gamma}_{app}\rangle_{calc}$ using $r = \bar{R}_d$.

and the initial position of the drop.

Experiments indicated that higher angular frequencies of the inner and outer cylinders were necessary in order to induce break-up for the AK5000 system ($\lambda = 1.515$) than for the other systems. The experimental series on AK5000 therefore has the largest span in terms of angular speed settings of the cylinders which makes these data most suitable for investigating the relationship between cylinder speeds and drop deformation.

In order to relate the measured drop deformation to the flow in the channel without knowing the detailed flow field we define an apparent shear rate $\dot{\gamma}_{app}(r,t)$ as the shear rate at radial position r obtained in two concentric cylinders with inner radius $R_i(t)$ and outer radius $R_o(t)$ rotating with angular frequencies $\dot{\omega}_i$ and $\dot{\omega}_o$. The radii $R_i(t)$ and $R_o(t)$ are found from the lengths $L_{i,0}$ and $L_{o,0}$ in Figure 4.4. At a given time t the flow is assumed to be steady state with negligible inertial contribution and thus the momentum balance reduces to:

$$\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}(rv_\theta)\right) = 0 \qquad (4.5\text{-}2)$$

By applying the boundary conditions $v_\theta(r = R_i(t)) = \dot{\omega}_i R_i(t)$ and $v_\theta(r = R_o(t)) = \dot{\omega}_o R_o(t)$ the apparent shear rate becomes:

$$\dot{\gamma}_{app}(r,t) = \dot{\gamma}_{\theta r} = r\frac{\partial}{\partial r}\left(\frac{v_\theta}{r}\right) = \frac{2}{r^2}\left(\frac{R_o(t)^2 R_i(t)^2(\dot{\omega}_o - \dot{\omega}_i)}{R_o(t)^2 - R_i(t)^2}\right) \tag{4.5-3}$$

where by definition counter clock-wise is taken as positive angular direction. The apparent shear rate, $\dot{\gamma}_{app}(t)$, experienced by the drop at time t is then found by setting $r = R_d(t)$ in equation 4.5-3 where $R_d(t)$ is the radial position of the drop in the channel. Since the global geometry of the channel is not taken into consideration in the definition of the apparent shear rate we do not expect $\dot{\gamma}_{app}(t)$ to agree in detail with the actual shear rate. Therefore $\dot{\gamma}_{app}(t)$ is not suited for investigating the transient local flow. Instead we calculate the time average apparent shear rate, i.e.

$$\langle\dot{\gamma}_{app}(t)\rangle = \frac{1}{t}\int_0^t \dot{\gamma}_{app}(t')dt' \tag{4.5-4}$$

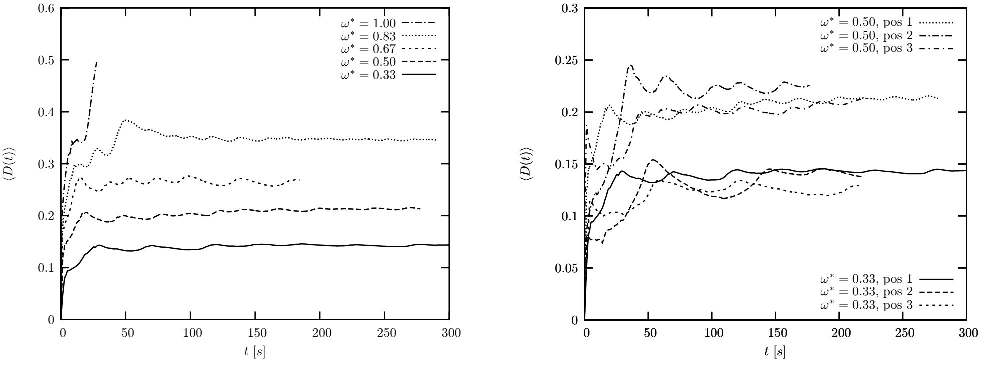
Figure 4.6 shows the time averaged drop deformation as a function of time for experiments carried out using AK5000 as drop phase. The experiments shown in Figure 4.6 (a) were all carried out using position 1 (cf. Figure 4.1) as the initial drop position in the cylinder channel. For comparing experiments carried out using different cylinder speed settings a dimensionless parameter ω^* is defined as:

$$\omega^* = \frac{\dot{\omega}_{o,k} - \dot{\omega}_{i,k}}{(\dot{\omega}_o - \dot{\omega}_i)_{max}} \tag{4.5-5}$$

which is the difference in the angular frequencies of the inner and outer cylinders for experiment k relative to the difference of the angular frequencies used in experiment 26, cf. Table 4.3.

It is seen from Figure 4.6 (a) that after an initial start-up period the fluctuations in the time averaged drop deformation, $\langle D(t)\rangle$, oscillate around a constant $\langle D\rangle$ with decreasing amplitudes as t increases. As expected the average drop deformation increases as the rotational speed of the cylinders is increased and in the case $\omega^* = 1$ the drop splits up short after start-up, hence for this experiment the critical capillary number Ca_{crit} is reached. In Figure 4.6 (b), $\langle D(t)\rangle$ is shown for different initial drop positions. It is seen that the initial drop deformation influences the initial fluctuations in the average drop deformation but in all cases these fluctuations seem to die out where after $\langle D(t)\rangle$ oscillates around a constant $\langle D\rangle$. Unfortunately experiments carried out with the drop initially positioned close to a cylinder wall cannot be carried out for long experimental times since the particle track tends to follow the rotation of the closer cylinder wall. This gives rise to experimental difficulties when following the drop with the cameras. However, comparison of the $\langle D(t)\rangle$ curves obtained for the various positions indicate that as time proceeds the time-averaged deformation tends towards the same limiting value for each ω^*.

Similarly to $\langle D(t)\rangle$ the time-averaged apparent shear rate, $\langle\dot{\gamma}_{app}(t)\rangle$, can be plotted as a function of time. This is shown in Figure 4.7. From this Figure it is seen that as t increases



(a) (b)

Figure 4.6: Time averaged drop deformation ($\langle D(t)\rangle$) as a function of time for experiments carried out using the AK5000 silicone oil as drop phase. (a) Experiments 18, 21 and 24-26 from Table 4.3; (b) Experiments 18-23 from Table 4.3

$\langle\dot{\gamma}_{app}(t)\rangle$ also oscillates around a constant $\langle\dot{\gamma}_{app}\rangle$ with decaying amplitude. It is also seen from Figure 4.7 that the time-averaged apparent shear rate, $\langle\dot{\gamma}_{app}\rangle$, seems to scale linearly with the angular velocities of the cylinders. It is therefore of interest to compare $\langle\dot{\gamma}_{app}\rangle$ to the shear rate obtained in between two concentric cylinders having radii corresponding to the average radii of the inner and outer cylinder walls of our device, i.e.

$$\langle\dot{\gamma}_{app}\rangle_{calc} = \frac{2}{r^2}\left(\frac{\bar{R}_o^2\bar{R}_i^2(\dot{\omega}_o - \dot{\omega}_i)}{\bar{R}_o^2 - \bar{R}_i^2}\right) \tag{4.5-6}$$

where the average cylinder wall radii $\bar{R}_i$ and $\bar{R}_o$ are calculated by

$$\bar{R} = \frac{1}{2\pi}\int_0^{2\pi} r(\theta)d\theta \tag{4.5-7}$$

Since equation 4.5-6 depends on r, a suitable average drop radius $\bar{R}_d$ needs to be defined. Since we do not know the drop particle track prior to an experiment, we will define it as the radius to the center of the channel. In Table 4.3 $\langle\dot{\gamma}_{app}\rangle$ values, which are chosen as the time averaged apparent shear rate obtained at the end of each experiment, and similar values obtained using (4.5-6) are given. These values show that the calculated values are within 10% of the values obtained from the experiments which indicates that on time average the drop experiences an apparent shear rate similar to one obtained in the centre of the channel formed by two concentric cylinders of radii $\bar{R}_i$ and $\bar{R}_o$. In Figure 4.8 the limiting time averaged drop deformation $\langle D\rangle$ for all AK5000 experiments is depicted as a function

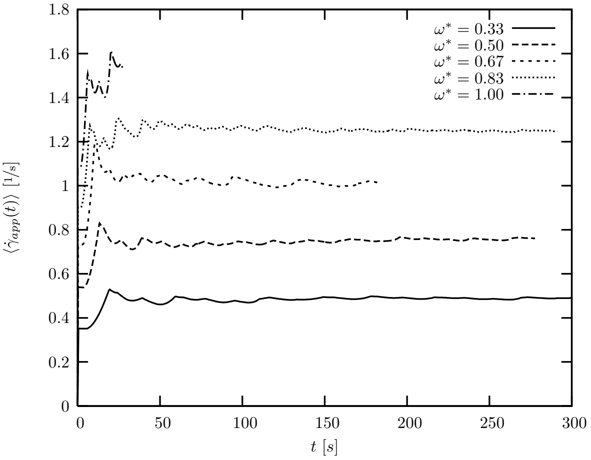


Figure 4.7: Time averaged apparent shear rate, $\langle \dot\gamma_{app}(t) \rangle$, as a function of time. Same experiments as shown in Figure 4.6 (a).

of the time averaged limiting apparent capillary number defined as

$$\langle Ca_{app} \rangle = \frac{\langle \dot\gamma_{app} \rangle_{calc} r_0 \mu_c}{\sigma} \tag{4.5-8}$$

Furthermore the Taylor approximation $\langle D \rangle = \langle Ca_{app} \rangle (19\mu_d + 16\mu_c)/(16\mu_d + 16\mu_c)$ is also shown. The figure indicates that at small capillary numbers the average drop deformation can be described by the Taylor approximation. However, as $\langle Ca_{app} \rangle$ increases, the linearity between $\langle D \rangle$ and $\langle Ca_{app} \rangle$ is lost which is to be expected.

Since the rest of the experiments in Table 4.3 are only carried out for two or three (AK2000) different settings of the rotational speed of the cylinders they are not suited for comparison between $\langle D \rangle$ and $\langle Ca_{app} \rangle$. However, plotting all the measured values of the time-average limiting deformation as a function of $\langle Ca_{app} \rangle \frac{19\mu_d+16\mu_c}{16\mu_d+16\mu_c}$ gives an impression of how close to the Taylor approximation $\langle D \rangle$ behaves. In Figure 4.9 such a plot can be seen. From the figure it is clear that at low deformations ($\langle Ca_{app} \rangle \frac{19\mu_d+16\mu_c}{16\mu_d+16\mu_c} \lessapprox 0.2$), $\langle D \rangle$ seems to be linear for all our systems.

The results presented above indicate that in absence of numerically calculated flow-field data the drop behavior in the rotor-stator device can, to some extent, be described using time-averaged values of the deformation and the defined apparent shear rate. One can therefore predict an approximate value of the average deformation by knowing the average radii and angular velocities of the cylinders, at least for small Ca numbers. This, though, does not give any detailed information on maximum values in the drop deformation which, obviously, is of interest if one needs to analyse the break-up process itself.

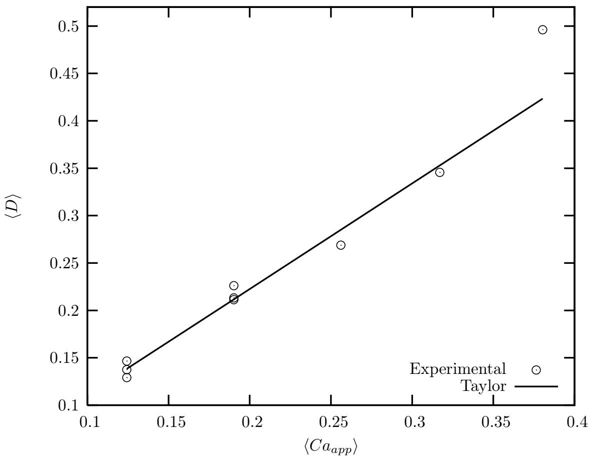


Figure 4.8: Time averaged deformation at steady-state $\langle D \rangle$ as a function of the apparent time averaged capillary number $\langle Ca_{app} \rangle$ for experiments carried out using the AK5000 system ($\lambda = 1.515$).

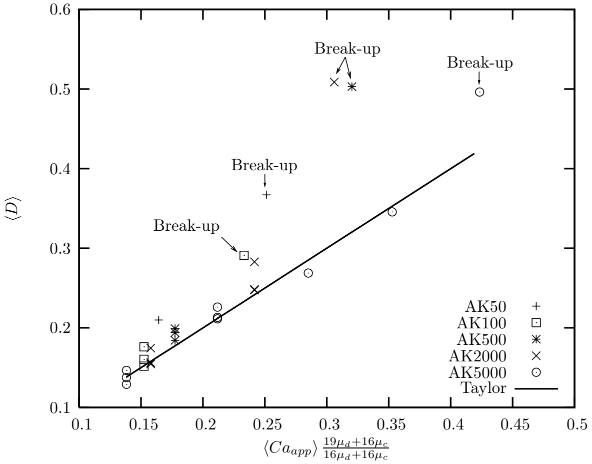


Figure 4.9: Time averaged deformation at steady-state $\langle D \rangle$ as a function of $\langle Ca_{crit} \rangle \frac{19\mu_d+16\mu_c}{16\mu_d+16\mu_c}$ for all experiments.

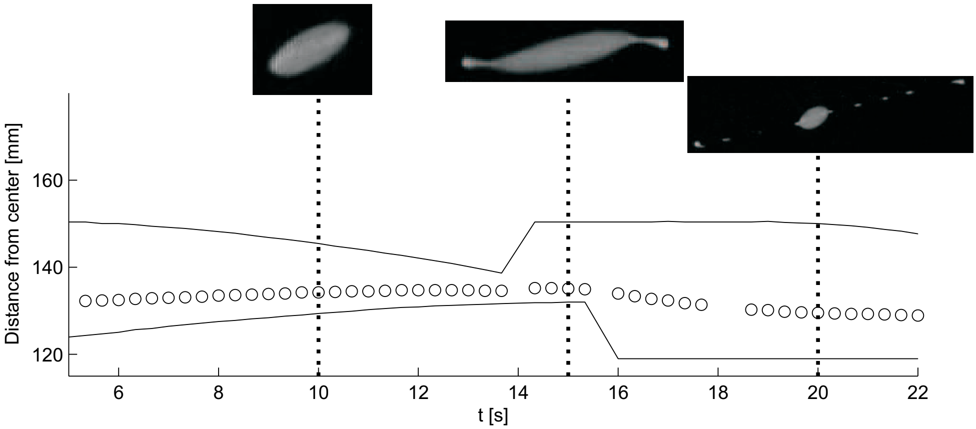


Figure 4.10: Investigation of drop-break-up for AK50 experiment 2 in Table 4.3. Full lines indicate the distance from the center of the cylinders to the inner and outer cylinders walls while circles indicate distance to the drop center as a function of time. Also shown are images recorded during the experiment at t = 10s, 15s and 20s.

4.5.2 Drop break-up

Here we will take a closer look at the experiments which resulted in drop break-up. These are experiments 2, 6, 10, 17 and 26 in Table 4.3. A number of experiments have also been carried out with the purpose of obtaining break-up. These are not included in Table 4.3, however, the results and experimental details are given in Figure C.1 - C.3 in Appendix C. In Figure 4.10 the distance from the center of the cylinders (or radius) to the drop (circles), inner cylinder wall (lower full curve) and outer cylinder wall (upper full curve) can be seen as a function of the experimental time for experiment 2 in Table 4.3 (AK50, λ = 0.015). These distances are obtained directly from the lengths $L_{i,0}$, d and $L_{o,0}$ in Figure 4.4 (b). Furthermore images of the drop are shown at 10s, 15s and 20s. The plot in Figure 4.10 gives an indication of the channel configuration relative to the drop position at any instant. For example the figure shows that at ≈ 14s the drop travels past a tooth on the outer cylinders and likewise it travels past a tooth on the inner cylinder at ≈ 15.5s. The break-up sequance shown in Figure 4.10 starts at around 14s where the ends of the drop start to separate and form small droplets. At 15s thin threads are seen to extend at the ends of the main drop (can be seen from the inserted images). As these threads become more stretched they split up into a number of small droplets (see inserted image at 20s). This break-up process is usually designated tip streaming. In Appendix C Figure C.1 the result is shown for another break-up experiment with the AK50 silicone oil. Here tip streaming is also observed.

In Figure 4.11 the drop position relative to the cylinder walls can be seen for experiment 6 in Table 4.3 (AK100, λ = 0.030). Also shown are images of the drop at selected times.

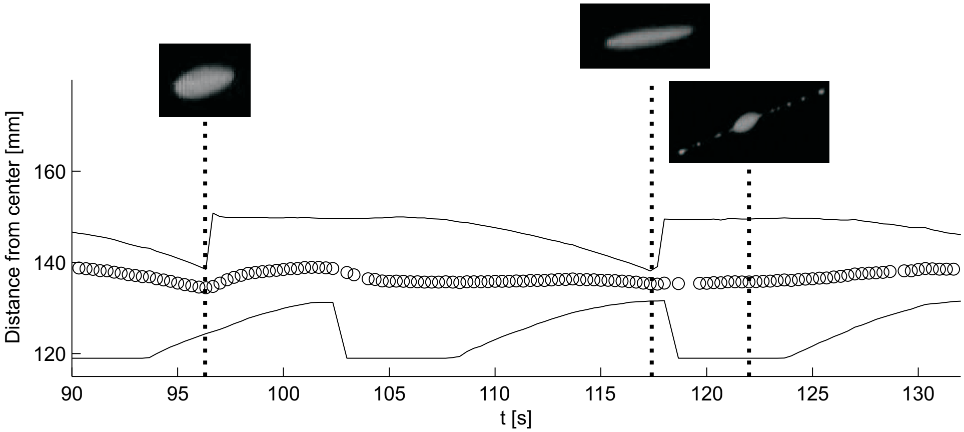


Figure 4.11: Investigation of drop-break-up for AK100 experiment 6 in Table 4.3. Full lines indicate the distance from the center of the cylinders to the inner and outer cylinders walls while circles indicate distance to the drop center as a function of time. Also shown are images recorded during the experiment at $t = 96.3$s, 117.4s and 122s.

At $t \approx 96$s the drop travels past a tooth on the outer wall and at $t \approx 103$s past a tooth on the inner wall. However, break-up is not seen in these two instances. On the other hand the drop starts to break-up at around $t \approx 117$ where the channel width is very close to a global minimum. When the drop has traveled past the tooth edges tip streaming is again observed. Figure C.2 in Appendix C shows the result from another AK100 experiment which also exhibits tip streaming.

