2017

Numerical Simulation of Viscoelastic Multiphase Flows Using an Improved Two-phase Flow Solver

Olabanji Shonibare
Michigan Technological University, oyshonib@mtu.edu

NUMERICAL SIMULATION OF VISCOELASTIC MULTIPHASE FLOWS USING AN IMPROVED TWO-PHASE FLOW SOLVER

By

Olabanji Y. Shonibare

A DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

In Mathematical Sciences

MICHIGAN TECHNOLOGICAL UNIVERSITY

2017

This dissertation has been approved in partial fulfillment of the requirements for the Degree of DOCTOR OF PHILOSOPHY in Mathematical Sciences.

Department of Mathematical Sciences

Dissertation Co-advisor:	*Prof. Feigl, Kathleen A*
Dissertation Co-advisor:	*Prof. Tanner, Franz X*
Committee Member:	*Prof. Narain, Amitabh*
Committee Member:	*Prof. Labovsky, Alexander E*
Department Chair:	*Prof. Gockenbach, Mark S*

Contents

List of Figures

List of Tables

Preface

This dissertation is submitted for the degree of Doctor of Philosophy at Michigan Technological University. The research described herein was conducted under the supervision of Prof. Kathleen A. Feigl and Prof. Franz X. Tanner in the Department of Mathematical Sciences, Michigan Technological University, between September 2011 and December 2016. This work is to the best of my knowledge original, except where references are made to previous work. Neither this, nor any substantially similar dissertation has been or is being submitted for any other degree, diploma or other qualification at any other university.

Acknowledgments

An old adage says, “to know the road ahead, ask those who are returning”.

What a great honor and privilege to finally have the opportunity to express my deepest gratitude to my advisors, Prof. Kathleen Feigl (Kathy) and Prof. Franz Tanner for their excellent guidance, caring and patience. I am especially grateful for allowing me to pursue my ideas and providing support when I needed it most. I feel lucky and thankful for the wonderful research environment and opportunities that I enjoyed under your tutelage. It is difficult to overstate my gratitude to you, Kathy. Your enthusiasm, ingenuity and great efforts in explaining things clearly and simply has helped to make mathematics fun for me. I appreciate all your contributions. Definitely, I would have been lost without you. I would also like to thank my committee members, Prof. Alex Labovsky and Prof. Amitabh Narain for their constructive remarks and suggestions.

The members of the FETA CFD group have contributed immensely to my personal and professional development at Michigan tech. I am especially grateful to the fun FETA group members who stuck it out in grad school with me: Samer Alokaily, Chao Liang, Abdallah Al-Habahbeh and Ahmad Baniabedalruhman. I would also like to acknowledge honorary group member, William Case for sharing invaluable insights

and comments during group meetings.

I am greatly indebted to Prof. Mark Gockenbach and Prof. Thomas Drummer (May you take comfort in knowing that you have an angel to watch over you now), whose work and influence has altered my line of thinking. Mark has been a constant source of moral support, and I hope that I can reciprocate some of all that affection you gave me.

I would like to acknowledge the financial support provided by the Department of Mathematics. After years of learning, it's easy to forget how little we knew in the first place. In this regard, I would like to sincerely thank the following lecturers from whose shoulders I have been able to see afar: Kathleen Feigl, Franz Tanner, Alexander Labovsky, Iosif Pinelis, Allan Struthers, Jiguang Sun, Zhengfu Xu, Todd King, Ann Humes and Elizabeth Reed. I am grateful to Jeanne and Margaret for helping with the smooth run of the department and for assisting me in many different ways. How would I forget Jason? Jason, thank you very much. I am always inspired by your personality.

Hmmm . . . (*misty eyes*). To the best parent in the world - Kofoworola Shonibare and Moshood Shonibare. You bore me, raised me, supported me, taught me, and loved me. I hope I've made you proud, and will always give you my deepest gratitude.

The All-Merciful, The Greatest, The Expediter, The Delayer, The First, The Last,

The Owner of All, The Light; To YOU I dedicate this thesis.

ABBREVIATIONS

Acronyms

FVM	Finite Volume Method
SIMPLE	Semi-Implicit Method for Pressure-Linked Equations
PISO	Pressure-Implicit with Splitting of Operators

Dimensionless numbers

Re	Reynolds number
De	Deborah number
Wi	Weissenberg number

Greek symbols

λ	relaxation time
$\boldsymbol{\tau}_p$	polymer stress component
$\boldsymbol{\tau}_s$	Newtonian stress component

ρ	density
η	shear viscosity
η_p	polymer viscosity
η_s	solvent viscosity
σ	surface tension
δ	Dirac delta function
$\dot{\gamma}$	shear rate

Roman symbols

N_1	first normal stress difference
N_2	second normal stress difference

Abstract

The production of uniformly-sized droplets has numerous applications in various fields including the biotechnology and chemical industries. For example, in the separation of mixtures based on their relative absorbency, an optimal arrangement of monodispersed droplets in columns is desired for an effective separation. However, very few numerical studies on the formation of viscoelastic droplets via cross-flow shear are available, none of which have considered the case when the flow of the continuous phase is Couette. In this work, a new solver capable of automatic mesh refinement is developed for the OpenFOAM® CFD toolbox to solve viscoelastic two-phase flow problems. The finite volume method is used to discretize the governing equations while employing the Volume of Fluid (VOF) coupled with the level set method to accurately describe the interface. The fourth-order least squares method is applied to the reinitialization of the level set function. Mesh refinement and coarsening procedure is based on a specified range of the volume fraction field. To validate the numerical technique, two-dimensional numerical simulation is conducted for a drop under static conditions, drop deformation in shear flow, the rise of a Newtonian drop in a Giesekus liquid and formation of viscoelastic droplet in a microfluidic T-junction. Furthermore, the effect of flow type and fluid elasticity on drop size and droplet formation dynamics was investigated in a viscoelastic-Newtonian system. The results obtained show good qualitative agreement with experimental work. In both cases

where the flow of the continuous phase is pressure-driven (P-flow) and plane Couette (C-flow), there was a decrease in drop size as the cross-flow shear rate increased. However, for a fixed average shear rate, the drop sizes generated in C-flow were found to be smaller than that in P-flow. It was also found that the influence of elasticity on drop size became accentuated as the cross-flow shear increased. An increase in elasticity was accompanied by a decrease in drop size.

Chapter 1

Introduction

1.1 Two-phase flows

There exists a huge number of natural and industrial operations where some form of fluid flow with material interfaces or free surfaces is present. In many application flows, we have short time scales and small length scales, and identifying spontaneous fluctuations of some flow features is challenging because dynamic measurements have to be resolved in time and space. The use of computer simulations can help alleviate these shortcomings. Since the hydrodynamic behavior of the fluid is largely dependent on properties that are influenced by the shape of the front, such as surface tension, an accurate estimation of the location and curvature of the interface is invaluable.