Figure 4.12 shows the break-up results for experiment 10 in Table 4.3 (AK500). Here the drop deformation is large as the drop travels past the tooth on the outer cylinder at around 11s. When the drop has passed the tooth edge it keeps deforming slowly. At $t = 13$s the drop has deformed into an almost threadlike structure which starts to separate into individual drops at around 16s. The break up process observed here seems to be something between binary and capillary break-up. However, as in the previous cases, the break-up takes place when the drop experiences an expansion in the channel width. The AK500 experiment shown in Figure C.3 in Appendix C clearly show binary break-up.

The AK2000 ($\lambda = 0.61$) experiment shown in Figure 4.13 again shows that break-up takes places after the drop has traveled past a tooth edge. Here a binary break-up process is observed which yields 4 drops of comparable size and one much smaller drop in the center.

Finally the result from the AK5000 ($\lambda = 1.51$) experiment 26 in Table 4.3 can be seen in Figure 4.14. Here it is seen that the drop experiences a global minimum in the channel gap width at around 18s which, as it travels past the tooth edges, results in a highly elongated drop. However, break-up is not observed until the drop travels past a second tooth at 25s

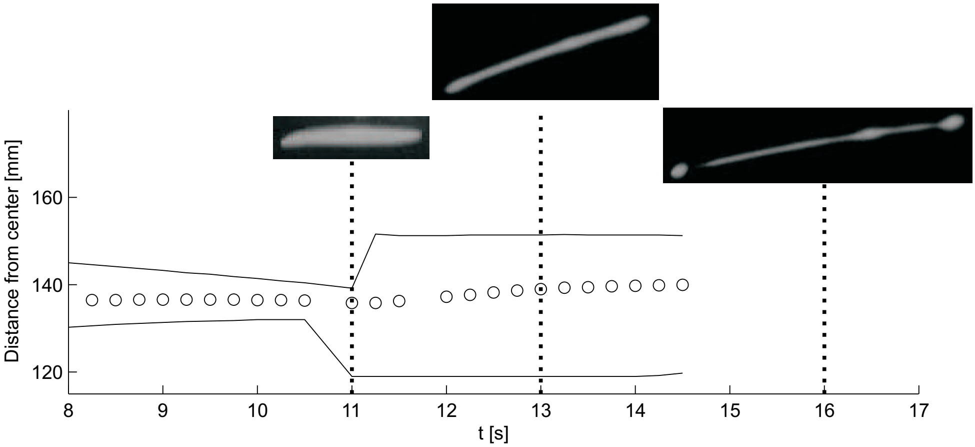


Figure 4.12: Investigation of drop-break-up for AK500 experiment 10 in Table 4.3. Full lines indicate the distance from the center of the cylinders to the inner and outer cylinders walls while circles indicate distance to the drop center as a function of time. Also shown are images recorded during the experiment at t = 11s, 13s and 16s.

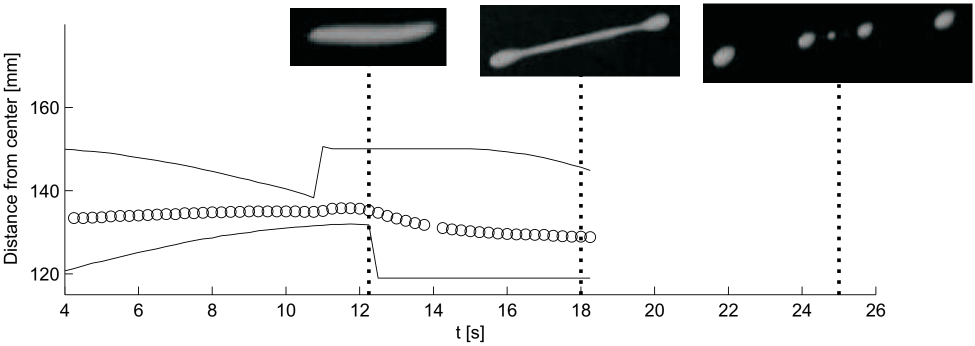


Figure 4.13: Investigation of drop-break-up for AK2000 experiment 17 in Table 4.3. Full lines indicate the distance from the center of the cylinders to the inner and outer cylinders walls while circles indicate distance to the drop center as a function of time. Also shown are images recorded during the experiment at t = 12.25s, 18s and 26s.

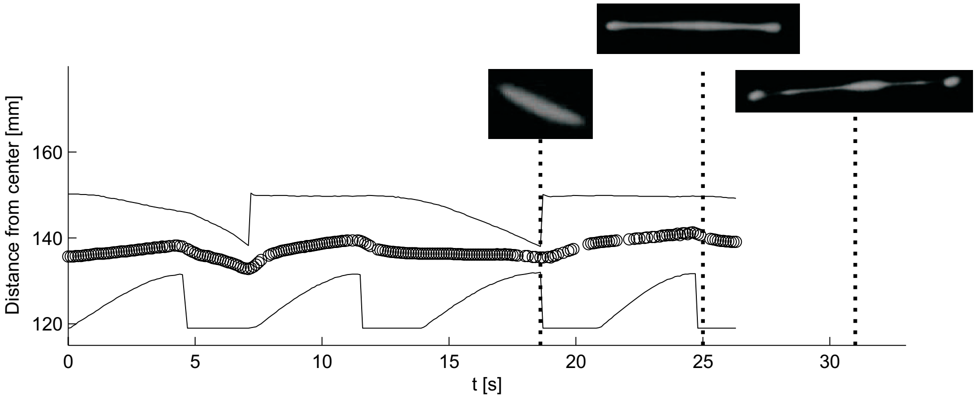


Figure 4.14: Investigation of drop-break-up for AK5000 experiment 26 in Table 4.3. Full lines indicate the distance from the center of the cylinders to the inner and outer cylinders walls while circles indicate distance to the drop center as a function of time. Also shown are images recorded during the experiment at t = 18s, 25s and 31s.

where the ends pinch off the main thread. At 31s the break-up process is nearly finished.

4.6 Discussion and conclusions

In this chapter a method for analysing the deformation of single drops suspended in a second immiscible liquid undergoing a time-dependent laminar and planar flow is described. The flow is generated in a rotor-stator device which consists of two concentric cylinders with teethed walls. By continuously monitoring the drop shape and its position in the device using two cameras, it is possible to obtain the transient deformation as well as the particle track of the drop by applying automated image analysis. Experimental results indicate that the time-averaged drop deformation scales linearly with an average apparent capillary number based on an apparent shear rate at least for small to medium deformations. Furthermore this linear relationship seems in agreement with Taylor's small deformation limit (Taylor 1934). Experiments carried out using different initial drop positions indicate that the limiting time-averaged drop deformation does not depend on the initial position.

Some of the experiments carried out resulted in drop break-up. From these experiments it is seen that the break-up process takes place as the drop travels from a minimum gap width region into a maximum gap width region. This indicates that the break up is induced by a relaxation of the flow field. The break-up experiments with the AK50/polyglycol and AK100/polyglycol systems exhibited tip streaming while the other systems either resulted in binary break-up or something between binary and capillary break-up.

Chapter 5

Comparison between simulations and experiments

In the previous chapter the experiments carried out in the rotor-stator device were presented. These included results based on an apparent shear rate and time-averaged deformation values. However, no detailed information on the flow field was used in the analysis. Now we will merge results from the rotor-stator experiments with our numerical VOF model and try to simulate the drop deformation history in the rotor-stator device. In order to do this we will use a procedure very much similar to the one described in (Feigl et al. 2003) (which is based on BI calculations). In this procedure the local flow field is first calculated by a numerical method along the particle track of the drop. In our calculations the particle track is obtained experimentally as explained in the previous chapter (image analysis). The local flow field is then used as input for generating time dependent BC's for our VOF simulation. Hereby the local flow experienced by the drop in the rotor-stator device is emulated in the calculational domain used for the VOF simulation. By using this method the VOF simulations can be carried out in the simple rectangular domain used previously for the simple shear and planar elongation simulations. More details will be given on the numerical procedure in the next section. Then in section 5.2 we will compare experimentally obtained drop deformations and simulated drop deformations. The simulations have also been carried out using a BI method[1] and in section 5.3 we compare the results from our VOF simulations with the BI simulations. Then some of the experimental results are combined with numerically calculated capillary numbers and compared to the Taylor small deformation theory in section 5.4. The chapter is ended with a discussion and some conclusions in section 5.5.

[1] These simulations were carried out by professor Kathleen Feigl at the Dept. of Mathematical Sciences, Michigan Technological University

5.1 Procedure for simulating the drop deformation in the rotor-stator device

The procedure for simulating the drop deformation can be separated into three steps these are:

1. Determine the particle track from the experiment.

2. Calculate the local flow numerically ($\dot{\gamma}$, $\dot{\epsilon}$).

3. Simulate the drop deformation.

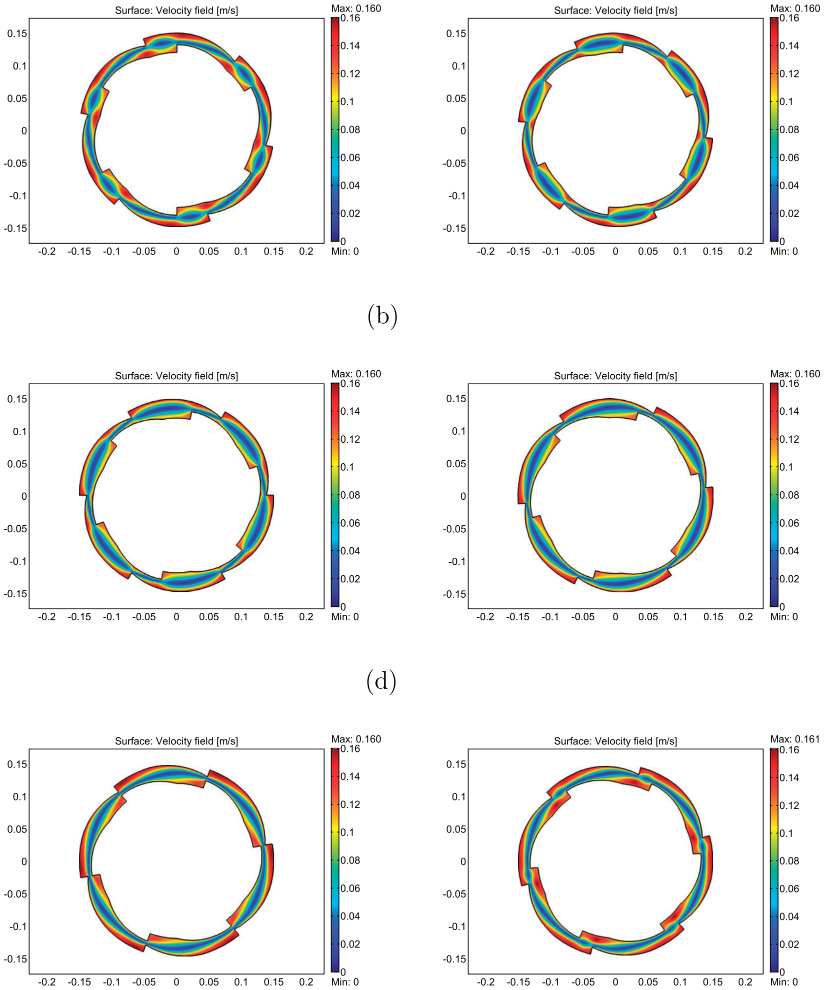
Step 1 is explained in the previous chapter (section 4.4).

The flow field calculations were carried out by Kathleen Feigl out using an Arbitrary Lagrangian-Eulerian (ALE) finite element scheme implemented specifically for simulating the flow in the rotor-stator device (Kikpka 2004). In order to illustrate how the flow field varies with the configuration of the cylinders, plots of the velocity field for different configurations can be seen in Figure 5.1. The simulations shown in Figure 5.1 are based on the Stokes flow solution of the momentum balance using the commercial Comsol Multiphysics CFD package and have only been carried out for illustrative purposes. In these example simulations the outer cylinder rotates counter clockwise while the inner cylinder rotates clockwise. Both cylinders rotate with angular frequency $\dot{\omega} = 1\mathrm{s}^{-1}$.

The flow field for the initial configuration of the cylinders (at time $t = 0\mathrm{s}$) is shown in Figure 5.1 (a). Due to the periodicity of the geometry the structure of the flow field is repeated 6 times. For each period the flow is characterized by a zone where it expands due to the gap width expansion. This expansion zone increases in size as the cylinders rotate (b)-(d). However, at the same time the edges of the teeth on the outer and inner cylinder approach each which results in a narrowing of the channel width along each tooth. In Figure 5.1 (e) the teeth edges on the outer and inner cylinder are exactly opposite each other leading to a global minimum in the gap width and the flow is seen to be quite uniform throughout the channel. In Figure 5.1 (f) the teeth edges have passed each other and the expansion zone seen in (a) starts to reform. It is noted that the initial configuration of the cylinders is obtained by rotating the inner cylinder clockwise by 5° and the outer cylinder counter clockwise by 5° in Figure 5.1 (f).

During an experiment the particle track depends on the initial location of the drop. In principle it is possible to calculate the particle track numerically using only the initial position of the drop and this has also been attempted (not included here). However, one problem with this procedure is that even a very small inaccuracy in the initial drop position determination leads to a 'wrong' particle track which due to the complexity of the flow field leads to a completely different result compared to the experimental particle track (which is the real particle track). Therefore all calculations have been based on the experimentally obtained particle track.

In order to couple the flow field in the rotor-stator device to the VOF simulations the local flow experienced by the drop along its particle track is decomposed into a shear rate



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.1: Numerically calculated flow field in the rotor-stator device. In the calculations the angular frequencies of the outer and inner cylinders were $\dot{\omega}_i = -\dot{\omega}_o = 1\text{s}^{-1}$. (a) Initial configuration of the cylinders used in the experiments. In (b) - (f) each cylinder is rotated by 5° relative to the previous figure.

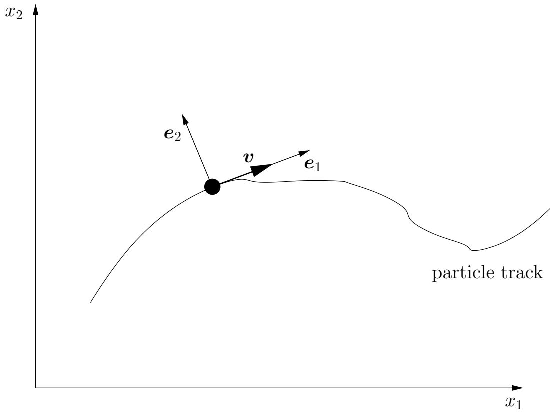


Figure 5.2: Decomposition of the flow into $\dot{\gamma}$ and $\dot{\epsilon}$.

and an elongation rate. It is noted that the flow in the channel is assumed to be planar, i.e. zero velocity component in the axial direction (x_3-direction). The flow field of interest is then characterized by the following velocity gradient tensor $\boldsymbol{E}$:

$$\boldsymbol{E} = \begin{pmatrix} \dot{\epsilon} & \dot{\gamma} \\ 0 & -\dot{\epsilon} \end{pmatrix} = G\alpha \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} + G(1-|\alpha|) \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \tag{5.1-1}$$

where G and α are defined in section 3.1. For a given time t the drop will be at position $\boldsymbol{x}(t)$ along its particle track and moving with velocity $\boldsymbol{v}(\boldsymbol{x})$, cf. Figure 5.2.

A new coordinate system is defined by the orthogonal unit vectors $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ with origin in the center of the drop $\boldsymbol{x}(t)$ and $\boldsymbol{e}_1$ pointing in the direction of the flow. Thus we have:

$$\begin{aligned} \boldsymbol{e}_1 &= \frac{1}{||\boldsymbol{v}||} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \\ \boldsymbol{e}_2 &= \frac{1}{||\boldsymbol{v}||} \begin{pmatrix} -v_2 \\ v_1 \end{pmatrix} \end{aligned}$$

In order to decompose the flow field at the drop center into the shear and elongation rates $\dot{\gamma}$ and $\dot{\epsilon}$ a rotational and translational coordinate transformation is applied to the numerically calculated velocity gradients, which yields (Feigl et al. 2003):

$$\dot{\gamma} = \boldsymbol{e}_1 \cdot \dot{\boldsymbol{\gamma}} \cdot \boldsymbol{e}_2, \quad \dot{\epsilon} = \frac{1}{2} \boldsymbol{e}_1 \cdot \dot{\boldsymbol{\gamma}} \cdot \boldsymbol{e}_1 \tag{5.1-2}$$

where $\dot{\boldsymbol{\gamma}}$ is the numerically calculated rate of strain tensor. Calculations of $\dot{\gamma}$ and $\dot{\epsilon}$ are carried out along the particle track at discrete time intervals and the results are tabulated. The third step in the computations is to carry out the VOF simulation of the drop shape.

This is done by using the tabulated values of the shear and elongation rate to apply time-dependent BC's on the domain boundaries. The domain BC's for the FE-VOF program are applied through the parameters α and G (equation 3.1-1, section 3.1). Thus for a given time step in the VOF simulation the shear and elongation rate are found by interpolation in the previously generated table and used for calculating α and G which now are time dependent. The relative contributions of elongation to the applied flow field will therefore vary with time, i.e. $\alpha = \alpha(t)$. Likewise the strength of the applied flow will also vary, i.e. $G = G(t)$.

5.2 Comparison between numerical and experimental data

Here comparison is made between experimentally obtained drop deformations and results from our FE-VOF code using the procedure described in the previous section. The flow field has been calculated numerically for experiments 11 and 18-20 in Table 4.3. However, due to the large computation times associated with the VOF simulations ($\approx$ 5 days with 8 processors) drop deformation simulations are carried out for experiment 11 ($\lambda = 0.61$) and 20 ($\lambda = 1.52$) only. The size of the fine region mesh used in the simulations is $[-5r_0, 5r_0] \times [-5r_0, 5r_0] \times [-6/5r_0, 0]$. The element side length of the fine part of the finite element mesh is $h_{fine} = r_0/5$ and the 2-level method with $N_v = 4$ is used. The total domain size is $[-24r_0, -24r_0] \times [-24r_0, -24r_0] \times [-24r_0, 0]$ with four finite elements spanning from the fine mesh region to the domain boundaries in each direction. It is noted that the orientation of the drop is unknown prior to the simulation and thus the dimension of the fine mesh domain in the x_2-direction has to be large enough to accommodate rotation of a deformed drop. It is therefore necessary to use a relatively coarse finite element mesh in order to carry out the simulations within reasonable time. The variable time step size is based on the setting CFL=1/4.

First an example dataset (experiment 11) obtained from the experiments and numerical flow field calculations is presented in Figure 5.3. Here the top figure shows the radius to the wall of the inner cylinder R_i, outer cylinder R_o and drop R_d as a function of time and indicates how the walls are positioned relative to the drop at any given time. The middle plot of Figure 5.3 shows the numerically calculated shear and elongation rate as a function of time. From this plot it is evident that the local flow field is highly complex with both shear and elongation present. Comparing the top and middle plot indicates that when the drop closes in on the edge of a tooth the magnitude of the shear rate increases until the drop travels past the edge, after which the shear rate diminishes. This is not surprising since the channel gap decreases as the drop approaches a tooth edge. This behavior is most noticeable when the drop is in between two approaching teeth. Although present, the magnitude of the elongation rate is somewhat smaller than the magnitude of the shear rate. However, it is seen that the elongation rate increases when the drop passes a tooth edge, which is in agreement with the expansion in the channel width occurring after each tooth. On the bottom plot of Figure 5.3 the experimentally obtained deformation as well as the numerically calculated capillary number are plotted as a function of time. The figure shows that the transient behavior of the numerically calculated Ca-number and experimentally obtained deformation are very similar with even small variation in Ca reflected by variations in D in some places.

In Figure 5.4 the results from the numerical calculations of the drop deformation can be seen for experiment 11 (upper plot) and 20 (lower plot). The top part of each figure shows measured and calculated deformations, the middle part shows the numerically calculated capillary number while the bottom part shows the numerically calculated α. Here we recall that the parameter α gives the relative contribution of elongation in the flow. This means that for $\alpha \in [0, 1]$ the flow corresponds to a mixture of shear and elongation with the extremes $\alpha = 0$ and $\alpha = 1$ corresponding to pure simple shear and pure planar elongation respectively. If values of α are negative the elongation direction is opposite compared to positive α's. Generally both plots in Figure 5.4 show that the numerically calculated deformation parameters are somewhat higher than those obtained experimentally but that the general, and in many cases also the detailed, deformation behavior is well described by the calculations. What is especially interesting is that peaks in the deformation are in many instances reproduced by the numerical calculations (although larger), e.g. at $t \simeq 20$s and $t \simeq 70$s in the upper plot in Figure 5.4. Correct prediction of the peaks is important if accurate predictions of drop break-up are to be made.

In both plots in Figure 5.4 the variation in the experimentally obtained deformations is reflected in the numerically calculated capillary number, this is especially evident in the lower plot in Figure 5.4 where flow past tooth edges is seen clearly as short timed decreases in the capillary number and deformation parameter. The decrease in D as the drop travels past an edge is also caught very well by the simulation shown in th elower plot in Figure 5.4. The variation in α gives a good indication of the flow experienced by the drop. In the lower plot in Figure 5.4 it is seen that α increases to a maximum and then decreases abruptly at the same time as the capillary number decreases. The increase in α is due to narrowing of the flow channel as teeth on the outer and inner wall are approaching, which results in an increase in elongational flow. When the drop passes a tooth, there is a sudden broadening of the channel which results in inversion of the elongation direction and thus a sign change in α. After the expansion, the channel width is constant or only changing slowly and thus the elongational contribution diminishes, i.e. $\alpha \simeq 0$. This behavior of α is also present in the upper plot in the figure but not as evident. It is noted that in the upper plot in Figure 5.4 the numerically calculated deformation shows a large peak at around 150s which is not seen in the experiment. The simulated drop shape at this peak was close to break-up with the formation of a neck in the central part of the drop. However, due to the flow field the drop did not break-up and relaxed back to an ellipsoidal shape. This behavior was not seen in the experiment. Nevertheless the results indicate that it is possible to simulate the qualitative transient drop shape in the rotor-stator device using the three-step procedure outlined in the previous section.

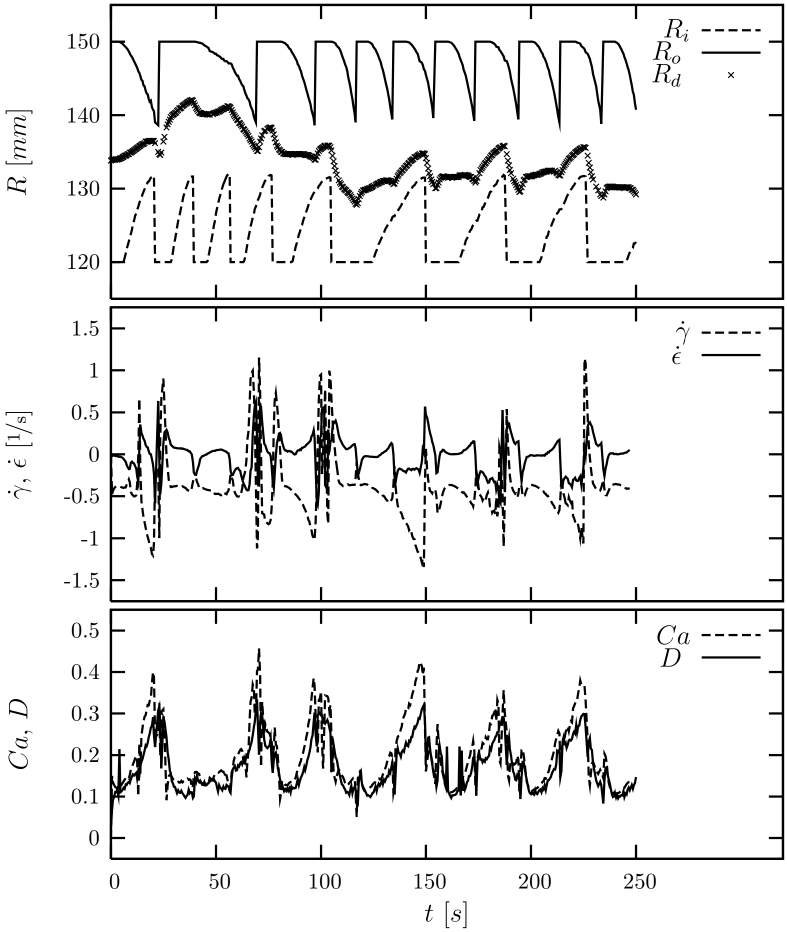
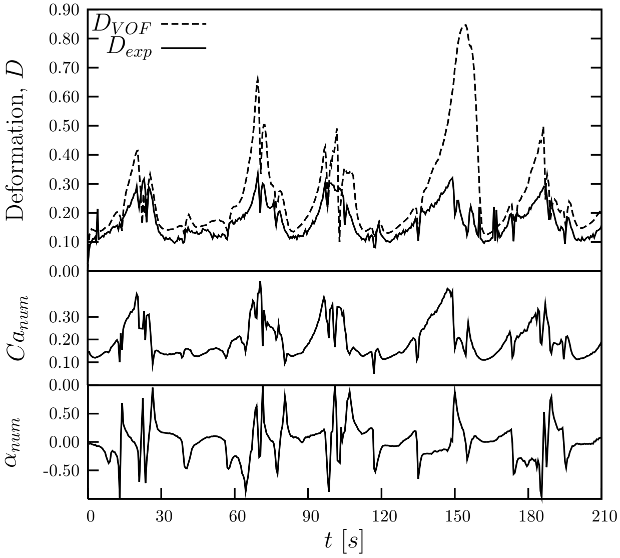


Figure 5.3: Bottom: Numerically calculated capillary number and experimentally obtained drop deformation as a function of time. Middle: numerically calculated shear and elongation rates as a function of time. Top: Radial distance from the centre of the rotor-stator device to the inner wall R_i, the outer wall R_o and the drop R_d as a function of time.



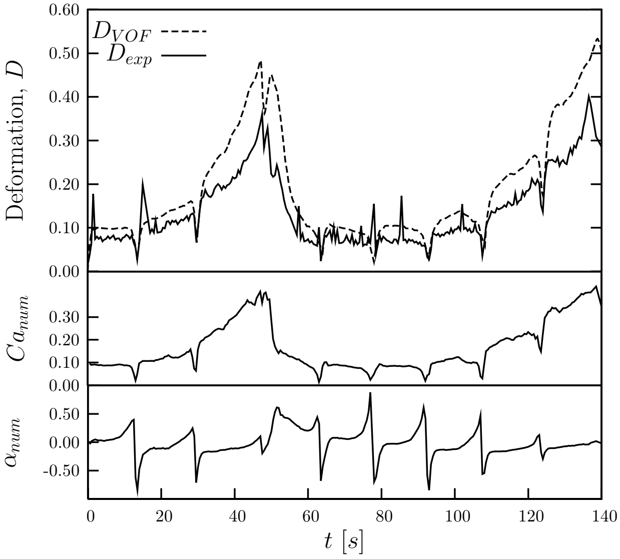


Figure 5.4: Plots showing the numerically and experimentally obtained deformation parameters D_{num} and D_{exp} for two different experiments. Also shown are the corresponding numerically calculated capillary numbers Ca_{num} and α-parameters α_{num}. Upper plot corresponds to experiment 11 in Table 4.3 while the lower plot corresponds to experiment 20 in Table 4.3.

5.3 Comparison between VOF and BI simulations

The drop deformations shown in Figure 5.4 have also been simulated using a BI code. In Figure 5.5 the results from the VOF simulations and the BI simulations are compared. In the upper figure the results from the simulations carried out for experiment 11 in Table 4.3 can be seen. Here excellent agreement between the two numerical methods is seen for most of the deformation sequence with the VOF simulation generally resulting in a slightly higher deformation compared to the BI simulation. However, there are some noticeable discrepancies around $t = 150$s which is where the VOF simulation resulted in a near break-up event. A similar behavior was also observed from the BI simulation (personal communication with Kathleen Feigl) and thus the differences in the deformation behavior here are probably due to differences in how the two methods handle the drop shape near the break-up limit.

In the lower plot in Figure 5.5 the simulated results for experiment 20 can be seen. Near perfect agreement is again seen here until $t \approx 110$s where after the results start to deviate considerably. The exact reason for this discrepancy is unknown, however, it is noted that the VOF result seems to more consistent with the experimental deformation (lower plot in Figure 5.4) than the the BI result.

With the exception of the results for $t > 110$s in the lower plot in Figure 5.5 both methods seem to predict the same drop deformation history which supports the validity of the implemented VOF model. It is noted that the coarseness of the finite element mesh used in the VOF simulations probably can explain the slightly higher deformations obtained compared to the BI simulations, cf. also section 3.3.2.

5.4 Comparison with Taylor theory

As mentioned earlier calculations of the flow field has, in addition to experiment 11 and 20 in Table 4.3, also been calculated for experiment 18 and 19. Since experiments 18-20 are carried out using the same silicone oil/polyglycol system λ is the same for these 3 experiments. Thus 3 datasets based on the same λ but with various starting positions of the drop are available. Therefore in order to investigate how closely the numerically calculated Ca-numbers and the experimentally obtained deformations are related, a plot of D vs. Ca is given in Figure 5.6 for experiment 18-20. From the figure it is seen that there is considerable spread in the data points. However, there seems to be a concentration of points along a single curve indicating that there is a relation between D and the numerically calculated Ca-numbers. Furthermore this curve seems to superimpose the Taylor small deformation asymptote as Ca decreases. The spread of the data points can partly be explained by uncertainties in the deformation measurements due to the manual camera movements which in some cases results in blurred images when the camera is moved fast relative to the camera shutter speed. However, it is also noted that the measured deformations are not necessarily steady state values wherefore one must expect fluctuations in the data.

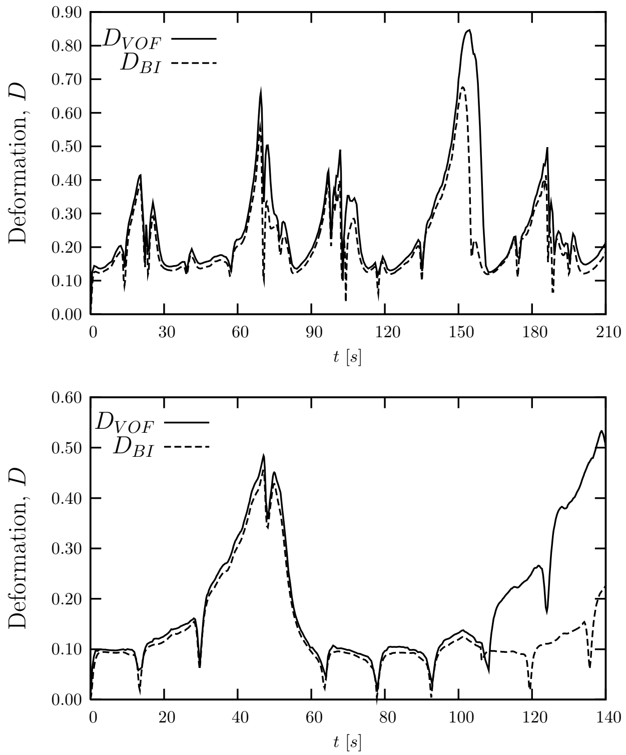


Figure 5.5: Comparison of drop deformation results from the VOF and BI simulations. Upper plot corresponds to experiment 11 and lower plot corresponds to experiment 20.

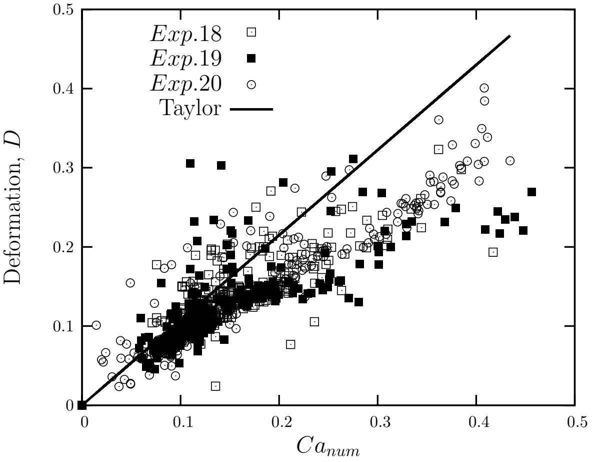


Figure 5.6: Experimentally obtained deformation as a function of the numerically calculated capillary numbers for experiment 18-20 in Table 4.3. Also shown is the Taylor small deformation limit.

5.5 Discussion and conclusions

A three step numerical procedure has been applied in order to simulate the deformation behavior of a drop suspended in a second immiscible liquid undergoing a flow generated by a rotor-stator device. The procedure consists of 1) determining the particle path of the drop from experimental data using automated image analysis, 2) numerical calculation of the local flow experienced by the drop along its particle path and 3) simulating the drop deformation in a simple rectangular domain using the previously calculated local flow field to generate appropriate time dependent BC's. One of the underlying assumptions in this procedure is that the real flow field in the proximity of the drop is homogeneous, i.e. the drop is small relative to the spatial variation of the velocity gradients. Furthermore wall effects are neglected which is necessary because the drop shape simulations (VOF) are carried out in a different domain than the 'macroscopic' flow field calculations.

Comparison of the simulated and experimentally obtained drop deformations shows that the simulations lead to a somewhat larger deformation than experimentally observed. However, the qualitative deformation history is generally very well predicted by the simulations. This includes the short timed decreases in the deformation parameter which usually occurs when the drop passes the edge of a tooth. Furthermore maxima in the deformation curve are also captured by the simulations although in some cases the maximum in the simulated curve is more than 100% larger than experimentally observed.

Drop shape simulations have also been carried out using a BI method and the results are compared to the VOF results. Generally good agreement is found between the two methods which supports the correctness of the drop shape calculations. The exact reason why the simulations lead to a higher deformation than experimentally observed is not known. However some possible explanations are:

1. Bias in the interfacial tension measurements (i.e. measured σ to low).
2. Bias in the viscosities.
3. Numerical flow field calculations over predict the strength of the flow.
4. The flow field cannot be assumed to be homogeneous throughout the vicinity of the drop.

1) would explain why the simulated deformation history is over predicted for both the simulations carried out. It is noted here that the interfacial tension measurements were carried out using the pure silicone oil (i.e. without TiO_2 added). The presence of TiO_2 might interact with the drop interface and change its surface properties, however, Feigl et al. (2007) show that the effect of TiO_2 on interfacial tension is minimal. Re 2) the drop viscosity used as input for the simulations is that for the pure silicone oil (as supplied by the manufactorer). Here the presence of the TiO_2 particles will increase the viscosity. However, if the particles are assumed to be spherical a rough estimated of the viscosity increase using the Einstein viscosity relation is $\approx 5\%$. This can hardly explain the differences in the simulated and experimentally observed drop deformations (for $\lambda = \mathcal{O}(1)$ the steady state deformation is a very weak function of λ, cf. Taylor's expression in equation 1.1-2). Other possible explanations for the differences in the simulated and experimental drop deformations are related to the numerical method itself. Nevertheless the results from this study do indicate that the numerical procedure is applicable for simulating drop deformation in highly complex and unsteady flows - at least for obtaining qualitative drop shape predictions.

Chapter 6

Summary and possibilities for future work

In this project a numerical model has been implemented for direct simulation of flow induced deformations of drops suspended in a second and immiscible liquid phase. The model is intended as a tool for designing emulsion systems. Therefore it is required that the model is able to handle drop deformations, including break-up and coalescence, in simple as well as in complex flow fields. For this reason we have chosen to base the modeling work on the Volume of Fluid (VOF) method combined with the Finite Element (FE) method. This choice of methods is connected with the ability of the VOF method to track complex interface geometries including interface separation and folding and the ability of the FE method to handle complex domain boundaries. However, due to the 3 dimensional nature of the flow problem the model needs to be fully 3D which requires long computation times. In order to reduce the computational cost associated with the simulations a 2-level method has been applied where different meshes are used for the flow field calculations and the fluid tracking problem. Thus the flow field calculations which are by far the most computationally expensive part of the code can be carried out on a relatively coarse mesh while the interface tracking can be carried out on finer mesh.