Current methods used to simulate flows with a material interface can be split into two major groups: volume methods and surface methods [2, 3, 4]. In the surface methods, marker points with specific attributes are used to explicitly track the interface or the computational mesh is aligned with the interface and programmed to advance with the interface. Although, with this method, the location of the interface is known all through the simulation and does not smear as the simulation progresses, it is very inefficient in handling large topological and interface movements. Examples of surface methods include front-tracking method [5] and level-set (LS) method [6, 7]. In the volume methods, an indicator function or particles that have no mass are used to identify the fluid on both sides of the interface. The major disadvantage of this approach is that since the precise location of the front is not determined explicitly, special procedures are required to reconstruct the interface. Examples of volume methods include marker and cell (MAC) method [8, 9] and volume of fluid (VOF) method [10, 11].

Among these numerical methods, the LS and VOF methods are very common. A level-set function ψ is used to characterize the interface when adopting the level-set approach. The function ψ is initialized as a signed distance function away from the interface - its zero level set represents the location of the interface and the surface height equals the distance from (x, y) to the nearest point on the interface so that ψ has a positive distance outside the interface contour and a negative distance inside it. The transport equation for ψ is then solved using the velocity field of the previous

time step to determine the next location of the interface. Typically, at later times, the level set function fails to remain a distance function which introduces error in further calculations and makes the conservation of mass inachievable. Thus, it needs to be "re-shaped". A numerical solution that has been extensively used is to solve a re-initialziation equation. The strength of the level-set method lies in the smooth variation of ψ across the interface which enables an accurate estimation of the normal vector and curvature at the interface.

In the VOF approach, the volume fraction ϕ for an individual fluid (say A) in any cell that contains the interface is evolved. As a result, ϕ can only take on the following values: 1 - the cell contains only the fluid A; 0 - the cell contains only the other fluid; $0 < \phi < 1$ - the cell contains both fluids (i.e. it contains the interface). There exist several variations of the VOF method [10, 11, 12, 13] and in general constitute three main procedures: (1) reconstruction of the interface (2) the VOF transport equation is solved to determine the volume fraction field for the next time step using the velocity distribution from the previous time-step and (3) the approximation of the surface tension force at the interface. A principal attribute inherent in the VOF methods is its conservation of volume throughout the simulation. However, a major concern when applying the VOF method to surface-tension-dominated flows is the presence of artificial currents. These spurious velocities are generated as a result of a numerical imbalance between the pressure-gradient terms and the surface tension force [14, 15]. To reduce these spurious currents, an extra variable such as the height

function [16, 17] or a level-set function [6] can be utilized only for the computation of the curvature at the interface.

Over the years, the application of computational fluid dynamics (CFD) models to solve multiphase flow problems has grown rapidly as a result of a combination of an ever-increasing computer efficacy and advanced numerical methods. Most CFD-related studies on two-phase immiscible flows have employed open source and commercial CFD packages like Fluent [18], Star-CCM+ [19], CFX [20], CFD-ACE+ [21], SU2 [22], Gerris [23] and CONVERGE [24].

The use of the open source software package, OpenFOAM® [25], which is based on the cell-centered finite volume method, is gaining popularity as an efficient alternative to commercial CFD packages. The wide range of pre-implemented fluid models and utilities available, automatic portability for parallel programs, polyhedral mesh support, unlimited extensibility and availability of a large and growing user community are only a few of the benefits derived from using OpenFOAM. Two-phase viscoelastic fluid flow is handled in OpenFOAM using the solver, `viscoelasticInterFoam`, developed by Favero et al. [26]. It is a transient solver for handling two incompressible, isothermal and immiscible fluids using the VOF approach on a static mesh. The new solver, `clsVeInterDymFoam`, developed in this thesis could be seen as an extension of the initial work by Favero et al. [26]. With the new solver, the estimation of the surface tension force from VOF function is now improved by coupling the VOF and

the LS method - the interface is captured using the VOF method while the interface normal vector and curvature are estimated from the smooth LS function. The new solver is also capable of adaptive re-meshing in regions of the mesh where the VOF function falls within a specific user-defined range. The AMR implemented in clsVeInterDymFoam has originally been developed by Baniabedalruhman [27].

1.2 Adaptive mesh refinement

In multiphase flows, the occurrence of wide variations in spatial resolution within the numerical domain is a regular feature in many applications such as interfaces between different phases and compressible flows with waves and shocks. In particular, one often encounters sections with steep gradient when solving hyperbolic partial differential equations. The solution obtained is far from reality as a result of the over-approximation of sharp gradient areas. Past attempts to resolve this problem employed body-fitted [28, 29] and unstructured meshes [110] but its robustness was strongly influenced by the type of problem investigated. Another approach uses a structured mesh as the base mesh and when an area with high spatial variation is detected, the grid cells are directed close to that neighborhood. The lapse in this approach is that it can lead to a strong level of mesh skewness [109]. In the mesh embedding method [28], the solution process starts on a coarse mesh and when the gradient of the solution within a zone of cells is observed to be very steep, a local

refinement of cells is done in that neighborhood while retaining the structure of the original mesh.

In general, the strategies used for grid adaptation can be grouped as either h-refinement, r-refinement, p-refinement or a hybrid of any e.g. hr- and hp- refinement. The h-refinement procedure involves adjusting the size of the mesh at certain regions by adding/removing points from the numerical domain and this ultimately results in the alteration of the grid connectivity. The r-refinement, on the other hand, adjusts the resolution of the mesh by relocating grid points towards the mesh region with a steep gradient in the solution while retaining the original number of computational points. The use of p-refinement is more common in finite element method. Here, the accuracy of the solution is improved by varying the order of discretization in an individual cell.

Based on the partitioning procedure and data structure monitoring the connectivity of the grid, the approach through which adaptive mesh refinement (AMR) is achieved can be grouped into four classes: patch-based, cell-based, block-based and hybrid block-based.

The patch-based approach was initially designed by Berger and Oliger [30, 31, 32]. The method starts with a cartesian mesh that is coarse. As the simulation advances, some clustering procedures are used to assemble cells already tagged for refinement,

resulting in the formation of rectangular grid boundaries. In the block-based approach, a block is defined in advance. If any cell inside this block is tagged for refinement, the whole block is refined. A benefit derived from using this approach is that the data structure associated with each block is straightforward and the subgrid is well-structured. The major drawback is that meshes are often over-refined in regions where it is not required. The integration of both concepts (i.e. patch-based and block-based) have already been attempted [33] with the sole aim of overcoming the imperfections in both methods.