Tests of the code in simple flow fields such as simple shear and planar elongation, generally show that results from the model agree well with results reported in the literature. However, some deterioration of the interface is seen when the resolution of the interface is much higher than the resolution of the calculational mesh. This is most noticeable when the calculational mesh is very coarse. A possible solution to this problem is discussed in section 3.5 and could be a subject of future work.

A generic problem with the VOF method is the production of artificial (or spurious) currents when interfacial tension is present. These currents manifest themselves as small circulation zones in the vicinity of the interface of an otherwise stagnant spherical drop. If the applied flow field is weak compared to the interfacial tension forces, these currents become significant, which results in an unstable solution. This problem was e.g. observed in a simulation with low capillary number in planar elongation (section 3.4). Methods which rely on smoothing of the F-field are known to reduce the magnitude of the spurious currents, e.g. (Renardy, Cristini & Li 2002). Applying such a smoothing procedure would require only small changes to the implemented model and thus a study of the effect of smoothing of the F-field could be the basis for a small future project.

In the numerical work carried out in this project both liquid phases are considered to be Newtonian and the interfacial tension coefficient is treated as constant, i.e. no surfactants are present. Therefore future numerical work could be to implement non-Newtonian effects in the flow field calculations, see e.g. (Khismatullin et al. 2006), and to implement a model for surfactant transport and variable interfacial tension, see e.g. (Renardy, Renardy & Cristini 2002). However, this would require some rather extensive changes/additions to the implemented code and thus would be suited as basis for larger projects.

In addition to the model implementation and testing a number of experiments have been carried out in order to study the effect of a complex flow field on the deformation behavior of single drops. These experiments were carried out using a rotor-stator device. One of the key elements in this work is a method for analysing the experimental raw data which consists of 2 movie sequences. Among other things, the analysis procedure consists of an automated image analysis, which is used for obtaining the transient drop deformation and the drop trajectory in the device. Due to the complexity associated with the boundaries of the rotor-stator device numerical calculations are necessary in order to obtain the flow field. However, in order to analyse the data without having to rely on numerical calculations a geometry based apparent shear rate is defined. The apparent shear rate is then used for analysing time averaged values of the experimentally obtained drop deformations. The analysis indicates that it is possible to relate the average drop deformation with the apparent shear rate. A number of the conducted experiments resulted in drop break-up. The break-up process is analysed qualitatively by inspection of the recorded movies and by relating the initiation of the break-up process with the configuration of the apparatus. From this qualitative analysis it is found that for small viscosity ratios ($\lambda \lesssim 0.1$) tip streaming is predominant. For larger viscosity ratios a mixture of binary and capillary break-up is observed. However, in all cases the break-up process is initiated when the drop travels past a minimum in the wall separation.

In order to simulate the deformation behavior in the rotor-stator device the three-step procedure presented in (Feigl et al. 2003) is applied. Here the experimentally obtained drop trajectory is used together with a numerical procedure to calculate a local flow field for the drop along its path. This local flow field is then tabulated in terms of time dependent shear and elongation rates which are used as input for the implemented FE-VOF model. Comparison between the simulated and the experimentally obtained transient drop deformations shows good qualitative agreement. However, the simulations lead to somewhat larger deformations than those experimentally observed. Nonetheless the results indicate that it is possible to predict the transient drop shape qualitatively using the three-step procedure.

The results from our FE-VOF simulations are also compared to Boundary Integral (BI) simulations (the BI simulations were carried out prior to the FE-VOF simulations in (Egholm, Fischer, Feigl & Szabo 2007)). Generally good agreement is found between the two methods which supports the validity of the implemented FE-VOF model.

Although simulations of drop break-up in the rotor-stator device was not attempted with the implemented model (this could be a subject for future studies) the results indicate that the model can be used as a tool for designing dispersion processes. Specifically the procedure can be used for investigating the transient drop shape along different particle tracks in a given dispersing device. For example a commercial CFD package can be used for obtaining the macroscopic flow field in the device including numerically calculated particle tracks and the associated time dependent local flow parameters ($\dot{\gamma}$ and $\dot{\epsilon}$). Then by using the local flow as input for the FE-VOF model the drop shape along each particle track can be studied numerically. This way one could e.g. study which domains in the dispersing device result in drop break-up and also study the break-up type and resultant drop size distribution.

Bibliography

Abramowitz, M. & Stegun, I. A. (1972), *Handbook of Mathematical Functions*, Dover, New York.

Acrivos, A. & Lo, T. S. (1978), 'Deformation and Breakup of a Single Slender Drop in an Extensional Flow', *Journal of Fluid Mechanics* **86**, 641–672.

Ashgriz, N. & Poo, J. Y. (1991), 'Flair: Flux Line-Segment Models for Advection and Interface Reconstruction', *Journal of Computational Physics* **93**, 449–468.

Babaei, R., Abdollahi, J., Homayonifa, P., Varahram, N. & Davani, P. (2006), 'Improved Advection Algorithm of Computational Modeling of Free Surface Flow Using Structured Grids', *Computer Methods in Applied Mechanics and Engineering* **195**, 775–795.

Bartok, W. & Mason, S. G. (1958), 'Particle Motions in Sheared Suspensions VII. Internal Circulation in Fluid Droplets (Theoretical)', *Journal of Colloid Science* **13**, 293–307.

Bentley, B. J. & Leal, L. G. (1986*a*), 'A Computer-Controlled Four-Roll Mill for Investigations of Particle and Drop Dynamics in Two-Dimensional Linear Shear Flows', *Journal of Fluid Mechanics* **167**, 219–240.

Bentley, B. J. & Leal, L. G. (1986*b*), 'An Experimental Investigation of Drop Deformation and Breakup in Steady Two-Dimensional Linear Flows', *Journal of Fluid Mechanics* **167**, 241–283.

Biausser, B., Guignard, S., Marcer, R. & Fraunié, P. (2004), '3D Two Phase Flows Numerical Simulations by SL-VOF Method', *International Journal for Numerical Methods in Fluids* **45**, 581–604.

Bird, R., Armstrong, R. C. & Hassager, O. (1987), *Dynamics of Polymeric Liquids Volume 1 Fluid Mechanics*, John Wiley & Sons.

Bird, R. B., Stewart, W. E. & Lightfoot, E. N. (2002), *Transport Phenomena*, 2nd. edn, John Wiley & Sons Inc.

Birkhofer, B. H., Eischen, J.-C., Megias-Alguacil, D., Fischer, P. & Windhab, E. J. (2005), 'Computer-Controlled Flow Cell for the Study of Particle and Drop Dynamics in Shear Flow Fields', *Industrial and Engineering Chemistry Research* **44**, 6999–7009.

Brackbill, J. U., Kothe, D. B. & Zemach, C. (1992), 'A Continuum Method for Modeling Surface Tension', *Journal of Computational Physics* **100**, 335–354.

Buckmaster, J. D. (1972), 'Pointed Bubbles in Slow Viscous Flow', *Journal of Fluid Mechanics* **55**, 385.

Buckmaster, J. D. (1973), 'The Bursting of Pointed Drops in Slow Viscous Flow', *Transactions of the ASME. Series E, Journal of Applied Mechanics* **40**, 18–24.

Chaffey, C. E. & Brenner, H. (1967), 'A Second-Order Theory for Shear Deformation of Drops', *Journal of Colloid and Interface Science* **24**, 258–269.

Cox, R. G. (1969), 'The Deformation of a Drop in a General Time-Dependent Fluid Flow', *Journal of Fluid Mechanics* **37**, 601–623.

Cristini, V., Blawzdziewicz, J. & Loewenberg, M. (2001), 'An Adaptive Mesh Algorithm for Evolving Surfaces: Simulation of Drop Breakup and Coalescence', *Journal of Computational Physics* **168**, 445–463.

Cristini, V., Guido, S., Alfani, A., Blawzdziewicz, J. & Loewenberg, M. (2003), 'Drop Breakup and Fragment Size Distribution in Shear Flow', *Journal of Rheology* **45**, 1283–1298.

Cunha, F. R. & Loewenberg, M. (2003), 'A Study of Emulsion Expansion by a Boundary Integral Method', *Mechanics Research Communications* **30**, 639–649.

DeBar, R. B. (1974), Fundamentals of the KRAKEN Code, Technical report, Lawrence Livermore Laboratory.

Egholm, R. D., Fischer, P., Feigl, K. & Szabo, P. (2007), 'Experimental and Numerical Analysis of Droplet Deformation in a Complex Flow Generated by a Rotor-Stator Device'. submitted to Chemical Engineering Science.

Elman, H. C., Silverster, D. J. & Wathen, A. J. (2005), *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, Oxford University Press.

Feigl, K., Kaufmann, S. F. M., Fischer, P. & Windhab, E. J. (2003), 'A Numerical Procedure for Calculating Droplet Deformation in Dispersing Flows and Experimental Verification', *Chemical Engineering Science* **58**, 2351–2363.

Feigl, K., Megias-Alguacil, D., Fischer, P. & Windhab, E. J. (2007), 'Simulation and Experiments of Droplet Deformation and Orientation in Simple Shear Flow with Surfactants', *Chemical Engineering Science* **52**, 3242–3258.

Ginzburg, I. & Wittum, G. (2001), 'Two-Phase Flows on Interface Refined Grids Modeled with VOF, Staggered Finite Volumes, and Spline Interpolants', *Journal of Computational Physics* **166**, 302–335.

Grace, H. P. (1982), ‘Dispersion Phenomena in High Viscosity Immiscible Fluid Systems and Application of Static Mixers as Dispersion Devices in Such Systems’, *Chemical Engineering Communications* **14**, 225–277.

Gueyffier, D., Lie, J., Nadim, A., Scardovelli, R. & Zaleski, S. (1999), ‘Volume-of-Fluid Interface Tracking with Smoothed Surface Stress Methods for Three-Dimensional Flows’, *Journal of Computational Physics* **152**, 423–456.

Guido, S. & Greco, F. (2001), ‘Drop Shape under Slow Steady Shear Flow and During Relaxation. Experimental Results and Comparison with Theory’, *Rheologica Acta* **40**, 176–184.

Guido, S., Greco, F. & Villone, M. (1999), ‘Experimental Determination of Drop Shape in Slow Steady Shear Flow’, *Journal of Colloid and Interface Science* **219**, 298–306.

Guido, S., Minale, M. & Maffetone, P. L. (2000), ‘Drop Shape Dynamics Under Shear Flow Reversal’, *Journal of Rheology* **44**, 1385–1399.

Guido, S. & Villone, M. (1998), ‘Three-Dimensional Shape of a Drop under Simple Shear Flow’, *Journal of Rheology.* **42**, 395–415.

Harvie, D. J. E. & Fletcher, D. F. (2000), ‘A New Volume of Fluid Advection Algorithm: The Stream Scheme’, *Journal of Computational Physics* **162**, 1–32.

Harvie, D. J. E. & Fletcher, D. F. (2001), ‘A New Volume of Fluid Advection Algorithm: The Defined Donating Region Scheme’, *International Journal for Numerical Methods in Fluids* **35**, 151–172.

Henriksen, P. (1991), Simulation of Ultra Filtration, PhD thesis, Department of Chemical Engineering, Technical University of Denmark.

Hinch, E. J. & Acrivos, A. (1979), ‘Steady Long Slender Droplets in a Two-Dimensional Straining Motion’, *Journal of Fluid Mechanics* **91**, 404–414.

Hinch, E. J. & Acrivos, A. (1980), ‘Long Slender Drops in a Simple Shear Flow’, *Journal of Fluid Mechanics* **98**, 305–328.

Hirt, C. W. & Nichols, B. D. (1981), ‘Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries’, *Journal of Computational Physics* **39**, 201–225.

Kaufmann, S. F. M. (2002), Experimentelle und numerische Untersuchungen von Tropfendispergiervorgängen in komplexen laminaren Strömungsfeldern, PhD thesis, ETH, Zürich.

Kennedy, M. R., Pozrikidis, C. & Skalak, R. (1994), ‘Motion and Deformation of Liquid Drops, and the Rheology of Dilute Emulsions in Simple Shear Flow’, *Computers and Fluids* **23**, 251–278.

Khismatullin, D., Renardy, Y. & Renardy, M. (2006), 'Development and Implementation of VOF-PROST for 3D Viscoelastic Liquid-Liquid Simulations', *Journal of Non-Newtonian Fluid Mechanics* **140**, 120–131.

Kikpka, R. J. (2004), Numerical Simulation of a Dispersing Device in the Arbitrary Lagrangian-Eulerian Formulation, Master's thesis, Michigan Technological University.

Kwak, S. & Pozrikidis, C. (1998), 'Adaptive Triangulation of Evolving Closed, or Open Surfaces by the Advancing-Front Method', *Journal of Computational Physics* **145**, 61–88.

Lafaurie, B., Nardone, C., Scardovelli, R., Zaleski, S. & Zanetti, G. (1994), 'Modelling Merging and Fragmentation in Multiphase Flows with SURFER', *Journal of Computational Physics* **113**, 134–147.

Li, J., Renardy, Y. & Renardy, M. (2000), 'Numerical Simulation of Breakup of a Viscous Drop in Simple Shear Flow through a Volume-of-Fluid Method', *Physics of Fluids* **12**, 269–282.

Li, S. & Li, H. (2004), 'A Novel Approach of Divergence-Free Reconstruction for Adaptive Mesh Refinement', *Journal of Computational Physic* **199**, 1–15.

Loewenberg, M. & Hinch, E. J. (1996), 'Numerical Simulation of a Concentrated Emulsion in Shear Flow', *Journal of Fluid Mechanics* **321**, 395–419.

Loewenberg, M. & Hinch, E. J. (1997), 'Collision of Two Deformable Drops in Shear Flow', *Journal of Fluid Mechanics* **338**, 299–315.

Mashayek, F. & Ashgriz, N. (1995), 'A Hybrid Finite-Element-Volume-of-Fluid Method for Simulating Free Surface Flows and Interfaces', *International Journal of Numerical Methods in Fluids* **20**, 1363–1380.

Megias-Alguacil, D., Feigl, K., Dressler, M., Fischer, P. & Windhab, E. J. (2005), 'Droplet Deformation under Simple Shear Investigated by Experiment, Numerical Simulation and Modeling ', *Journal of Non-Newtonian Fluid Mechanics* **126**, 153–161.

Noh, W. F. & Woodward, P. R. (1976), SLIC (Simple Line Interface Method), *in* 'Lecture Notes in Phys.', Vol. 59, Springer-Verlag, Berlin/New York, p. 330.

Pantuso, D., Jiang, L., Shankar, S. & Skokov, S. (2003), 'A FEM/VOF Hybrid Formulation for Underfill Encapsulation Modeling', *Computers and Structures* **81**, 879–885.

Parker, B. J. & Youngs, D. L. (1992), Two and Three Dimensional Eulerian Simulation of Fluid Flow with Material Interfaces, Technical report, UK Atomic Weapons Establishment, Aldermaston, Berkshire.

Pilliod, J. E. J. & Puckett, E. G. (2004), 'Second-Order Accurate Volume-of-Fluid Algorithms for Tracking Material Interfaces', *Journal of Computational Physics* **199**, 465–502.

Pozrikidis, C. (2001), 'Numerical Investigation of the Effect of Surfactants on the Stability and Rheology of Emulsions and Foam', *Journal of Engineering Mathematics* **41**, 237–258.

Puckett, E. G. (1991), A Volume-of-Fluid Interface Tracking Algorithm with Applications to Computing Schock Wave Refraction, *in* H. Dwyer, ed., 'Proceedings of the Fourth International Symbosium on Computational Fluid Dynamics', Davis, Ca, pp. 933–938.

Rallison, J. M. (1980), 'Note on the Time-Dependent Deformation of a Viscous Drop which is Almost Spherical', *Journal of Fluid Mechanics* **98**, 625–633.

Rallison, J. M. (1981), 'A Numerical Study of the Deformation and Burst of a Viscous Drop in General Shear', *Journal of Fluid Mechanics* **109**, 465–482.

Rallison, J. M. (1984), 'The Deformation of Small Viscous Drops and Bubbles in Shear Flows', *Annual review of fluid mechanics* **16**, 45–66.

Rallison, J. M. & Acrivos, A. (1978), 'A Numerical Study of the Deformation and Burst of a Viscous Drop in an Extensional flow', *Journal of Fluid Mechanics* **89**, 191–200.

Renardy, Y. (2007), 'The Effects of Confinement and Inertia on the Production of Droplets', *Rheologica Acta* **46**, 521–529.

Renardy, Y., Cristini, V. & Li, J. (2002), 'Drop Fragment Distributions under Shear with Inertia', *International Journal of Multiphase Flow* **28**, 1125–1147.

Renardy, Y. & Renardy, M. (2002), 'PROST: A Parabolic Reconstruction of Surface Tension for the Volume-of-Fluid Method', *Journal of Computational Physics* **183**, 400–421.

Renardy, Y. Y., Renardy, M. & Cristini, V. (2002), 'A new Volume-of-Fluid Formulation for Surfactants and Simulations of Drop Deformation under Shear at a Low Viscosity Ratio', *European Journal of Mechanics* **21**, 49–59.

Rider, W. J. & Kothe, D. B. (1998), 'Reconstructing Volume Tracking', *Journal of Computational Physics* **141**, 112–152.

Rudman, M. (1998), 'A Volume-Tracking Method for Incompressible Multifluid Flows with Large Density Variations', *International Journal for Numerical Methods in Fluids* **28**, 357–378.

Rumscheidt, F. D. & Mason, S. G. (1961), 'Particle Motions in Sheared Suspensions XII. Deformation and Burst of Fluid Drops in Shear and Hyperbolic Flow', *Journal of Colloid Science* **16**, 238–261.

Scardovelli, R. & Zaleski, S. (1999), 'Direct Numerical Simulation of Free-Surface and Interfacial Flow', *Annual review of fluid mechanics* **31**, 567–603.

Scardovelli, R. & Zaleski, S. (2000), 'Analytical Relations Connecting Linear Interfaces and Volume Fractions in Rectangular Grids', *Journal of Computational Physics* **164**, 228–237.

Scardovelli, R. & Zaleski, S. (2003), 'Interface Reconstruction with Least-Square Fit and Split Eulerian-Lagrangian Advection', *International Journal for Numerical Methods in Fluids* **41**, 251–274.

Shenck, O. & Gärtner, K. (2004), 'Solving Unsymmetric Sparse Systems of Linear Equations with PARDISO', *Journal of Future Generation Computer Systems* **20**, 475–487.

Shenck, O. & Gärtner, K. (2006), 'On Fast Factorization Pivoting Methods for Symmetric Indefinite Systems', *Electronic Transactions on Numerical Analysis* **23**, 158–179.

Shenck, O., Gärtner, K. & Fichtner, W. (2000), 'Efficient Sparse LU Factorization with Left-Right Looking Strategy on Shared Memory Multiprocessors', *BIT* **40**, 158–176.

Shin, S. & Lee, W. I. (2000), 'Finite Element Analysis of Incompressible Viscous Flow with Moving Free Surface by Selective Volume of Fluid Method', *International Journal of Heat and Fluid Flow* **21**, 197–206.

Stone, H. A., Bentley, B. J. & Leal, L. G. (1986), 'An Experimental Study of Transient Effects in the Breakup of Viscous Drops', *Journal of Fluid Mechanics* **173**, 131–158.

Szabo, P. (1994), Simulation of Multiphase Flow, PhD thesis, Department of Chemical Engineering, Technical University of Denmark.

Taylor, G. I. (1932), 'The Viscosity of a Fluid Containing Small Drops of Another Fluid', *Proceedings of the Royal Society of London. Series A* **138**, 41–48.

Taylor, G. I. (1934), 'The Formation of Emulsions in Definable Fields of Flow', *Proceedings of the Royal Society of London. Series A* **146**, 501–523.

Taylor, G. I. (1964), Conical Free Surfaces and Fluid Interfaces, *in* 'Proceedings of the 11th International Congress of Applied Mechanics', Springer-Verlag, Munich, pp. 790–796.

Torza, S., Cox, R. G. & Mason, S. G. (1972), 'Particle Motions in Sheared Suspensions XXVII Transient and Steady Deformation and Burst of Liquid Drops', *Journal of Colloid and Interface Science* **38**, 395–411.

Wang, G. (2002), 'Finite Element Simulations of Free Surface Flows With Surface Tension in Complex Geometries', *Transactions of the ASME* **124**, 584–594.

Wu, J., Yo, S.-T. & Jiang, B.-N. (1998), 'Simulation of Two-Fluid Flows by The Least-Squares Finite Element Method using a Continuum Surface Tension Model', *International Journal for Numerical Methods in Engineering* **42**, 583–600.

Yon, S. & Pozrikidis, C. (1998), 'A Finite-Volume/Boundary-Element Method for Flow past Interfaces in the Presence of Surfactans, with Application to Shear Flow past a Viscous Drop', *Computers and Fluids* **27**, 879–902.

Youngs, D. L. (1982), Time-Dependent Multi-Material Flow with Large Fluid Distortion, *in* 'Numerical Methods for Fluid Dynamics', Academic Press, New York, p. 273.

Zalesak, S. T. (1979), 'Fully Multidemensional Flux-Corrected Transport Algorithms for Fluids', *Journal of Computational Physics* **31**, 335–362.

Zienkiewicz, O. C. & Taylor, R. L. (2000), *The Finite Element Method Volume 1: The Basis*, 5th. edn, Butterworth-Heinemann.

Appendix A

Derivation of the discrete equations

The total molecular stress tensor $\boldsymbol{\pi}$ is given by $\boldsymbol{\pi} = \boldsymbol{\tau} + p\boldsymbol{\delta}$ where

$$\boldsymbol{\tau} = -\mu \left[\begin{array}{ccc} 2\frac{\partial v_1}{\partial x_1} & \frac{\partial v_1}{\partial x_2} + \frac{\partial v_2}{\partial x1} & \frac{\partial v_1}{\partial x_3} + \frac{\partial v_3}{\partial x1} \\ \frac{\partial v_1}{\partial x_2} + \frac{\partial v_2}{\partial x1} & 2\frac{\partial v_2}{\partial x_2} & \frac{\partial v_2}{\partial x_3} + \frac{\partial v_3}{\partial x2} \\ \frac{\partial v_1}{\partial x_3} + \frac{\partial v_3}{\partial x1} & \frac{\partial v_2}{\partial x_3} + \frac{\partial v_3}{\partial x2} & 2\frac{\partial v_3}{\partial x_3} \end{array} \right] \tag{A.1}$$

In the discrete case the spatial derivatives of the velocity components are approximated by the finite element shape functions. Hence the element-wise volume integral over the total molecular stress tensor and the external force term becomes:

$$\sum_{n=1}^{N_e} \int\limits_{\Omega_n} \hat{\boldsymbol{\pi}} \cdot \nabla N_i^v dV + \sum_{k=1}^{N_e} \int\limits_{\Omega_n} N_i^v \boldsymbol{f} dV =$$

x1-direction:

$$\sum_{n=1}^{N_e} \int\limits_{\Omega_n} \left\{ -\mu \left[2\frac{\partial N_{i(I,n)}}{\partial x_1} \sum_{j=1}^{20} \frac{\partial N_j}{\partial x_1} \hat{v}_{1,j} + \frac{\partial N_{i(I,n)}}{\partial x_2} \left(\sum_{j=1}^{20} \frac{\partial N_j}{\partial x_2} \hat{v}_{1,j} + \sum_{j=1}^{20} \frac{\partial N_j}{\partial x_1} \hat{v}_{2,j} \right) \right.\right.$$

$$\left.\left. + \frac{\partial N_{i(I,n)}}{\partial x_3} \left(\sum_{j=1}^{20} \frac{\partial N_j}{\partial x_3} \hat{v}_{1,j} + \sum_{j=1}^{20} \frac{\partial N_j}{\partial x_1} \hat{v}_{3,j} \right) \right] + \frac{\partial N_{i(I,n)}}{\partial x_1} \hat{p}_n + N_{i(I,n)}^v f_1 \right\} dV =$$

$$\sum_{n=1}^{N_e} \left(-\mu \sum_{j=1}^{20} \hat{v}_{1,j} \int\limits_{\Omega_n} 2\frac{\partial N_{i(I,n)}}{\partial x_1} \frac{\partial N_j}{\partial x_1} + \frac{\partial N_{i(I,n)}}{\partial x_2} \frac{\partial N_j}{\partial x_2} + \frac{\partial N_{i(I,n)}}{\partial x_3} \frac{\partial N_j}{\partial x_3} dV \right.$$

$$\left. -\mu \sum_{j=1}^{20} \hat{v}_{2,j} \int\limits_{\Omega_n} \frac{\partial N_{i(I,n)}}{\partial x_2} \frac{\partial N_j}{\partial x_1} dV - \mu \sum_{j=1}^{20} \hat{v}_{3,j} \int\limits_{\Omega_n} \frac{\partial N_{i(I,n)}}{\partial x_3} \frac{\partial N_j}{\partial x_1} dV + \hat{p}_n \int\limits_{\Omega_n} \frac{\partial N_{i(I,n)}}{\partial x_1} dV + \int\limits_{\Omega_n} N_{i(I,n)} f_1 dV \right) =$$

$$\sum_{n=1}^{N_e} \left(\sum_{k1=1}^{3} A_{1k1}^{i(I,n)} + C_1^{i(I,n)} + f_1^{i(I,n)} \right), \quad I = 1, \ldots, N_n$$

x2-direction:

$$\sum_{n=1}^{N_e}\int\limits_{\Omega_n}\left\{-\mu\left[\frac{\partial N_{i(I,n)}}{\partial x_1}\left(\sum_{j=1}^{20}\frac{\partial N_j}{\partial x_2}\hat{v}_{1,j}+\sum_{j=1}^{20}\frac{\partial N_j}{\partial x_1}\hat{v}_{2,j}\right)+2\frac{\partial N_{i(I,n)}}{\partial x_2}\sum_{j=1}^{20}\frac{\partial N_j}{\partial x_2}\hat{v}_{2,j}\right.\right.$$

$$\left.\left.+\frac{\partial N_{i(I,n)}}{\partial x_3}\left(\sum_{j=1}^{20}\frac{\partial N_j}{\partial x_3}\hat{v}_{2,j}+\sum_{j=1}^{20}\frac{\partial N_j}{\partial x_2}\hat{v}_{3,j}\right)\right]+\frac{\partial N_{i(I,n)}}{\partial x_2}\hat{p}_n+N^v_{i(I,n)}f_2\right\}dV=$$

$$\sum_{n=1}^{N_e}\left(-\mu\sum_{j=1}^{20}\hat{v}_{1,j}\int\limits_{\Omega_n}\frac{\partial N_{i(I,n)}}{\partial x_1}\frac{\partial N_j}{\partial x_2}dV-\mu\sum_{j=1}^{20}\hat{v}_{2,j}\int\limits_{\Omega_n}2\frac{\partial N_{i(I,n)}}{\partial x_2}\frac{\partial N_j}{\partial x_2}+\frac{\partial N_{i(I,n)}}{\partial x_1}\frac{\partial N_j}{\partial x_1}+\frac{\partial N_{i(I,n)}}{\partial x_3}\frac{\partial N_j}{\partial x_3}dV\right.$$

$$\left.-\mu\sum_{j=1}^{20}\hat{v}_{3,j}\int\limits_{\Omega_n}\frac{\partial N_{i(I,n)}}{\partial x_3}\frac{\partial N_j}{\partial x_2}dV+\hat{p}_n\int\limits_{\Omega_n}\frac{\partial N_{i(I,n)}}{\partial x_2}dV+\int\limits_{\Omega_n}N_{i(I,n)}f_2dV\right)=$$

$$\sum_{n=1}^{N_e}\left(\sum_{k1=1}^{3}A_{2k1}^{i(I,n)}+C_2^{i(I,n)}+f_2^{i(I,n)}\right),\quad I=1,\ldots,N_n$$

x3-direction:

$$\sum_{n=1}^{N_e}\int\limits_{\Omega_n}\left\{-\mu\left[\frac{\partial N_{i(I,n)}}{\partial x_1}\left(\sum_{j=1}^{20}\frac{\partial N_j}{\partial x_3}\hat{v}_{1,j}+\sum_{j=1}^{20}\frac{\partial N_j}{\partial x_1}\hat{v}_{3,j}\right)+\frac{\partial N_{i(I,n)}}{\partial x_2}\left(\sum_{j=1}^{20}\frac{\partial N_j}{\partial x_3}\hat{v}_{2,j}+\sum_{j=1}^{20}\frac{\partial N_j}{\partial x_2}\hat{v}_{3,j}\right)\right.\right.$$

$$\left.\left.+2\frac{\partial N_{i(I,n)}}{\partial x_3}\sum_{j=1}^{20}\frac{\partial N_j}{\partial x_3}\hat{v}_{3,j}\right]+\frac{\partial N_{i(I,n)}}{\partial x_3}\hat{p}_n+N^v_{i(I,n)}f_3\right\}dV=$$

$$\sum_{n=1}^{N_e}\left(-\mu\sum_{j=1}^{20}\hat{v}_{1,j}\int\limits_{\Omega_n}\frac{\partial N_{i(I,n)}}{\partial x_1}\frac{\partial N_j}{\partial x_3}dV-\mu\sum_{j=1}^{20}\hat{v}_{2,j}\int\limits_{\Omega_n}\frac{\partial N_{i(I,n)}}{\partial x_2}\frac{\partial N_j}{\partial x_3}dV\right.$$

$$\left.-\mu\sum_{j=1}^{20}\hat{v}_{3,j}\int\limits_{\Omega_n}\frac{\partial N_{i(I,n)}}{\partial x_1}\frac{\partial N_j}{\partial x_1}+\frac{\partial N_{i(I,n)}}{\partial x_2}\frac{\partial N_j}{\partial x_2}+2\frac{\partial N_{i(I,n)}}{\partial x_3}\frac{\partial N_j}{\partial x_3}dV+\hat{p}_n\int\limits_{\Omega_n}\frac{\partial N_{i(I,n)}}{\partial x_3}dV+\int\limits_{\Omega_n}N_{i(I,n)}f_3dV\right)=$$

$$\sum_{n=1}^{N_e}\left(\sum_{k1=1}^{3}A_{3k1}^{i(I,n)}+C_3^{i(I,n)}+f_3^{i(I,n)}\right),\quad I=1,\ldots,N_n$$

In the above equations the subscripts i and j refer to local nodes on element n. It is noted that given a global node I located on element n there is a unique corresponding local node i and vice versa. The similar is of course also true for a local and global node j and J respectively. From the equations deduced above one sees that there are $3N_n$ equations associated with the momentum balance. Furthermore since there are N_n velocity nodes in the domain there are also $3N_n$ velocity unknowns.

Appendix B

The CSF interfacial tension model

In the procedure described by Brackbill et al. (1992) (The CSF model) interfacial tension is included through the volume force:

$$\boldsymbol{F}_s = [\kappa\sigma\hat{\boldsymbol{n}}]\delta_s$$

where κ is the interface curvature, $\hat{\boldsymbol{n}}$ is the interface normal and δ_s is the interface delta function. Thus the jump in the total molecular stress tensor across the interface can be written as:

$$\int\limits_{\Gamma_{12}} \phi^v[\boldsymbol{n}_1 \cdot (\boldsymbol{\pi}_1 - \boldsymbol{\pi}_2)]dS = \int\limits_{\Omega} \phi^v[\kappa\sigma\hat{\boldsymbol{n}}]\delta_s(\hat{\boldsymbol{n}}(\boldsymbol{x} - \boldsymbol{x}_s))dV$$

The interface curvature κ is given by:

$$\kappa = -\nabla \cdot \hat{\boldsymbol{n}} = \frac{1}{|\boldsymbol{n}|}\left[\left(\frac{\boldsymbol{n}}{|\boldsymbol{n}|} \cdot \nabla\right)|\boldsymbol{n}| - \nabla \cdot \boldsymbol{n}\right] \tag{B.1}$$

The interface normal is calculated either as the gradient of the original F-field or as the gradient of a smoothed F-field $\tilde{F}$. Similarly δ_s is defined either by

$$\delta_s = \left|\nabla\tilde{F}\right|$$

or

$$\delta_s = |\nabla F|$$

However, as discussed in the main text (section 2.5.1) Brackbill et al. (1992) argue that the latter method is better. It is noted that in our implementation no smoothing is carried out. To summarize we calculate the force F_s as:

$$\boldsymbol{F}_s = \sigma\kappa\boldsymbol{n} = \sigma\frac{1}{|\nabla F|}\left[\left(\frac{\nabla F}{|\nabla F|} \cdot \nabla\right)|\nabla F| - \nabla \cdot \nabla F\right]\nabla F \tag{B.2}$$

Discretization of the CSF model
In order to obtain the discrete derivatives of F we use a procedure where first ∇F is

calculated at the linear nodes on the interface VOF cells. This is done using the following way:

First the cell under consideration (active cell) and all its neighbouring cells are identified which leads to a $3 \times 3 \times 3$ block with the active cell in the centre. A ghost mesh is then overlaid this block of cells such that the vertices of the ghost mesh elements are positioned in the geometrical centre of each VOF cell. This leads to a ghost mesh with size $2 \times 2 \times 2$ (see Figure 2.2 in section 2.4.2 for reference). A mapping from the cartesian coordinate system to the local curvilinear system ξ, η, ζ is then made using tri-linear finite element shape functions associated with the ghost mesh. The F-field can therefore be interpolated everywhere within a ghost element as (for the 3D case):

$$F(\boldsymbol{x}(\xi,\eta,\zeta)) = \sum_{i=1}^{8} L_i(\xi,\eta,\zeta) F_i \tag{B.3}$$

And similarly we can determine the gradient of F within the ghost element as

$$\left.\frac{\partial F}{\partial x_k}\right|_{(\xi,\eta,\zeta)} = \sum_{i=1}^{8} \left.\frac{\partial L_i}{\partial x_k}\right|_{(\xi,\eta,\zeta)} F_i, \quad k = 1, \ldots 3 \tag{B.4}$$

The derivatives of the linear shape functions with respect to the global coordinates $\frac{\partial L_i}{\partial x_k}$ are calculated according to

$$\begin{bmatrix} \left.\frac{\partial L_i}{\partial x_1}\right|_{(\xi,\eta,\zeta)} \\ \left.\frac{\partial L_i}{\partial x_2}\right|_{(\xi,\eta,\zeta)} \\ \left.\frac{\partial L_i}{\partial x_3}\right|_{(\xi,\eta,\zeta)} \end{bmatrix} = \mathbf{J}^{-1}\big|_{(\xi,\eta,\zeta)} \begin{bmatrix} \left.\frac{\partial L_i}{\partial \xi}\right|_{(\xi,\eta,\zeta)} \\ \left.\frac{\partial L_i}{\partial \eta}\right|_{(\xi,\eta,\zeta)} \\ \left.\frac{\partial L_i}{\partial \zeta}\right|_{(\xi,\eta,\zeta)} \end{bmatrix} \tag{B.5}$$

where the jacobian matrix is given by equation 2.2-19 but with the quadratic shape functions N_i interchanged by the linear shape functions L_i (and $i = 1, \ldots, 8$).

In order to evaluate ∇F according to equation B.4 we first calculate the position of the vertex point in the local coordinate system as described in section 2.4.2 (cf. also Figure 2.3). Hereafter equation B.4 can be used directly. The interface normal ($\boldsymbol{n} = \nabla F$) can now be estimated everywhere within the VOF cell m using tri-linear interpolation functions associated with the VOF cell, i.e.:

$$\left.\frac{\partial F}{\partial x_k}\right|_{\boldsymbol{x}_m} = \sum_{i=1}^{8} L_i(\boldsymbol{x}_m(\xi,\eta,\zeta))(\nabla F)_{k,i}, \quad k = 1, \ldots 3 \tag{B.6}$$

where $(\nabla F)_{k,i}$ is k component of the gradient of F on local node i.

The divergence of ∇F everywhere within the active VOF cell is calculated as:

$$\nabla \cdot (\nabla F)|_{(\xi,\eta,\zeta)} = \sum_{k=1}^{3} \sum_{i=1}^{8} \left.\frac{\partial L_i}{\partial x_k}\right|_{(\xi,\eta,\zeta)} (\nabla F)_i \tag{B.7}$$

where $\frac{\partial L_i}{\partial x_k}$ now is associated with the VOF cell (and not the ghost cell).