The cell-based AMR was proposed and developed by Powell and coworkers [34, 35, 36, 37] and Berger and Levque [38]. In this approach, each cell may be refined individually and is then stored via a tree data structure. This strategy is flexible and readily allows for the local refinement of the mesh by keeping track of the computational cell connectivity as new grid points are generated by the refinement process. This is the approach used in OpenFOAM.

Applying adaptive strategies as a standard approach to solve classical dynamic problems has been in progress for a long period of time. Berger's work [30] served as the cornerstone for subsequent development seen in structured adaptive mesh refinement (AMR) applications. Bank [39], and Banbuska and Rheinboldt [40] proposed, in their papers, adaptive finite element methods to solve elliptic problems. Similar efforts were

made by Sherman and Sager [41], and Davis and Flaherty [42] for parabolic equations. Several researchers [43, 44, 45] have also developed adaptive mesh procedures to solve hyperbolic equations. We should note, however, that there are several other implementations of AMR different from those derived from Berger's principles. Some examples can be found in [46, 47].

1.3 Application background

The dispersions of two or more immiscible liquids is referred to as an emulsion or polymer blend. Some examples of emulsions include mayonnaise, vinaigrettes and butter. Emulsions have found great uses in many industries. For example, in the agriculture industry, emulsion technology aids dilution and provides better sprayability of insecticides and pesticides; in the pharmaceutical industry, they are applied to make drugs more edible and fine-tune dosage of active ingredients while in the food industry, emulsions influence the physical appearance and mouthfeel of food products. Major emulsification methods used to produce uniformly-sized microdroplets include microfluidic processes, microchannel emulsification and membrane emulsification. A very good understanding of droplet formation mechanism enables the determination of the feasibility and boundary of the use of membrane emulsification in different kinds of applications.

Typical apparatus for the production of emulsions include agitators, rotor-stator systems, high-pressure valve homogenizers and ultrasound systems. These apparatus rely on turbulence for the disintegration of large drops in a pre-emulsion. A major drawback with this technique is that in most cases it results in droplets that are highly polydisperse [48]. The drop size distribution has a strong influence on the physical and chemical properties of emulsions. For example, in the food industry, monodispersed emulsions improve the qualities of a product such as mouthfeel, physical appearance, flavor and shelf-life [49]. Also, the generation of droplets in these devices are accompanied with energy consumption and shear stresses that are very high. This is not only expensive but has a damaging effect on food and pharmaceutical products [50]. Improvements on these techniques were made over the years, some of which include membrane emulsification, microchannel emulsification and microfluidic processes. In membrane emulsification, droplets are generated either as a result of the decomposition of a coarse emulsion after being forced through a membrane channel or shearing of the pure injection source by the continuous phase. The membrane used in membrane emulsification devices could be either fixed or dynamic, where the rotation/ vibration of the membrane also aids in the pinch-off of droplets from the membrane surface [51, 52]. In general, microfluidic devices are categorized as either flow focusing or microfluidic junctions. Of all microfluidic junctions, the T-junction is easiest to construct [49, 53]. In the T-junction, the dispersed phase is injected at normal direction into a flowing stream of the continuous phase, droplets

then detach from the tip of the injection source as a result of several factors including the accumulation of pressure upstream of the growing droplet and drag from the continuous phase. Although productivity of droplets via membrane emulsification is higher when compared to either microchannel emulsification or microfluidic processes, they are highly polydisperse [54]. On the other hand, with microfluidic processes, the user gains precision over the size, homogeneity and even the inner composition of the droplets.

In a membrane emulsification apparatus, when the ratio between the pore distance and size is small, it can result in droplet coalescence. In an attempt to resolve this problem, Schadler and Windhab [55] devised a rotating membrane emulsification apparatus with an adjustable distance between pores. They conducted a study to determine the effects of rotational speed and the volume ratio of the drop and matrix phase on drop detachment characteristics. They found that the rotational speed of the membrane has a direct relationship on the size of droplets formed and also claimed that the width between gaps has a strong impact on the formation mechanism. Similar results were found in other experiments [56] . The presence of surfactant in emulsions reduces the interfacial tension between different pairs of phases which helps to lower the emulsification pressure and promote stability of droplets. Several authors have investigated the role of surfactants in the droplet formation process of membrane devices [50, 57, 58, 59]. Their results showed that increasing interfacial tension results in an increase in formation time and drop size of droplets. Van der Graaf *et al.* [49]

studied droplet formation in a T-shaped model system for a cross-flow membrane emulsification device. He observed a direct relationship between droplet size and flow rate. It has also been shown in some studies that size of droplets decreases as the wall shear stress is increased [58, 60].

Most experimental studies on droplet formation using elastic fluids have utilized flow focusing devices [61, 62, 63] while only a few were performed with a T-shaped microchannel [64, 65]. Hong and Cooper-White [61] investigated the formation of carbopol dispersions that shear thin and possess yield stress via a flow-focusing micro geometry. They claimed that below a critical value of the continuous phase flow rate, Q_c, there is a direct relationship between the size of droplets formed and the viscosity ratio but beyond this critical value, non-Newtonian properties of the fluid begin to surface and results in a decrease in droplet size when viscosity ratio increases. They also argued that in the absence of satellite droplets, the shear thinning and elastic property of the fluid results in the formation of drops with smaller sizes than the case when elasticity is neglected. They attributed this result to the formation time being shorter. Steinhaus *et al.* [63] studied the effect of channel dimension and fluid elasticity on the generation of polymeric drops within a Newtonian matrix in a flow-focusing micro channel. Their results showed that increasing elasticity produced longer thread lengths and longer detachment times. Similar results were also found in [61, 62]. Husny and Cooper-White [64] studied the detachment dynamics of droplets formed in a T-junction. A Boger fluid was used as the dispersed phase

and silicone oil for the continuous phase. It was determined that the presence of elasticity in the drop phase precipitated elongated filaments, which also resulted in the production of satellite droplets between drops produced at regular intervals. The form of these filaments was shown to depend on the viscosity ratio and molecular weight of the polymers. In addition, the characterization of secondary drops formed was investigated in great detail and it was concluded that the monodispersity of these satellite droplets depended predominantly on the viscosity ratio and the flow rate of the continuous phase.

A few numerical studies have been conducted on the characterization of viscoelastic droplet formation in a Newtonian stream. For example, viscoelastic drop formation at an aperture [66, 67] and in a flow-focusing channel [68]. To the best of our knowledge, no numerical study on the formation of viscoelastic droplets in a T-junction has been considered.

1.4 Goals

The goals of this thesis are to: (1) develop an improved two-phase flow solver for viscoelastic and Newtonian fluid systems; (2) test the improved solver on a series of test problems; and (3) use the improved solver to study the formation and detachment of viscoelastic drops in a T-shaped microchannel.