We now denote the curvature calculated according to equation B.2 with the discrete derivatives of F based on equation B.6 and B.7 as $\hat{\kappa}$. Similarly the discrete interface normal is denoted $\boldsymbol{n}_s$ $(= \nabla F)$. The discretized version of equation B.2 then becomes:

$$\sum_{n=1}^{N_e} \sum_{k=1}^{3} \left(\int\limits_{\Omega_n} N^v_{i(I,n)} \sigma \hat{\kappa}^n \boldsymbol{n}^n_{s,k} dV \right) \boldsymbol{\delta}_k, \quad I = 1, \ldots N_n \tag{B.8}$$

where $I = 1, \ldots, N_n$ is the number of global finite element nodes. Thus equation I will get a contribution from the interfacial tension if global node I it is connected to elements in which $\boldsymbol{n}_s$ is non-zero.

Appendix C

Drop break-up

Results from rotor-stator experiments which resulted in drop break-up. The figures show the distance from the center of the cylinders to the inner and outer walls (solid curves) and the drop center (circles). Also shown are drop images at selected experimental times.

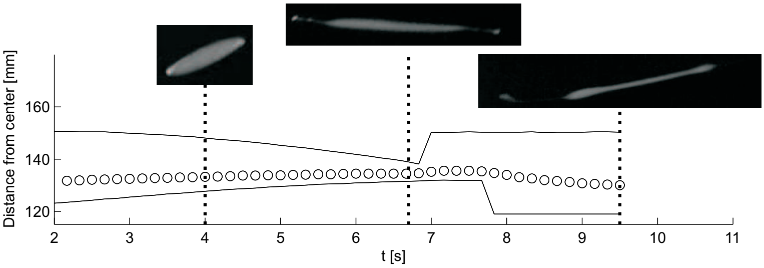


Figure C.1: Drop break-up experiment with AK50 ($\lambda = 0.0152$, $\sigma = 5.1$), $\dot{\omega}_i = 0.1242s^{-1}$, $\dot{\omega}_o = 0.1176s^{-1}$, $r_0 = 590\mu m$

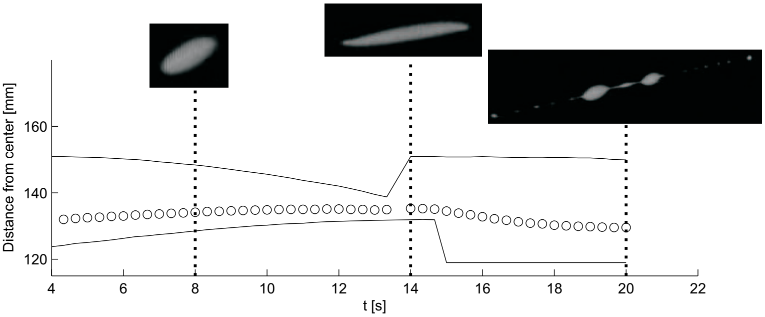


Figure C.2: Drop break-up experiment with AK100 ($\lambda = 0.0303$, $\sigma = 5.2$), $\dot{\omega}_i = 0.0621s^{-1}$, $\dot{\omega}_o = 0.0588s^{-1}$ $r_0 = 570\mu m$

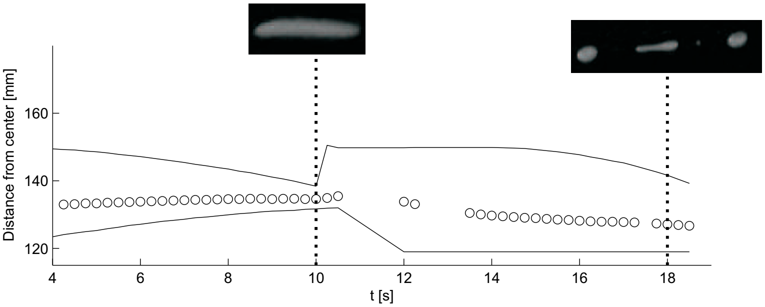


Figure C.3: Drop break-up experiment with AK500 ($\lambda = 0.152$, $\sigma = 5.3$), $\dot{\omega}_i = 0.0828s^{-1}$, $\dot{\omega}_o = 0.0784s^{-1}$ $r_0 = 479\mu m$

Appendix D

The FE-VOF program

Here an explanation of the implemented algorithms is given. This includes a brief description of all the global variables and parameters and a description of the various subroutines. Furthermore the full program, which is written in Fortran 90/95, consists of 6 modules. These modules are:

Module name		**Description**
parameters.f90	:	Definition of parameters.
globals.f90	:	Definition of global variables/arrays.
mainfem.f90	:	Main Finite Element (FE) code. Includes all subroutines used for setting up the coefficient matrix. Furthermore the main time stepping loop resides in the program part of this module where calls are made to the PARDISO solver and the VOF routines.
femmesh.f90	:	Routines for setting up the finite element mesh used for the simulations in this project. Here boundary conditions are also specified.
vofmesh.f90	:	Routines for setting up the Volume of Fluid (VOF) mesh and for carrying out the advection procedures. Furthermore the subroutines associated with the interfacial tension calculations also reside in this module.
vofnormals.f90	:	Some routines used for calculating interface normals.

Here we will start by listing parameters and global variables.

D.1 List of variables and parameters

All non-integers are specified as double precision using `implicit double precision(a-h, o-z)`, i.e. all arrays are double precision except those whose name starts with `i,j,k,l,m` or `n` which are integers. It is noted that in Fortran there is no distinction between upper-case and lower-case characters.

Variable		**Description**
`alphavof(i)`	:	The value of α from the PLIC reconstruction in VOF cell `i` (cf. equation 2.4-12).
`ALx`	:	Half length of the computational box in the x_1-direction (cf. Figure 3.1).
`ALy`	:	Half length of the computational box in the x_2-direction (cf. Figure 3.1).
`ALz`	:	Half length of the computational box in the x_3-direction (cf. Figure 3.1).
`AR(*)`	:	Coefficient matrix stored in compressed sparse row format (cf. section 2.2.7).
`CFL`	:	A CFL number specified by the user (determines the variable time step size).
`cntroid(i,k)`	:	Coordinate component `k` of the geometrical center in VOF cell `i`.
`cntrlocnormal(i,k)`	:	Interface normal component `k` in local coordinates in center of VOF cell `i`. Used for the PLIC reconstruction.
`dFnode(i,k)`	:	Component `k` of the gradient of the F-field on linear node `i` of the VOF-mesh. Used only when the CSF interfacial tension model is applied.
`Djac(i,m)`	:	Determinant of the Jacobian matrix at integration point `m` in finite element `i` (cf. equation 2.2-21).
`dLdglob_vof_node(i,iloc,jloc,k)`	:	Global derivative of the linear shape function associated with local ghost mesh node `jloc` and evaluated at local node `iloc` on VOF cell `i`. Index `k` is the differentiation direction. Used for calculating the values in dFnode.

Variable		**Description**
`dLdglob(iloc,m,i,k)`	:	Value at integration point `m` of component `k` of the global derivative of the linear shape function associated with local node `iloc`. `i`$=1,\ldots,N_1N_2N_3$ is the local VOF cell number in a given finite element.
`dLdloc(iloc,m,k)`	:	Component `k` (local coordinate) of the local derivative of the linear shape function associated with local node `iloc` evaluated at integration point `m`.
`dNdeta(m,iloc)`	:	Local derivative with respect to η of the quadratic shape function associated with local finite element node `iloc` evaluated at integration point `m`.
`dNdface(iloc,m,iface,k)`	:	Local derivative with respect to coordinate `k` of the 2D quadratic shape function associated with local node `iloc`. `m` is the integration point where the derivative is evaluated and `iface` is the element face under consideration (cf. equation 2.4-29).
`dNdglob(i,m,iloc,k)`	:	Component `k` of the global derivative of the quadratic shape function associated with local node `iloc` on local VOF cell `i`. The value is evaluated on integration point `m` (c.f. equation 2.5-44).
`dNdlocvof(i,m,iloc,k)`	:	Component `k` of the local derivative of the quadratic shape function associated with local node `iloc` on local VOF cell `i`. The value is evaluated on integration point m (used for evaluating `dNdglob`).
`dNdxi(m,iloc)`	:	Local derivative with respect to ξ of the quadratic shape function associated with with local finite element node `iloc` at integration point `m`.
`dNdzeta(m,iloc)`	:	Local derivative with respect to ζ of the quadratic shape function associated with with local finite element node `iloc` at integration point `m`.

Variable		Description
`dNdx(iloc,m,i)`	:	Global derivative with respect to x_1 of the quadratic shape function associated with local node `iloc` on finite element `i`. Evaluated at integration point `m`.
`dNdy(iloc,m,i)`	:	Global derivative with respect to x_2 of the quadratic shape function associated with local node `iloc` on finite element `i`. Evaluated at integration point `m`.
`dNdz(iloc,m,i)`	:	Global derivative with respect to x_3 of the quadratic shape function associated with local node `iloc` on finite element `i`. Evaluated at integration point `m`.
`elongrate(*)`	:	Array for storing elongation rates which are read in from a data file (only relevant for externally determined time dependent elongation rates).
`ematrixAR(iloc,jloc,k1,i)`	:	Local element matrices associated with the velocity terms. Used for building the coefficient matrix. `iloc` and `jloc` are local node indices, `k1`$= 1,\ldots,9$ corresponds to a given sub-system and `i` is the finite element number.
`ematrixp(iloc,k1,i)`	:	Local element matrices associated with the pressure term. Used for building the coefficient matrix. `iloc` is a local node index, `k1`$= 1,\ldots,3$ corresponds to a given subsystem and `i` is the finite element number.
`eta(*)`	:	Array with local η-coordinates used for setting up the VOF mesh (i.e. in association with the 2-level procedure).
`etaInt(m)`	:	Local coordinate η for integration point `m`.
`F(*)`	:	The F-field.
`fluxmax`	:	Maximum flux on the finite element mesh.
`Fnode(*)`	:	The F-field interpolated onto the VOF-cell nodes.
`gacc(k)`	:	Gravitational acceleration in direction `k`.
`H(m)`	:	Integration weight at integration point `m`.
`h_fine(k)`	:	VOF cell side length in the fine mesh region in direction `k`.

Variable		Description
IA(*)	:	Row index pointer array for the sparse coefficient matrix (cf. section 2.2.7).
itensionmet	:	Input flag. If itensionmet=1 then the CSS method is used else if itensionmet=2 the CSF method is used.
iwritedata	:	Number of time steps between writing output data to files.
JA(*)	:	Column index pointer array for the sparse coefficient matrix (cf. section 2.2.7).
nactive	:	Number of active VOF cells (i.e. number of interface cells plus nearest neighbours).
nactiveinterface(*)	:	Pointer array which points to the interface VOF cells.
nactiveinterfacefe(*)	:	Flag array for specifying active finite elements.
nactivevof(*)	:	Pointer array which points to active VOF cells (interface cells plus nearest neighbours).
NAR	:	Number of non-zero entries in coefficient matrix AR(*).
nb(*,*)	:	Array for specifying boundary conditions.
nbcfile	:	Flag. If nbcfile=1 then shear and elongation rates are read in from file else if nbcfile=0 the specified values of G and a (α) are used.
nbcnodes(*,k)	:	Array which points to boundary nodes for which velocity component k is specified.
nbtot(k)	:	Total number of specified boundary nodes in direction k.
ncoordination_table(i,iloc)	:	Global VOF cell numbers for the $3 \times 3 \times 3$ block of cells with cell i in the center. Index iloc correspond to the local numbering in the block of cells.
Nefe	:	Total number of finite elements.
Nevof	:	Total number of VOF cells.
Nfe	:	Total number of velocity nodes.
NFE2VOF(i,iloc)	:	Global VOF cell number for local VOF cell iloc in finite element i.
NoInt	:	Number of integration points used in the flow solver (paramater). NoInt=27 or 8.

Variable		Description
ninterface	:	Number of interface VOF cells.
Nnodes	:	Total number of velocity degrees of freedom.
nobcnodes(k)	:	Total number of specified velocities for component k.
nodalcoordination(*,*)	:	Pointer array associated with array ncoordination_table.
nodenumFe(i,n)	:	Array used for setting up the sparse coefficient matrix. Here i is a global node. Then the finite elements which are connected to node i are given by n= $1, \dots, 8$ while the local number of node i in each of the connected finite element is given by n= $9, \dots, 16$. Finally n=17 is the total number of elements connected to node i.
np(*)	:	Array used for shifting node numbering when periodic boundary conditions are applied.
numFe(i,iloc)	:	Finite element mesh table. Global node number of local node iloc on finite elment i.
numFe2(*,*)	:	Modified Finite element table with specified boundary nodes exluded. Used for setting up the sparse coefficient matrix.
numVOF(i,iloc)	:	VOF mesh table. Global node number of local (linear) node iloc (= $1, \dots, 8$) on VOF cell i.
numVOFFEM(i,iloc)	:	Local VOF mesh table. VOF cell node number within a finite element (sub-mesh). Index iloc (= $1, \dots, 20$) is the local quadratic node number on local VOF cell i.
NVOF2FE(i)	:	Table which specifies to which finite element VOF cell i belongs.
NVx, NVy, NVz	:	Number of VOF cells in each direction in parent finite element (i.e. N_1, N_1, N_3 cf. section 2.4.2.4).
Nx, Ny, Nz	:	Number of finite elements in each direction for the calculation 'box' used in this project.
nzsym	:	If nzsym=1 then symmetry in the x_3-direction otherwise nzsym=2.
rho(i)	:	Average density in finite element i.

Variable		Description
`rhoc`	:	Density of continuous phase.
`rhod`	:	Density of drop phase.
`RHS(*)`	:	Right Hand Side vector.
`shapeN(m,iloc)`	:	Quadratic shape function associated with local node `iloc` and evaluated at integration point `m`.
`shapeNface(iloc,m,iface)`	:	Quadratic shape function associated with local node `iloc` and evaluated at integration points `m` $(= 1, \dots, 9)$ on finite element face `iface`. Used for calculating fluxes across finite element boundaries.
`shapeNface2(i,iloc,m,iface)`	:	Quadratic shape function associated with local VOF cell node `iloc` on local cell `i` and evaluated at integration points `m` $(= 1, \dots, 9)$ on VOF cell face `iface`. Used for calculating fluxes across VOF cell boundaries.
`shapevof(i,m,iloc)`	:	Quadratic shape function associated with local finite element node `iloc` and evaluated at integration point `m` in local VOF cell `i`. Used for calculating interfacial tension with the CSF model.
`shapeL(iloc,m)`	:	Linear shape function used for interpolating the interface normal from linear local VOF node `iloc` onto integration point `m`. Used for calculating interfacial tension with the CSF model.
`shearrate(*)`	:	Array for storing shear rates which are read in from a data file (only relevant for externally determined time dependent shear rates).
`sigma`	:	Interfacial tension coefficient.
`simtime`	:	Total simulation time.
`time(*)`	:	Array for storing the time variable associated with elongation and shear rates which are read in from a data file.
`tot_time`	:	Total simulation time.
`vel(*)`	:	Nodal velocity vector.
`VF0`	:	Initial volume of disperse phase.
`visc(i)`	:	Average viscosity in finite element `i`.

Variable		Description
`viscc`	:	Viscosity of continuous phase.
`viscd`	:	Viscosity of disperse phase.
`vofcoord(i,k)`	:	Component `k` of global coordinate for linear VOF node `i`.
`vofint(i,m,k)`	:	Coordinate `k` in local coordinates for integration point `m` in local VOF cell `i`.
`vofvolarea(i,n)`	:	Volume of VOF cell `i` when `n=1`. Index `n`= $2, \ldots, 7$ corresponds to area of face 1 to 6 on VOF cell `i`.
`vofweight(i,m)`	:	Integration weight `m` in local VOF cell `i`.
`xfe(i)`	:	x_1 component of the global coordinate for finite element node `i`.
`xi(*)`	:	Array with local ξ-coordinates used for setting up the VOF mesh (i.e. in association with the 2-level procedure).
`xiInt(m)`	:	Local coordinate ξ for integration point `m`.
`xint(*,*)`	:	Integration points on finite element faces.
`x0`	:	Position of the left boundary of the computational 'box'.
`yfe(i)`	:	x_2 component of the global coordinate for finite element node `i`.
`y0`	:	Position of the bottom boundary of the computational 'box'.
`zeta(*)`	:	Array with local ζ-coordinates used for setting up the VOF mesh (i.e. in association with the 2-level procedure).
`zetaInt(m)`	:	Local coordinate ζ for integration point `m`.
`zfe(i)`	:	x_3 component of the global coordinate for finite element node `i`.
`z0`	:	Position of the first boundary in the x_3-direction of the computational 'box'.
`w(m)`	:	Integration weight `m` in the parent finite element.

D.2 Subroutines in module femmesh.f90

The subroutines residing in the module femmesh.f90 are described below.

D.2.1 Femcoord

In this subroutine the coordinates for the quadratic velocity nodes associated with a regular finite element mesh are set up. The domain consists of a box given by `[-ALx+x0,ALx+x0]` × `[-ALy+y0,ALy+y0]` × `[-ALz+z0,(nzsym-1)ALx+x0]` where `ALx`, `ALy`, `ALz`, `x0`, `y0`, `z0`, `nzsym` are input variables as described in the variable list. The element density can be varied through the local input variables `Ncx`, `Ncy`, `Ncz`, `Nfx`, `Nfy`, `Nfz`, `ALfx`, `ALfy`, `ALfz`, `alphax`, `alphay`, `alphaz` where in each case the last character corresponds to direction x_1 x_2 and x_3 respectively. With reference to Figure 3.1 these variables have the following meaning:

`ALfx`, `ALfy`, `ALfz` : Size of the fine mesh region which is given by `[-ALfx+x0,ALfx+x0]` × `[-ALfy+y0,ALfy+y0]` × `[-ALfz+z0,(nzsym-1)ALfx+x0]`.

`Nfx`, `Nfy`, `Nfz` : Number of elements in the fine mesh region such that the element side lengths are $h_{fine,1}$ =`ALfx/Nfx`, $h_{fine,2}$=`ALfy/Nfy`, $h_{fine,3}$=`ALfz/Nfz`.

`Ncx`, `Ncy`, `Ncz` : Number of elements in each direction spanning from the fine region to the domain walls.

`alphax`, `alphay`, `alphaz` : These variables (0 <> 1) determine how the element size varies in the coarse mesh region such that when going from the wall to the fine mesh region the element side length in, say, the x_1 direction is decreased by a factor `alphax` for each element.

These mesh parameters are specified in the input file *meshparam.dat*. It is noted that if a different mesh is used the code needs to be modified, e.g. by reading in the coordinates from a data file. The output from the routine are the coordinate arrays `xfe(*)`, `yfe(*)` and `zfe(*)`.

D.2.2 LocalToGlobal

This routine is used for setting up the finite element mesh table in array `numFE(*,*)` such that the global node number for local node `iloc` on element `i` is given by `numFE(i,iloc)`. The inputs for this routine are the global mesh variables `Nx`, `Ny` and `Nz` while the output is `numFE(*,1:20)`. This routine is only applicable for regular meshes with local node numbering as indicated on Figure 2.2.

D.2.3 Boundarycond

Here Boundary Conditions (BC's) for the computatonal 'box' are specified. The type of possible BC's is discussed in section 2.2.8 and section 3.1. The BC's are specified by the user through the following local input variables:

`G` : Strain rate, i.e. $G = |\dot{\gamma}| + |\dot{\epsilon}|$ cf. section 3.1.

`a` : Relative magnitude of elongation in the flow, i.e. $\alpha = |\dot{\epsilon}|/G$ cf. section 3.1.

`vterm(i,k,j)` : Array used for specifying which terms (with sign) in equation 3.1-1 are included on each wall. Here `i`$= 1,\ldots,6$ is the wall number, `k`$= 1,\ldots,3$ is the velocity component and `j`=1 is the shear term while `j`=2 is the elongation term. The wall numbering is: 1 wall at `x0`, 2 wall at `2ALx+x0`, 3 wall at `z0`, 4 wall at `nzsym*ALz+z0`, 5 wall at `y0` and 6 wall at `2ALy+y0`. Values in `vterm(i,k,j)` can be either 1, -1 or 0.

`nbd(i,k)` : Array used for specifying on which walls velocity component `k` is specified. If, say, velocity component 1 is specified on wall 1 then `nbd(1,1)=1` otherwise it should be `0`.

`nbp(j)` : Array used for specifying periodic BC's in the x_1 (`j`=1) and x_3-direction (`j`=2). If periodicity in the x_1-direction then `nbp(1)`=1 otherwise `nbp(1)`=0 and similarly with the x_3-direction.

It is noted that the BC's are set up for planar flow problems where the velocity component in the x_3-direction of the applied flow is zero. Non-planar flows can be applied by making some minor changes to this subroutine.
In order to illustrate the use of the variables described above a simple example is given. Here we want to apply a simple shear flow where all velocity components are specified on walls 1, 2 and 5, 6 while on wall 3 and 4 velocity component v_1 and v_2 are unspecified and component v_3 is specified to zero. This means that `a=0` and `G` is the shear rate. The array `vterm(i,k,j)` is specified as:

`vterm(i,k,j)`:

	j=1			j=2		
i/k	1	2	3	1	2	3
1	1	0	0	1	-1	0
2	1	0	0	1	-1	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	1	0	0	1	-1	0
6	1	0	0	1	-1	0

This is the standard specification of `vterm` for planar flows when velocity components v_1 and v_2 are unspecified on boundary walls 3 and 4.

The array nbd(i,k) is specified as:

nbd(i,k):

i/k	1	2	3
1	1	1	1
2	1	1	1
3	0	0	1
4	0	0	1
5	1	1	1
6	1	1	1

Since domain periodicity is not specified nbp(1:2)=0. It is noted that using vterm and nbd as specified above it is possible to apply a planar flow any strength and consisting of any mixture of shear and elongation by changing G and a (i.e. G and α in equation 3.1-1). The outputs from the routine are the arrays nb(*,1:2), nbcnodes(*,1:3), nobcnodes(1:3) and np(*).

D.3 Subroutines in module mainfem.f90

The subroutines residing in the module mainfem.f90 are described below.

D.3.1 Integrationpoints

Integration points and weights are set up for the Gaussian quadrature. Input is the parameter NoInt which is the number of integration points used. NoInt=8 or NoInt=27. The outputs from the routine are the arrays H(1:NoInt) (weights), xiInt(1:NoInt), zetaInt(1:NoInt) and etaInt(1:NoInt) (local coordinates).

D.3.2 dNlocal

The quadratic finite element shape functions and the local derivatives thereof are evaluated at the integration points. Inputs to the routine are NoInt, xiInt(1:NoInt), zetaInt(1:NoInt) and etaInt(1:NoInt). The outputs are shapeN(1:NoInt,1:20), dNdxi(1:NoInt,1:20), dNdeta(1:NoInt,1:20) and dNdzeta(1:NoInt,1:20).

D.3.3 dNglobal

This routine sets up the global derivatives of the finite finite element shape functions at the integration points. Inputs are the nodal coordinate arrays xfe(*), yfe(*), zfe(*), the element mesh table numFE(*,1:20) and the local derivatives of the shape functions dNdxi(1:NoInt,1:20), dNdeta(1:NoInt,1:20) and dNdzeta(1:NoInt,1:20). Outputs are the global derivatives of the shape functions dNdx(1:20,1:NoInt,*), dNdy(1:20,1:NoInt,*), dNdz(1:20,1:NoInt,*) and the determinant of the Jacobian Djac(*,1:NoInt) (cf. section 2.2.1).

D.3.4 Ematrix

In this routine the element matrices are set up. The inputs are the mesh table numFE(*,1:20), the total number of finite elements Nefe, the number of integration points NoInt, the global derivatives of the shape functions dNdx(1:20,1:NoInt,*), dNdy(1:20,1:NoInt,*), dNdz(1:20,1:NoInt,*), the determinant of the Jacobian matrix Djac(*,1:NoInt) and the quadrature weights H(1:NoInt). The outputs are the arrays ematrixAR(1:20,1:20,1:9,*) and ematrixp(1:20,1:9,*). The entries in these arrays are contributions to the coefficients in the global equation system. It is noted that the viscosity does not enter in this routine and thus the routine only needs to be called once outside the time loop.

D.3.5 SetupElementTableBC

Here a new element table numFE2 is set up where boundary nodes which are specified by BC's are omitted. This new table is used for setting up the pointing arrays IA and JA. Inputs to the routine are the total number of finite elements Nefe, the total number of velocity nodes Nfe, the original mesh table numFe(*,1:20), the array which specifies velocity BC nodes nb(*,1:2) and the array which specifies periodic BC nodes np(*). The output is numFe2(*,41). Here numFe2(i(k-1),1:20) specifies global node numbers for element i which have been shifted in order to omit BC nodes when setting up the global equation system. numFe2(i(k-1),21:40) specifies the corresponding local node numbers and numFe2(i(k-1),41) specifies how many nodes are unspecified for element i. Index k corresponds to the velocity component in question.

D.3.6 Pointingarrays

The pointing arrays IA(*) and JA(*) used in connection with the sparse coefficient matrix are set up here (cf. also section 2.2.7). The inputs are Nefe, Nfe, the mesh tables numFe(*,1:20), numFe2(*,1:41) and the BC table nb(*,1:2). The outputs are IA(*), JA(*), the total number of non-zero entries in the coefficient matrix NAR, the size of the coefficient matrix Nnodes and an auxiliary array nodenumFe(*,1:17).
It is noted that this routine is only called once outside the time loop. However, it is quite slow and probably could be optimized.

D.3.7 Element_viscosity

The average viscosity is calculated in each finite element, cf. section 2.6. Inputs are Nefe, continuous and drop phase viscosities viscc and viscd, number of VOF cells per finite

element in each direction NVx, NVy and NVz, VOF cell volumes vofvolarea(*,1), the F-field F(*) and a table which relates local and global numbering of VOF cells NFE2VOF(*,*). Output is the array visc(*) with average viscosities in each finite element.
It is noted that this routine is called once each time step (when the viscosity ratio is different from one).

D.3.8 SetupAR

The non-zero entries in the global coefficient matrix are calculated here using the element matrices ematrixAR(1:20,1:20,1:9,*) and ematrixp(1:20,1:9,*). The coefficients are placed in the array AR(*). Other inputs are the pointer arrays IA(*) and JA(*), the number of non-zeros NAR, the number of elements Nefe, the mesh table numFe2(*,1:41), the auxiliary array nodenumFe(*,1:17), the BC array nb(*,1:2)+, the size of the equation system Nnodes and the viscosity array visc(*). The output is the sparse matrix array AR(*).
It is noted that if the viscosity ratio is different from one the equation system needs to be setup once for each time loop.

D.3.9 Read_In_Time_Dependent_BC

Externally determined shear and elongation rates are read in from datafile *bcdata.dat*. This routine is only used if shear and elongation rates need to be read from a file. Then the values of G and a (α) specified in the subroutine Boundarycond are overwritten. The local variable ntimesteps specifies the number of row entries in the data file to be read. Outputs are time(*), shearrate(*) , elongrate(*).

D.3.10 Time_Dependent_BC

The shear and elongation rates read from a file are used for generating time dependent BC's. For a given simulation time t $\dot{\gamma}(t)$ and $\dot{\epsilon}(t)$ are found from linear interpolation using the input arrays time(*), shearrate(*) , elongrate(*). These are then used for specifying the velocities on boundary nodes according to equation 3.1-1. Other inputs are Nfe, the boundary condition arrays nbcnodes(*,1:3) and nobcnodes(1:3) and the coordinate arrays xfe(*), yfe(*) and zfe(*). The output is the velocity array vel(*).
It is noted that one call is made to this routine for each time loop if BC's are based on a data file.

D.3.11 SetupRHS

Here the RHS vector is setup, i.e. the coefficients associated with boundary nodes on which velocities are specified are moved to the RHS side. Inputs are Nefe, the mesh tables numFe(*,1:20) and numFe2(*,1:41), the BC tables nb(*,1:2) and np(*) and the element matrices ematrixAR(1:20,1:20,1:9,*) and ematrixp(1:20,1:9,*). The output is the vector RHS(*).

D.3.12 BCbackInsert

When the flow field has been solved the nodal velocities are placed in the appropriate positions in the velocity vector vel(*). Inputs are the arrays nb(*,1:2), np(*), RHS(*) and vel(*). Output is vel(*).

D.4 Subroutines in module vofmesh.f90

The subroutines residing in the module vofmesh.f90 are described below. It is noted here that the VOF mesh is set up by subdividing each finite element into NVx×NVy×NVz VOF cells and that the block of VOF cells bounded by a given finite element is referred to as a VOF sub-mesh. Furthermore in a given sub-mesh a local VOF cell numbering is used thus when the term local VOF cell number is used it refers to the sub-mesh.

D.4.1 Setup_FE2VOF_and_numVOF

The VOF mesh table numVOF(*,1:8) and the tables NFE2VOF(*,*) and NVOF2FE(*) are generated. The array NFE2VOF(i,iloc) specifies the global VOF cell number of the local cell iloc in global finite element i. On the other hand the array NVOF2FE(i) specifies which global finite element global VOF cell i belongs to. The mesh table numVOF(*,1:8) relates the global and local node numbering of the linear nodes on the VOF mesh.
Inputs are Nefe, the finite element mesh parameters Nx, Ny and Nz and the VOF mesh parameters NVx, NVy and NVz. The outputs are numVOF(*,1:8), NFE2VOF(*,*), NVOF2FE(*), the total number of VOF cells Nevof and the total number of linear VOF nodes NvofNodes.
It is noted that this subroutine can only be associated with a regular mesh.

D.4.2 Setup_coordination_table

This routine sets up tables which are used for identifying VOF cells surrounding a central cell and for identifying VOF cell connected to a particular VOF mesh node. These table arrays are used both for calculating interface normals and for carrying out the interface advection. The array ncoordination_table(i,n) is used for identifying the n$= 1, \dots, 27$ VOF cells in the $3 \times 3 \times 3$ block of cells with VOF cell i in the center. For local node number iloc in a given VOF cell the array nodalcoordination(jloc,iloc) gives the position in the table ncoordination_table of the jloc$= 1, \dots, 8$ cells connected to the node. Thus e.g. local node 1 on VOF cell 20 is connected to cells ncoordination_table(20, nodalcoordination(1:8,1)). In principle the array

`nodalcoordination(1:8,1:8)` is used for specifying the ghost mesh described in section 2.4.2.
The inputs for the routine are the mesh parameters `Nx`, `Ny`, `Nz`, `NVx`, `NVy` and `NVz`. It is noted that for a regular mesh symmetry in the F-field is imposed by specification of `ncoordination_table(*,1:27)`. Furthermore this routine has been set up for a regular mesh and thus needs changes if another mesh is used.

D.4.3 Setup_vofcoord

Here the global coordinates for the linear nodes on the VOF mesh are specified and stored in array `vofcoord(*,1:3)`. Furthermore the coordinates for the center of each VOF cell are stored in `cntroid(*,1:3)`. The inputs are `Nefe`, the VOF mesh parameters `NVx`, `NVy` and `NVz`, the finite element mesh table `numFe(*,1:20)`, the finite element coordinates `xfe(*)`, `yfe(*)` and `zfe(*)`, the array `NFE2VOF(*,*)`, the VOF mesh table `numVOF(*,1:8)` and `NvofNodes`.
The routine calls the subroutine `LinearShape` which for given local coordinates ξ, η and ζ in a given parent finite element returns the value of the linear shape function `shapeL`. The global coordinates of the VOF nodes are based on interpolation using these linear shape functions.
This routine can be used for a general mesh.

D.4.4 Linear_shape_functions

Linear shape functions and the local derivative of the linear shape functions are evaluated at the integration points. Inputs are the number of integration points `NoInt`, the local coordinates of the integration points `xiInt(1:NoInt)`, `etaInt(1:NoInt)`, and `zetaInt(1:NoInt)`. The outputs are the liner shape functions `shapeL(1:8,1:NoInt)` and the derivative thereof `dLdloc(1:8,1:NoInt,1:3)`.

D.4.5 Setup_xi_eta_zeta_on_finite_element

The local coordinates of quadratic nodes on the local VOF cells residing in the parent finite element are set up. These are stored in the arrays `xi(*)`, `eta(*)` and `zeta(*)`. Inputs are `NVx`, `NVy`, `NVz`.

D.4.6 Setup_x_y_z_on_finite_element

Routine for calculating the global coordinates of the quadratic VOF cell nodes in a given finite element. Here the node numbering is based on the sub mesh system defined by the finite element. Inputs are the finite element in question `nelem`, the number of quadratic nodes NN associated with the VOF cells residing in the finite element, the arrays `xi(*)`, `eta(*)` and `zeta(*)` and the finite element mesh table `numFe(*,1:20)`.

The routine calls an auxiliary routine `QuadraticShape(xi(i),eta(i),zeta(i),quadshape)` which calculates the value of the quadratic finite element shape function at local coordinates `xi(i)`, `eta(i)` and `zeta(i)`. Index `i` is the node number in the sub mesh system. The quadratic shape function is used in the calculation of the global coordinates of the quadratic nodes. The outputs are the arrays `x(*)`, `y(*)` and `z(*)`.

D.4.7 Setup_num_vof_on_finite_element

Here a local VOF mesh table is set up which relates local and 'global' node numbering for the sub VOF mesh residing within a finite element. The inputs are the VOF mesh parameters `NVx`, `NVy` and `NVz`. The output is the array `numVOFFEM(*,1:20)`.

D.4.8 Local_nodes_on_element_faces

This routines sets up the array `nlocface(*,*)` which specifies the node numbering convention used on element faces (for face flux calculations). For local node `iloc` on face `i`$= 1, \ldots, 6$, `nlocface(iloc,i)` specifies the local node number on the 3D parent element. It is noted that the numbering convention affects the sign of the calculated fluxes and should only be changed with care.

D.4.9 Integration_points_for_flux_calculations

Sets up the integration points and weights for flux calculations across element faces. The input is array `nlocface(1:8,1:6)`. The Output is the integration points coordinate array `xint(1:9,1:3)` and the integration `weights(1:9)`.

D.4.10 Local_shape_functions_on_element_faces

In this routine quadratic shape functions are evaluated at the integration points on element faces. The value of the shape function associated with local node `iloc`$= 1, \ldots, 20$ at integration point `m` on finite element face `iface` is given by array `shapeNface(iloc,m,iface)`. A similar array is set up for the integration points on the VOF cell faces. These are given by the array `shapeNface2(i,iloc,m,iface)` where `i` is the local VOF cell number. The local derivative of the 2D quadratic shape functions associated with face `iface` and evaluated at integration point `m` is also calculated here and stored in `dNdface(iloc,m,iface,k)`. For this array `iloc`$= 1, \ldots, 8$ is the local node numbering used for each face and `k`=1,2 are the non-constant components of the local coordinates for face `iface`.

Inputs to this routine are the VOF mesh parameters `NVx`, `NVy`, and `NVz` and the array `xint(1:9,1:3)`.

D.4.11 Face_normals

Here the face normal vectors on integration points m= $1, \ldots, 9$ on face iface= $1, \ldots, 6$ are evaluated. The inputs are the array dNdface(1:8,1:9,1:6,1:2), nodal coordinates x(*), y(*), z(*), a mesh table num(*,1:20), the number of elements to be treated by the routine Ne and the array nlocface(1:8,1:6). The output is the array facenormals(1:Ne,m,k,iface) where k= $1, \ldots, 3$ are the components of the normal vector. It is noted that the routine can be used for calculating interface normals on VOF cells faces or on finite element faces by using appropriate values of the arrays x(*), y(*), z(*) and num(*,1:20).

D.4.12 Calc_fluxes

This subroutine calculates the volumetric fluxes on the faces of all the VOF cells residing in a given finite element. Here a call is made to the subroutine Face_normals in order to evaluate the face normals on the integration points. The procedure for calculating the fluxes is described in section 2.4.2.4. Inputs are the number of integration points NoInt, the local mesh table numVOFFEM(*,1:20), the VOF mesh parameters NVx, NVy and NVz, the quadratic shape functions shapeNface(1:20,1:9,1:6) and shapeNface2(*,1:9,1:6), the finite element mesh table numFe(*,1:20), the velocity array vec(*), the array NFE2VOF(*,*), the integration weights w(1:9) and the arrays xfe(*), yfe(*) and zfe(*). The routine Face_normals is called in order to generate the necessary face normals used in the flux calculations.
The face fluxes for global VOF cell i are stored in vofflux(i,iface). Furthermore the face areas are stored in the array vofvolarea(i,iface+1).
When the face fluxes have been calculated for all VOF cells in the finite element under consideration a call is made to the subroutine Adjust_fluxes_div_free which adjusts the fluxes such that all VOF cells become divergence free. This subroutine is described below.