1.5 Contributions of this thesis

This thesis makes several contributions to the field of Computational Viscoelastic Fluid Dynamics. The major contributions are:

1. The two-phase incompressible flow solver for viscoelastic two-phase flows, `viscoelasticInterFoam` of Favero [26] has been improved by coupling it with level set method in OpenFOAM-2.3.x to accurately describe the interface.

2. The original adaptive mesh refinement (AMR) engine in OpenFOAM is programmed to work for only three-dimensional numerical simulations. The code has been modified by Baniabedalruhman [27] to also work in two-dimensional planar and axisymmetric geometries. The modifications have been coded into binary executable library files called `dynamicRefineFvMesh2D` and `dynamicRefineFvMeshAxi`, and can be linked dynamically at run-time in OpenFOAM. This extra functionality has been incorporated in the newly developed solver called `clsVeInterDymFoam`.

3. The new solver was then validated by applying it to the following two-dimensional test problems:

   (a) 2D drop under static conditions.

   (b) Drop formation in shear flow.

(c) Rise of a Newtonian drop in a Giesekus fluid. The computational domain here is axisymmetric.

(d) Drop formation in a T-junction under experimental conditions considered by Li et al. [1].

4. The modified code was further validated by comparing a two dimensional numerical simulation of droplet formation in a T-junction with experiments. The results were found to be consistent with experimental observations.

5. The effect of flow type, wall shear rate, and fluid elasticity on drop size and droplet formation dynamics was investigated in a viscoelastic-Newtonian system via a T-shaped micro channel.

6. To the best of our knowledge, this is the first consideration of formation of drops in a microfluidic T-channel where the flow of the continuous phase is Couette.

The codes are documented in the appendix.

Chapter 2

Theoretical Models

2.1 Introduction

Solutions containing polymers constitute interpenetrating chain molecules that are oriented in a random fashion. There is a correspondence between the elasticity of a polymeric fluid and both the random molecular movement and average expansion of polymer chains. To ensure the molecular chains are randomly oriented, Brownian motion resists any expansion or parallel configuration of the chains. In other words, it opposes the stretching and alignment of the chains because it acts to keep the chains in random configurations. When a fluid is subjected to deformation, the amount of molecular elongation seen is determined by the degree to which viscous stress is more

than that generated from the random movement of the chains. As soon as the force is removed, the deformation of a Newtonian fluid stops immediately but still continues for a viscoelastic fluid. The ability of the internal structure of a viscoelastic fluid to retain stress for a short period of time, typically referred to as the *relaxation time*, is responsible for this behavior. The amount of elasticity in a polymeric fluid is assessed from the relaxation time.

An additional method through which viscoelasticity can be gauged is via stress relaxation. For example, when a purely viscous liquid undergoes a step-strain shear, an instantaneous decline to zero of the stress is observed but for a polymeric liquid, the stress comes to rest in an exponential manner. Representing stress relaxation data in terms of a relaxation modulus,

$$G(t) = \frac{\boldsymbol{\tau}(t)}{\gamma}, \tag{2.1}$$

we observe that for small strains ($\gamma < \hat{\gamma}$), the relaxation modulus coincides.

This linear dependence of stress relaxation on strain is called *linear viscoelasticity.* In general terms, a fluid which has a linear dependence between the current stress value and its strain history is referred to as a *linear viscoelastic fluid.* In Figure 2.1, G_o is referred to as the plateau modulus and it is the limiting value of the relaxation modulus for a small duration. For higher strains, however, the relaxation is now dependent on strain (see Figure 2.1). Equation 2.1 can now be written as

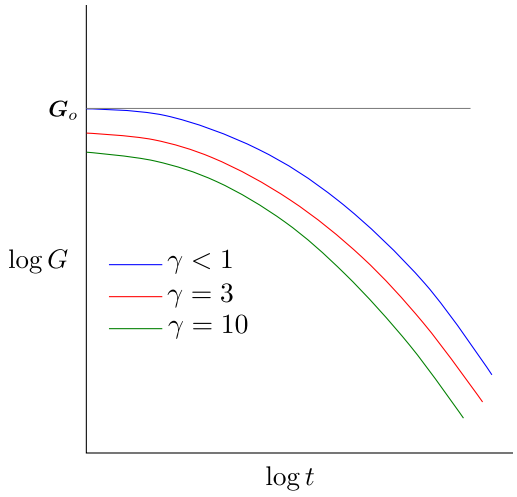


Figure 2.1: Log-log plot of shear modulus against time

$$G(t,\gamma) = \frac{\boldsymbol{\tau}(t,\gamma)}{\gamma} \quad (\gamma > 1) \tag{2.2}$$

and this viscoelastic behavior is termed *nonlinear*.

Nearly all polymeric fluids, when subjected to shear flow, show disparate steady-state reactions and unsteady-state reactions. There exist several types of unsteady shear flows. Some of these include shear-stress growth, shear-stress decay, shear creep, step shear strain and small-amplitude oscillatory shear. A concrete understanding of the range of nonlinear characteristics can be obtained by examining the unsteady state responses from these tests. The outcome from any of the tests can only be utilized to estimate the responses from other tests when the deformation rate is very low. On the other hand, when the deformation rate is large (nonlinear regime), rheological data obtained from any of these nonlinear probes are only useful to the test at hand.

Investigating the rheology of polymeric liquids is very challenging. The dimensionless Deborah number, De, is typically employed as an initial step to gain qualitative information about the fluid flow and is defined by

$$De = \frac{\tau}{t}$$

where τ is the material's characteristic relaxation time and t represents the characteristic flow time. In general, when $De < 1$, the rheology is considered linear and for $De \leq O(1)$, the material is assumed liquid-like. On the other hand, when $De \gg 1$, the material is said to possess a solid-like behavior. We remark here that the critical De depends on the type of flow.

It should be noted that there are exceptions for steady linear flows. For instance, there have been reported cases [130, 131, 132, 133] of elastic turbulence seen in polymer solutions subjected to a very high deformation rate at low Reynolds number (See Figure 2.2).

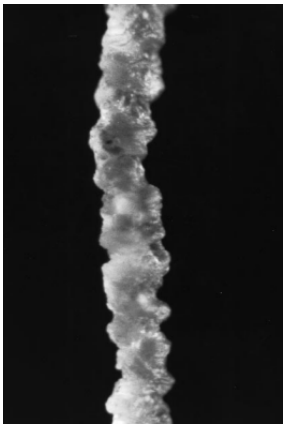


Figure 2.2: An example of elastic turbulence seen from a polymer melt forced through a circular orifice at low *Re*. Reprinted from Fluid dynamics: Turbulence without inertia, by Larson, R. G. [134], 2000, Nature, p. 27. Copyright 2000, with permission from Nature. See documentation in Appendix A.

2.2 Nonlinear phenomena

There are two kinds of nonlinearities that are usually seen in applications and they are referred to as *geometric* and *material* nonlinearities. Material nonlinearities is used to describe nonlinear stress-strain reaction that appears based upon the intrinsic properties of the material. Geometric nonlinearities, on the other hand, are usually associated with solids, are said to occur if the assumption of linear relationship between stress and strain fails to hold as a result of the strain and displacement reaching a high value. We should note that there are some materials (e.g. rubber) where a linear relationship holds between stress and strain for strains as high as 60% [123]. It is also possible to observe both kinds of non-linearities for high strain.

A clear comprehension of nonlinear processes is extremely important when it comes to developing polymeric compounds. Polymeric fluids, in general, exhibit different characteristics that cannot be reproduced by viscous laws. Some of these features are discussed below:

2.2.1 Normal stress differences in shear

When a transducer is used to determine the stress on a surface, what is being felt by the device is the total stress, $\mathbf{\Pi} = \boldsymbol{\tau} - p\boldsymbol{I}$, i.e.

$$\mathbf{\Pi} = \begin{pmatrix} \tau_{11} - p & \tau_{12} & \tau_{13} \\ \tau_{21} & \tau_{22} - p & \tau_{23} \\ \tau_{31} & \tau_{32} & \tau_{33} - p \end{pmatrix}.$$

Suppose we wish to determine the stress at an arbitrary point, O, in the medium. Then, Π_{mn} represents the stress at O on an $m-$plane in the $n-$direction. For example, Π_{12} connotes the stress at O on a plane whose unit normal is $\hat{\boldsymbol{i}}$ and is parallel to $\hat{\boldsymbol{j}}$, Π_{23} connotes the stress at O on a plane whose unit normal is $\hat{\boldsymbol{j}}$ and is parallel to $\hat{\boldsymbol{k}}$.

Each entry in the diagonal of the total stress is a sum of the pressure and normal extra stresses $(\tau_{11}, \tau_{22}, \tau_{33})$. This makes the independent evaluation of pressure on a

surface difficult for an incompressible non-Newtonian fluid subjected to shear flow. For a compressible fluid, it is not difficult to compute singly, since the pressure can be obtained using the ideal-gas law ($p = RT/V$). The case when the incompressible fluid is purely viscous is trivial because the normal stresses are each zero. To resolve the complications with viscoelastic fluids, the normal-stress differences are computed as an alternative to normal stresses. Thus, when a fluid is subjected to any of the standard flows, the normal stress is assessed by determining the *First normal stress difference*, $N_1 \equiv \Pi_{11} - \Pi_{22}$ to obtain

$$N_1 = \tau_{11} - \tau_{22}$$

and the *Second normal stress difference*, $N_2 \equiv \Pi_{22} - \Pi_{33}$, which gives

$$N_2 = \tau_{22} - \tau_{33}.$$

For a viscoelastic fluid subjected to shear flow, these normal stress differences are non-zero. In most cases, N_1 is positive and N_2 is negative [113]. A very good example that demonstrates the development of normal stresses in viscoelastic fluids is the rod-climbing phenomena (See Fig. 2.3). A rotating rod placed in a fluid generates circular streamlines in the flow. When the fluid is viscoelastic, the polymers generate tension

along the streamlines as a result of its elasticity - when the polymers are close to the rotating rod, shear causes the polymers to stretch. To retain its original form, it exerts a force towards the rod. An aggregation of forces from all the polymers directed close to the rod pushes the surrounding fluid up. The normal stress instigated by the shear in the viscoelastic fluid is the reason for this behavior. When the shear rates are small, the relationship between τ_{12} and $\dot{\gamma}$ approaches linear. Thus, (N_1, N_2) tend to ($N_1 \propto \dot{\gamma}^2, N_2 \propto \dot{\gamma}^2$), so that the normal stress coefficients,

$$\Psi_1 \equiv \frac{\tau_{11} - \tau_{22}}{\dot{\gamma}^2} \tag{2.3}$$

$$\Psi_2 \equiv \frac{\tau_{11} - \tau_{22}}{\dot{\gamma}^2} \tag{2.4}$$

emerge as constants.

2.2.2 Shear thinning

The decrease in viscosity of polymer solutions as shear rate is increased is referred to as *Shear thinning* or *pseudoplasticity.* Figure 2.4 shows a typical plot of viscosity against shear rate for a shear-thinning fluid. The viscosity is approximately constant at low-shear rate limit, decreases as shear rate increases, and then may approach a

Figure 2.3: The Rod-Climbing effect. Reprinted from An introduction to Rheology, by Barnes, H. A. et al. [115], 1989, Elseiver, p. 61. Copyright 1989, with permission from Elseiver. See documentation in Appendix B.

constant in the high-shear rate limit. The two plateau sections in Fig. 2.4 are often termed the "first Newtonian region" and "second Newtonian region". The value of the viscosity within the first Newtonian region is called the *zero-shear viscosity*.

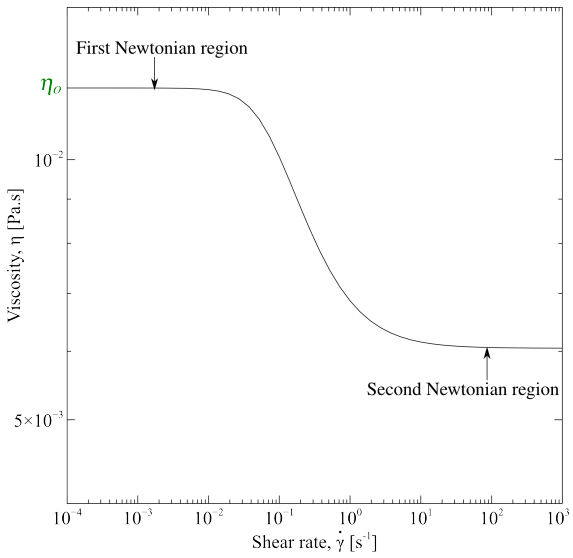


Figure 2.4: Schematic representation of a shear thinning material

Unlike the Newtonian case, the viscosity which now depends on time is determined

by

$$\eta(\dot{\gamma}, t) = \frac{\tau_{12}(\dot{\gamma}, t)}{\dot{\gamma}} \tag{2.5}$$

2.2.3 Extension and shear thickening

It is typical for the viscosity of a polymeric fluid to decrease with increasing deformation rate in shear flow. In an extensional flow, the reverse often occurs. The extensional viscosity, η_e, of some materials, which is a measure of the resistance of the material when subjected to stretching, decreases as the extension rate, $\dot{\epsilon}$, grows while for some, η_e grows as $\dot{\epsilon}$ rises. The behavior of the former is referred to as *tension-thinning* while that of the latter is described as *tension-thickening*. To avoid breakage, especially during a film blowing and fiber spinning operation, it is beneficial to determine beforehand if the material used exhibits tension-thickening or tension-thinning. Tension-thinning materials typically break when the strain exceeds a certain limit [129]. The elongational viscosity as a function of the extension rate for a solution containing dekalin and polybutadiene [128] is shown in Fig. 2.5. As η_e increases, the mixture is seen to tension-thicken and after a limit it tension-thins.