D.4.13 Adjust_fluxes_div_free

This routine adjusts the calculated face fluxes according to the method described in section 2.4.2.5. The inputs to the routine are the VOF mesh parameters NVx, NVy and NVz, the finite element which is considered nelem, the array NFE2VOF(*,*), the arrays vofflux(*,1:6) and vofvolarea(*,1:7). The output is vofflux(*,1:6) where the fluxes in vofflux(NFE2VOF(nelem,1:NN),1:6) have been adjusted (NN is the number of VOF cells per finite element).

D.4.14 Setup_Ffield

In this subroutine the F-field is read in from the data file *Ffield.dat.* The data must be stored as a column vector in the file such that a given entry corresponds to the VOF cell number in the computational domain. The input to the routine is the total number of VOF cells `Nevof` while the output is `F(*)`.

D.4.15 Active_vof_interface_cells

Active interface VOF cells, i.e. cells with $0 < F < 1$, are identified (or tracked) in this subroutine. In the first call to the routine a search is made through all VOF cells in order to find cells with $0 < F < 1$. In the succeeding calls the search is limited to cells which are in the vicinity of the interface since the interface does not move more than a cell length for a given call to the advection routine (see later). The inputs to the routine are the total number of VOF cells `Nevof`, the F-field `F(*)`, the array `ncoordination_table(*,1:27)`, the number of active VOF cells, `nactive`, and the pointing array which points to the active VOF cells `nactivevof(*)`. The outputs are the total number of interface VOF cells `ninterface` and the array `nactiveinterface(*)` which points to the interface cells.

D.4.16 Active_vof_cells

All active VOF cells are tracked by this subroutine. The active VOF cells are interface cells include their neighbours. The inputs to the routine are `Nevof`, `F(*)`, `ncoordination_table(*,27)`, `nactiveinterface(*)`, `ninterface` (see previous routine for description). The outputs are the total number of active VOF cells, `nactive`, and the array `nactivevof(*)` which points to the active cells. It is noted that initially all VOF cell are defined as active.

D.4.17 Calc_face_flux_on_active_vof_cells

In this routine the face fluxes are calculated on all VOF cells which reside in finite elements in which one or more VOF cells are active. This routine calls the previously described subroutine `Calc_fluxes` in order to calculate the fluxes. Inputs to the routine are `nactive`, `nactivevof(*)`, `NVOF2FE(*)`, `Nevof` and `Nefe`.

D.4.18 Calc_dFdloc_in_vof_centre

The interface normal in the center of interface VOF cells is calculated in the local coordinate system. This normal is used in the PLIC reconstruction, cf. section 2.4.2.2. The interface segment position parameter α is also calculated in this routine via calls to the subroutine `Calc_alpha`. Inputs to the routine are `ninterface`, `nactiveinterface(*)`, `F(*)`, `Fnode(*)` and `Nevof`. The array `Fnode(*)` holds the values of the F-field interpolated onto the linear VOF cell nodes. The subroutine for making this interpolation is specified later (resides in module vofnormals.f90). The outputs from the routine are the center normals array `cntrlocnormal(*,1:3)` and the array `alphavof(*)` which holds the VOF cell values of α.

D.4.19 Calc_alpha

Routine for calculating interface segment position α using bisection iteration, c.f. section 2.4.2.2. The subroutine `Cut_volume` is called for calculating the cut volumes. The inputs for the routine are `F`, `h(1:3)`, `xn(1:3)`, `tol`, `tol2`. Here `F` is the value of the F-field in the VOF cell under consideration, `h(1:3)` are the element side lengths, `xn(1:3)` is the interface normal vector, `tol` is a tolerance used to determine if components of the interface normal should be treated as zero and `tol2` is the tolerance used in the stopping criterion for the bisection iteration. The output is `alpha`, i.e. the α-parameter.

D.4.20 Cut_volume

This subroutine is used for calculating the volume bounded by a plane segment which intersects the sides of an element, c.f. equation 2.4-12 and equation 2.4-13. The inputs are the α-parameter `alpha`, interface normal `xn(1:3)`, element side lengths `h(1:3)` and the tolerance `tol` which has the same meaning as in the subroutine described above. The output is the cut volume `vol`.

D.4.21 Advection

This is the main advection routine. The input is the array `ns(1:3)` which determines in which order the (split-operator) advection is to be carried out. For example if `ns=[1,2,3]` then the F-field is advected in the x_1-direction first where after the interface is reconstructed (new face normals and α's are calculated). Then this is repeated for the x_2-direction and finally for the x_3-direction. The routine first calls the routine `Calc_face_flux_on_active_vof_cells` in order to refresh the VOF cell face fluxes. Then the time step size `dt` is determined by calling the routine `Find_dt`. The actual advection is carried out by calls to the routines `Updatecolorfield`, `Active_vof_interface_cells` and `Calc_dFdloc_in_vof_centre` where the last routine updates the interface segment positions after each fractional advection step.

D.4.22 Find_dt

This routine determines the size of the time step size to be used in the advection procedure at a given time step number. The inputs to the routine are the arrays `nactivevof(*)`, `vofflux(*,1:6)` and `vofvolarea(*,1:7)`, the number of active VOF cells `nactive` and a user defined `CFL` number, cf. equation 2.4-26. The output is the time step size `dt`.

D.4.23 Updatecolorfield

Here the Lagrangian advection is performed, cf. section 2.4.2.3. Inputs are `Nevof`, `vofflux(*,1:6)`, `ncoordination_table(*,1:27)`, `nactivevof(*)`, `nactive`, `vofvolarea(*,1:7)`, `cntrlocnormal(*,1:3)`, `alphavof(*)`, the time step size `dt` and the advection direction under consideration `kk`. In this routine the time step size is adjusted according to equation 2.4-24. The routine utilizes the subroutine `Cut_volume` in order to calculate volumes leaving and entering cells. The output is the updated F-field `F(*)`.

D.4.24 SetupFaces

This routine writes the coordinates of the PLIC faces to the data file *facecoord[nc].dat* where *[nc]* is the value of the counter `nc` associated with the time step number. The routine also makes a call to the subroutine `Coordinates` which calculates the global coordinates of the faces. The format of the data file *facecoord[nc].dat* is as follows: For a given VOF cell the PLIC face is defined in terms of six coordinates (this is the maximum number of possible vertices of a given face plane). These six coordinates are written to the data file as blocks with 6 columns and 3 rows where row 1 corresponds to coordinate direction 1, row 2 to direction 2 and row 3 to direction 3. Coordinates from one VOF cell are not separated from coordinates for another cell. Thus when the PLIC faces are plotted in a post processing routine it is necessary to read the data in a consecutive order. The total number of faces in the file *facecoord[nc].dat* is given by the file *numfaces[nc].dat.* The Matlab script plotplic.m can be used for plotting the interface PLIC segments.

D.4.25 VOF_integration_points_on_finite_element

Here integration points and weights are set up for the VOF cells in the local coordinate system and scaled appropriately if the 2-level procedure is applied. The inputs are the number of integration points `NoInt` and the VOF mesh parameters `NVx`, `NVy` and `NVz`. The outputs are the integration points coordinate array `vofint(*,1:Noint,1:3)` and the integration weights `vofweight(*,1:NoInt)`. Each VOF cell within the parent finite element is given a set of integration points and weights such that the local VOF cell `i` has integration points `vofint(i,1:Noint,1:3)`.

D.4.26 VOF_dNlocal_on_finite_element

Local quadratic velocity shape functions and derivatives thereof are evaluated at the integration points in the VOF cells. The inputs are the number of integration points per cell `NoInt`, the VOF mesh parameters `NVx`, `NVy` and `NVz` and the integration point coordinates for the VOF cells `vofint(*,1:Noint,1:3)`. The outputs are the arrays `shapevof(*,1:NoInt,1:20)` and `dNdlocvof(*,1:NoInt,1:20,1:3)`. The value of the quadratic velocity shape function associated with local velocity node `iloc` and evaluated at integration point `m` in local VOF cell `i` is thus given by `shapevof(i,m,iloc)`. And similarly for the derivatives of the quadratic shape function `dNdlocvof(i,m,iloc,k)` where `k` is the direction component for which the shape function is differentiated with respect to (in the local system).

D.4.27 VOF_dNglobal_on_Finite_Element

This routine evaluates the global derivatives of the quadratic velocity shape functions at the VOF cell integration points. The routine operates with local numbering of the VOF cells, i.e. only cells within a given finite element are treated by the algorithm. Inputs are `NoInt`, `NVx`, `NVy`, `NVz`, `xfe(*)`, `yfe(*)`, `zfe(*)`, `numFe(*,1:20)` and the array `dNdlocvof(*,1:NoInt,1:20,1:3)` (local derivatives of the shape functions). The outputs are the array with the global derivative `dNdglob(*,1:NoInt,1:20,1:3)` and the determinant of the Jacobian matrix evaluated at the integration points `Djacvof(*,1:Noint)`. Thus the global derivative with respect to direction component `k` of the shape function associated with local velocity node `iloc` and evaluated at integration point `m` in local VOF cell `i` is given by `dNdglob(i,m,iloc,k)`. It is noted that this array corresponds to the quadratic velocity shape function in equation 2.5-44.

D.4.28 Calc_vof_element_volumes

In this routine the volume of each VOF cell is calculated and stored in `vofvolarea(*,1)`. The routine utilizes the Jacobian calculated in the routine `VOF_dNglobal_on_Finite_Element` (see above). Main inputs are `Nefe` and `NFE2VOF(*,*)`. Furthermore a call is made to `VOF_dNglobal_on_Finite_Element` for each finite element in the domain.

D.4.29 dLdglobal

Here global derivatives of the linear VOF cell shape functions are evaluated at the VOF cell integration points. The routine carries out the calculations for the VOF sub-mesh in finite element `nelem`. Other inputs are the coordinates of the linear VOF mesh nodes `vofcoord(*,1:3)`, the VOF mesh table `numVOF(*,1:8)`, the local derivatives of the linear shape functions `dLdloc(1:8,1:NoInt,1:3)`, the VOF mesh parameters `NVx`, `NVy` and `NVz` and the array `NFE2VOF(*,*)`. The output is the array `dLdglob(1:8,1:NoInt,*,1:3)` where the global derivative of the linear shape function associated with local VOF node `iloc` on local VOF cell `i` evaluated at point `m` is given by `dLdglob(iloc,m,i,k)` (`k` is the differentiation direction).

The output array is used for calculating the interface normal at the integration points of the VOF cells when the CSS interfacial tension model is applied. When the CSF model is applied the array is used for calculating the interface curvature at the integration points.

D.4.30 Identify_interface_finite_elements

This subroutine is used for searching for finite elements in which one or more active VOF cells reside. The inputs are `nactive`, `nactivevof(*)`, and `NVOF2FE(*)`. The output is a flag array `nactiveinterfacefe(i)` which has the value 1 if finite element `i` is part of the interface region and 0 otherwise.

D.4.31 Insert_vof_RHS

Here the terms in equation 2.5-44 (CSS) or equation B.8 (CSF) are calculated and added to the Right Hand Side vector RHS(*). The routine makes a call to the subroutine `Calc_F_at_nodes` if the CSS method is used or to `dF_at_nodes` if the CSF method is used (both reside in the vofnormals.f90 module). Then for each finite element which is considered to be part of the interface, the VOF cell contributions to the interfacial tension are calculated and summed together. It is noted that for each finite element calls are made to `vof_dNglobal_on_Finite_Element` and `dLdglobal`. The inputs for the routine are `Nefe`, `RHS(*)`, `numFe2(*,1:41)`, `vofweight(*,1:NoInt)`, `NoInt`, `shapevof(*,1:NoInt,1:20)`, `Nefe`, `Djacvof(*,1:NoInt)`, `NVx`, `NVy`, `NVz`, `sigma`, `Fnode(*)` and `dFnode(*)`. Furthermore the arrays `dLdglob(1:8,1:NoInt,*,1:3)` and `dNdglob(*,1:NoInt,1:20,1:3)` are obtained from routine calls inside the subroutine. The flag `itensionmet` controls wheter the CSS (`itensionmet`=1) or CSF (`itensionmet`=2) method is used. The output is the updated RHS vector.

D.5 Subroutines in module vofnormals.f90

The subroutines residing in the module vofnormals.f90 are now described.

D.5.1 Newtonit

This routine finds the local coordinates in the parent element for a given point with global coordinates `coordglob(1:3)`. This is done by solving the equations arising from the iso-parametric coordinate mapping between the global and local system using the linear shape functions, cf. equation 2.4-7. The routine also requires a coordinate array `coord(1:8,1:3)` with the global coordinates of the linear nodes and an array `sn(1:8,1:3)` with the corresponding local coordinates as input. The output is the vector `coordloc(1:3)` with the local coordinates.

D.5.2 dLdglob_at_vof_nodes

Here linear shape functions and global derivatives thereof are evaluated on VOF cell nodes. For a given node the routine makes a call to `Newtonit` in order to find the local coordinates for the node in the ghost cell enclosing it , cf. section 2.4.2.1. Then a call is made to the auxiliary routine `dLdloc_at_xi_eta_zeta` which evaluates the shape function and the local derivatives thereof at the local point given by `coordloc(1:3)`. The inputs to the routine are `Nx`, `Ny`, `Nz`, `NVx`, `NVy`, `NVz`, `cntroid(*,1:3)`, `nodalcoordination(1:8,1:8)`, `ncoordination_table(*,1:27)`, `numVOF(*,1:8)`, `vofcoord(*,1:3)`, `Nevof`, `h_fine(*)`, `x0`, `y0`, `z0`, `ALx`, `ALy`, `ALz`. It is noted that walls are treated as symmetric when boundary nodes are considered. This also means that if a non-regular domain is used some changes need to be made to the routine. The outputs from the routine are the arrays `shapeL_on_VOF_nodes(*,1:8,1:8)` and `dLdglob_vof_node(*,1:8,1:8,1:3)`. Here the linear shape function associated with local node `iloc` on the ghost cell and evaluated at local node `jloc` on VOF cell `i` is given by `shapeL_on_VOF_nodes(i,jloc,iloc)`. Similarly the global derivatives of the linear shape function are given by `dLdglob_vof_node(i,jloc,iloc,k)` where `k` is the differentiation direction.

D.5.3 calc_F_at_nodes

The F-field is interpolated onto the linear nodes on the VOF mesh. Inputs are `nactive`, `nactivevof(*)`, `numVOF(*,1:8)`, `F(*)`, `nodalcoordination(1:8,1:8)`, `ncoordination_table(*,1:27)` and `shapeL_on_VOF_nodes(*,1:8,1:8)`. The output is the array `Fnode(*)`.

D.5.4 dF_at_nodes

The gradient of the F-field is evaluated on the linear nodes on the VOF mesh. Inputs are `nactive`, `nactivevof(*)`, `numVOF(*,1:8)`, `F(*)`, `nodalcoordination(1:8,1:8)`, `ncoordination_table(*,1:27)` and `dLdglob_vof_node(*,1:8,1:8,1:3)`. The output is the array `dFnode(*)`.

D.6 Input files

The various input parameters which control the behaviour of the program are specified in one of three input files. These files are: *meshparam.dat*, *bcparam.dat* and *mainparam.dat*. In the file *meshparam.dat* all parameters associated with regular mesh set up by the subroutine Femcoord are specified. The file *bcparam.dat* is used for specifying which type of BC's are applied to the computational domain and in the file *mainparam.dat* physical paramters such as viscosity and density are specified. Furthermore in *mainparam.dat* some parameters which control the execution of the program are specified. The table below shows which parameters are specified in each file.

File name		**Parameters**
meshparam.dat	:	`Ncx`, `Ncy`, `Ncz`, `ALx`, `ALy`, `ALz`, `ALfx`, `ALfy`, `ALfz`, `x0`, `y0`, `z0`, `alphax`, `alphay`, `alphaz` (see subsection D.2.1 for further details).
bcparam.dat	:	`G`, `a`, `vterm(1:6,1:3,1:2)`, `nbd(1:6,1:3)`, `nbp(1:2)` (see subsection D.2.3 for further details).
mainparam.dat	:	`viscc`, `viscd`, `rhoc`, `rhod`, `itensionmet` (flag), `time_tot`, `CFL`, `nbcfile` (flag), `gacc(1:3)`, `iwritedata`.

D.7 Some additional notes

During a simulation a considerable amount of data is written to various output files. Since many of these are problem specific (e.g. mesh type and symmetry planes) they will not be described in detail. However, Matlab scripts have been written in order to analyse the output data, e.g. for plotting the PLIC faces and the velocity field. These scripts should be documented sufficiently in order to understand the purpose and structure of each data file. In the table below the output files are briefly described and when relevant the corresponding Matlab script is referenced.

File name		**Parameters**
time.dat	:	The time step number and simulation time are written to this file for each time step.
stepnumber.dat	:	The time step number is written to this file for each increment of nc which is controlled by the input parameter iwritedata (see also below). Together with the data in *time.dat* it is possible to relate the simulation time to the data files described below.
facecoord[nc].dat	:	Coordinates for the PLIC interface segments at time step nc*iwritedata (see also subsection D.4.24). Matlab script plotplic.m can be used for plotting the interface segments.
numfaces.dat	:	Number of PLIC segments at time step nc*iwritedata (see also subsection D.4.24).
Fout[nc].dat	:	F-field for the active VOF cells at time step nc*iwritedata. Matlab script plotFfield.m can be used for plotting the iso-surface from the F-filed.
velsym[nc].dat	:	Nodal coordinates and velocity field on the symmetry plane at time step nc*iwritedata. Can be plotted using the Matlab script plotvel.m.
psym[nc].dat	:	Pressure field on the symmetry plane at time step nc*iwritedata. Can be plotted using the Matlab script plotpres.m.
vofmesh.dat	:	VOF mesh center coordinates.
vofmeshsize.dat	:	VOF mesh size in the x_1, x_2 and x_3 direction.

The main time loop resides in the program part of the module mainfem.f90. Furthermore various arrays are initialized here by calling the appropriate subroutines outside the time loop. As described in the main text of this thesis the sparse linear solver PARDISO is used for solving the flow field variables. The behaviour of the solver is controlled by a relatively large number of input flags and parameters, e.g. for defining the type of coefficient matrix, the method used for solving the system, various tolerances and many more. Details on all the parameters and flags are given in the PARDISO documentation which can be found at *http://www.pardiso-project.org/index.html*. However, the input parameters used for solving linear equations arising in this work are described briefly in the module mainfem.f90.

Rúni Ditlev Egholm

Emulsion Design
Analysis of Drop Deformations in Mixed Flows

ISBN: 978-87-91435-73-0