Comparing the graph of viscosity against strain rate for polymer melts and a dilute polymer solutions provides a faster way to easily determine qualitative differences between both fluids. We show in Fig. 2.6 a representational plot of viscosity against

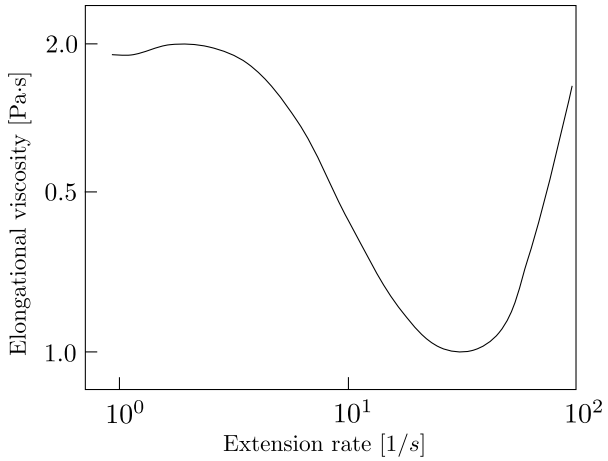


Figure 2.5: Elongation viscosity for a polymer solution containing polybutadiene and dekalin. Reprinted from Correlation and molecular interpretation of data obtained in elongational flow, by Hudson, N. et al. [128], 1976, Transactions of The Society of Rheology , p. 280. Copyright 1976, with permission from AIP. See documentation in Appendix C.

extension rate for dilute polymer solutions and polymer melts. For the dilute solution (a), we observe a sharp increase in η_e beyond a certain strain rate but (b) shows that η_e changes considerably as the extension rate is increased.

Although not common, there are situations where the viscosity is seen to increase as the shear rate grows. This property is known as *shear-thickening*. Examples of such materials include concentrated suspensions of titanium dioxide in a solution of sucrose and a mixture containing ethylene, glycol, corn starch and water [126, 127]. The response of TiO2 suspensions undergoing shear is delineated in Fig. 2.7. We observe that the suspensions shear thin when the shear rate is low but within the high-shear rate region, a rise in the viscosity of the suspensions is noticed as the shear rate rises (i.e. it shear-thickens). The reason for this shear-thickening in suspensions

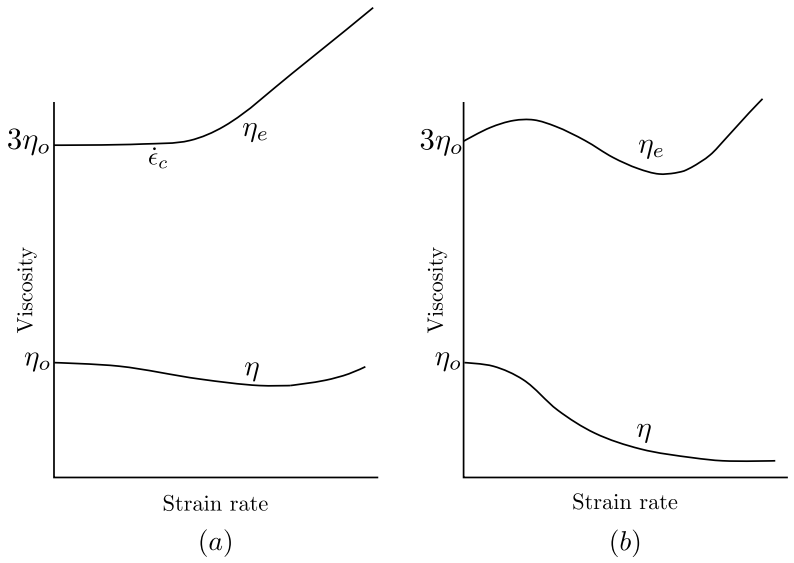


Figure 2.6: General behavior of viscosity against strain rate for (a) a dilute polymer solution; (b) a polymer melt. Reprinted from An introduction to rheology, by Barnes, H. A. et al. [115], 1989, Elseiver, p. 100. Copyright 1989, with permission from Elseiver. See documentation in Appendix D.

has been attributed to the growth in its volume as shear rises and it begins just after the volume begins to expand [127]. Other cases of shear-thickening in solutions containing polymers have also been reported elsewhere [125].

2.3 Constitutive models

Without doubt, viscoelasticity theory provides a clearer understanding of the stress-strain time-dependent characteristics displayed by some fluids. Recently, most research has centered on the improvement of current constitutive models so as to be able to obtain a better approximation of the mechanical responses of a material under

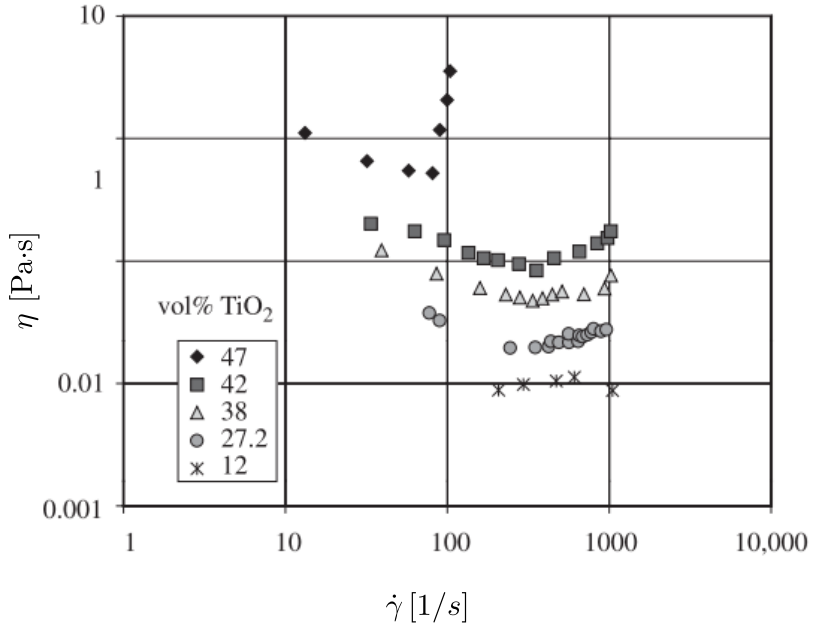


Figure 2.7: Shear viscosity data determined with a Couette rheometer. Reprinted from Flow behavior of concentrated (dilatant) suspensions, by Metzner, A. et al. [127], 1958, Transactions of The Society of Rheology , p. 243. Copyright 1958, with permission from AIP.

varying conditions.

In general, constitutive models are material-dependent and are devised using empirical evidence. These models must also hold regardless of the coordinates of reference being considered. In practice, they are applied with other physical laws to determine the solution to practical problems. Constitutive laws are very often times reduced to a basic variation and the material's property is used as the proportionality constant e.g Hooke's law. To resolve the anisotropy of the material, a tensor is required in place of the scalar variable.

Common models are discussed below.

2.3.1 Second-order fluid

This is the most basic constitutive equation with a non-zero prediction of the first normal stress difference [122]. The stress tensor is obtained as

$$\boldsymbol{\tau} = 2\eta_o \boldsymbol{D} - \psi_{1,0} \overset{\triangledown}{\boldsymbol{D}} + 4\psi_{2,0} \boldsymbol{D} \cdot \boldsymbol{D} \tag{2.6}$$

where η_o, $\psi_{1,0}$ and $\psi_{2,0}$ denote the low shear rate values of the viscosity, first and second normal stress coefficients respectively and $D = (\nabla \boldsymbol{v} + (\nabla \boldsymbol{v})^T)/2$ represents the deformation rate tensor. The upper convected time derivative of $\boldsymbol{D}$ is defined as

$$\overset{\triangledown}{\boldsymbol{D}} \equiv \frac{\partial \boldsymbol{D}}{\partial t} + \boldsymbol{v} \cdot \nabla \boldsymbol{D} - (\nabla v)^T . \boldsymbol{D} - \boldsymbol{D} \cdot \nabla \boldsymbol{v}. \tag{2.7}$$

The term, $\psi_{1,0} \overset{\triangledown}{\boldsymbol{D}}$ accounts for elastic effect (weak). The "second" in its name is associated with the number of derivatives of the velocity field from which the tensors can be formed from. The Newtonian model is derived on considering only first-order derivatives terms (i.e. $2\eta_o \boldsymbol{D}$). Coleman and Noll [124] showed in their paper that viscoelastic fluid flows that are slow, with very little fluctuation in their physical properties, satisfy Eq. (3.12). In practice, however, the fluid velocities of many polymeric

materials are faster than the Second-order fluid model can predict.

2.3.2 Upper-convected Maxwell model

Starting from the linear Maxwell equation,

$$\boldsymbol{\tau} + \lambda \frac{d\boldsymbol{\tau}}{dt} = \eta_o \dot{\boldsymbol{\gamma}}, \tag{2.8}$$

the material derivative is substituted with the upper-convected derivative of the stress tensor to obtain

$$\boldsymbol{\tau} + \lambda \overset{\triangledown}{\boldsymbol{\tau}} = 2\eta_p \boldsymbol{D} \tag{2.9}$$

Equation (2.9) is referred to as the *Upper-convected maxwell (UCM) model*. The linear Maxwell model can be reproduced from the UCM model. For example, in low-amplitude oscillatory strain, the upper convected derivative changes to the material derivative as the nonlinear terms reduces to zero. On the other hand, for a steady flow with a low deformation rate, the derivative term in Eq. (2.9) can be ignored and we thus obtain the Newtonian fluid model. In steady shear flow, the solution of the UCM model for the shear viscosity, η, the first normal stress coefficient, Ψ_1 and

second normal stress coefficient, Ψ_2, is obtained respectively as [104]

$$\eta = \eta_o \tag{2.10a}$$

$$\Psi_1 = 2\eta_o\lambda \tag{2.10b}$$

$$\Psi_2 = 0 \tag{2.10c}$$

where $\eta_o = \eta_s + \eta_p$. As shown in Eqs. (2.10), the UCM model does not predict shear thinning nor shear second normal stress differences. It predicts a constant shear first normal stress differences.

Fluids with constant viscosity that are highly elastic and highly viscous were developed initially by Boger [116]. Thus they are referred to as Boger fluids. The Boger fluid can be produced by dissolving in a highly viscous solvent, a little quantity of polymer e.g. aqueous solutions that contain little quantities of polyacrylamide dissolved in corn syrup [117]. The usefulness of Boger fluids lies in the ability to isolate viscous flow features from elastic features in a test conducted with a viscoelastic material. This is achieved by running the same test carried out with a Boger fluid on a Newtonian fluid of similar viscosity. When both fluids are subjected to identical rates of deformation, any contrast seen is due to elastic effect only. This is very helpful, since it was previously arduous to ascertain whether a non-Newtonian feature

observed during an experiment was either as a result of elasticity or shear-thinning.

Several constitutive models available can reproduce shear experimental data precisely but they perform poorly in fitting extensional rheological data. The Oldroyd-B constitutive model is well-suited in numerous cases and is given as

$$\boldsymbol{\tau} + \lambda \overset{\triangledown}{\boldsymbol{\tau}} = 2\eta_o(\boldsymbol{D} + \lambda_r \overset{\triangledown}{\boldsymbol{D}}), \tag{2.11}$$

where λ is the relaxation time and λ_r represent the retardation time. Equation (2.11) can be derived by summing the contribution of stress from both the solvent,

$$\boldsymbol{\tau}_s = 2\eta_s \boldsymbol{D} \tag{2.12}$$

and the polymer, $\boldsymbol{\tau}_p$, which is given by the Maxwell model (see Eq. (2.9)). In Eq. (2.12), η_s represents the viscosity of the solvent.

In steady shear flow, the exact solution for the shear viscosity, η, the first normal stress coefficient, Ψ_1 and second normal stress coefficient, Ψ_2, of the Oldroyd-B model is obtained respectively as [104]

$$\eta = \eta_o \tag{2.13a}$$

$$\Psi_1 = 2\eta_o(\lambda - \lambda_r) \tag{2.13b}$$

$$\Psi_2 = 0 \tag{2.13c}$$

where $\eta_o = \eta_s + \eta_p$ and $\lambda_r = \lambda\eta_s/(\eta_s + \eta_p)$. The Oldroyd-B model predicts shear first normal stress differences. However, it does not predict shear thinning.

2.3.3 Maxwell-type constitutive models

The generalization of the Upper-convected Maxwell equation enables the improvement in the accuracy of its prediction of features that are not linear with time.

There has been several propositions of nonlinear constitutive models over the years which have similar form as the Maxwell differential constitutive model. In general, they can be written as [114]

$$\overset{\triangledown}{\boldsymbol{\tau}} + \boldsymbol{m}_b + \frac{1}{\lambda}\boldsymbol{\tau} + \boldsymbol{m}_d = 2G\boldsymbol{D} \tag{2.14}$$

In Eq. (2.14), $\boldsymbol{m}_b$ adjusts the change in accumulation of stress with time while $\boldsymbol{m}_d$ functions to alter the decline of stress with time. The UCM model is recovered from Eq. (2.14) by setting $\boldsymbol{m}_b = \boldsymbol{m}_d = 0$. With the parameters, $\boldsymbol{m}_b$ and $\boldsymbol{m}_d$, nonlinear

effects like shear thinning can be easily incorporated into Eq. (2.14).

Table 2.1 enumerates popular constitutive models that can be written in the form of Eq. (2.14) and their uses. Each model is most satisfactory for specific kind of problems. The parameters, a, α, β, ξ in Table 2.1 are fit using data obtained from rheological experiments. To ensure an optimal fit with experimental data, it is most appropriate to employ a series of relaxation modes, whereby the absolute stress is then obtained as the summation of all stresses from individual modes.

In steady shear flow, the exact solution for the shear viscosity, η, and first normal stress coefficient, Ψ_1, of the Giesekus model is obtained, respectively, as [104]

$$\eta = \eta_o \left(\frac{\lambda_r}{\lambda} + \left(1 - \frac{\lambda_r}{\lambda}\right) \frac{(1-f)^2}{1 + (1-2\alpha)f} \right) \tag{2.15}$$

and

$$\Psi_1 = 2\eta_o(\lambda - \lambda_r) \frac{f(1-\alpha f)}{(\lambda\dot{\gamma})^2 \alpha (1-f)}, \tag{2.16}$$

where

$$f = \frac{1-\chi}{1 + (1-2\alpha)\chi} \tag{2.17a}$$

and

$$\chi^2 = \frac{(1 + 16\alpha(1-\alpha)(\lambda\dot{\gamma})^2)^{\frac{1}{2}} - 1}{8\alpha(1-\alpha)(\lambda\dot{\gamma})^2} \tag{2.17b}$$

In Eqs. (2.15) to (2.17), $\dot{\gamma}$ denotes the shear rate and α is a parameter that accounts for the anisotropy of the drag on polymer molecules in a fluid flow.

Table 2.1
Examples of differential constitutive models and their uses.

Models	$\boldsymbol{m}_b$	$\boldsymbol{m}_d$	Strengths/ weaknesses
White and Metzner [118]	$a\sqrt{2\boldsymbol{D}:\boldsymbol{D}}$	0	Not good for step shear flow; It produces singularities in steady elongation flows and it predicts a zero second normal stress difference. ($N_2 = 0$).
Giesekus [119]	0	$\frac{\alpha}{\lambda G}\boldsymbol{\tau}\cdot\boldsymbol{\tau}$	Very good fits in steady and transient shear; behaves poorly in extensional flows.
Larson [120]	$\frac{2\alpha}{3G}\boldsymbol{D}:\boldsymbol{\tau}(\boldsymbol{\tau}+G\boldsymbol{I})$	0	Matches data fairly well for a range of deformation. $N_2 = 0$.
Phan Thien and Thanner [121]	$\xi(\boldsymbol{D}\cdot\boldsymbol{\tau}+\boldsymbol{\tau}\cdot\boldsymbol{D})$	$\frac{1}{\lambda}\exp(\frac{\beta}{G}\mathrm{tr}\,\boldsymbol{\tau})(\boldsymbol{\tau}-\boldsymbol{I})$	Matches satisfactorily well for numerous kind of deformations; It produces unphysical oscillations at the beginning of steady shear flows.

Chapter 3

Computational Method

Computational rheology constitutes the construction and actualization of of numerical algorithms that have the potential to reproduce qualitative and quantitive features in an experimental study of viscoelastic fluid flows. A major difficulty faced by computational rheologists, engineers and fluid dynamicists is the ability to generate efficient and accurate numerical algorithms.

There has been lots of progress concerning the construction of numerical techniques and constitutive models that rely on microscopic and macroscopic methods. To solve the underlying macroscopic model of a viscoelastic fluid flow, two major steps are involved. The initial stage constitutes the discretization of the partial differential equations and second, seeking a solution to the final equation using an appropriate

procedure. Among the several methods that can be used to achieve the first process are the finite volume method, finite element method, finite difference method and spectral methods. It is important to note that when these techniques are employed for a viscoelastic model, a few changes needs to be made to account for the hyperbolicity of the fluid.

The inability to reach a converged solution beyond a certain We is a major effect of not employing suitable numerical techniques for viscoelastic models and this is not a newly accepted fact. This behavior is typically referred to as the High Weissenberg Number Problem (HWNP). The reason attributed to the cause of this oddity, as explained in many studies [69, 70, 71], is that during the discretization of the underlying partial differential equation, numerical errors are introduced which then causes the conformation tensor to lose its positive-definitiveness. A first approach to solve these problems would be to resort to refining the mesh further or employ a more accurate technique. Unfortunately, the mesh refinement strategy just worsens the scenario as a result of the hyperbolicity of the partial differential equations [71] while a more accurate approach prolongs when iterative convergence fails to hold.

A lot of progress has been made over the last decade towards obtaining stable and accurate numerical solutions to viscoelastic multiphase flow problems. In most cases, a large number of researchers have employed the standard finite difference [72, 73], finite element [74, 75] and finite volume method [76, 77, 78]. The early contributions

to computational viscoelastic fluid dynamics were in the mid nineteen seventies and they used finite difference methods [79]. Later on, studies using finite element method (FEM) began to prevail. Studies carried out with FEM became widespread as a result of its success in reproducing viscoelastic fluid flow behavior. Some examples of very popular finite element methods include the Discontinuous Galerkin method of Fortin and Fortin [74], the elastic-viscous-split-stress method of Rajagopalan et al. [80] and the explicitly elliptic momentum equation method of King et al. [75]. Today, the finite volume method (FVM) is widely used in many CFD codes and in comparison with other numerical methods, it has been shown to display very close or even better performance in relation to accuracy and robustness [71] and at a lesser computational cost [81]. In this thesis, the finite volume method is used.

In general, the finite volume method can be classified based on grid arrangement as either staggered or collocated (or cell-centered). All the dependent variables solved for and all fluid properties are stored in the control volume center for a collocated grid while on a staggered grid, the components of the velocity vector are typically arranged between two adjacent pressure points as shown in Figure 3.1.

The collocated grid is simpler to use than the staggered grid but there is a high possibility of obtaining unphysical pressure fields like the checkerboard when a pressure-velocity coupling procedure like the SIMPLE algorithm of Patankar and Spalding [82]

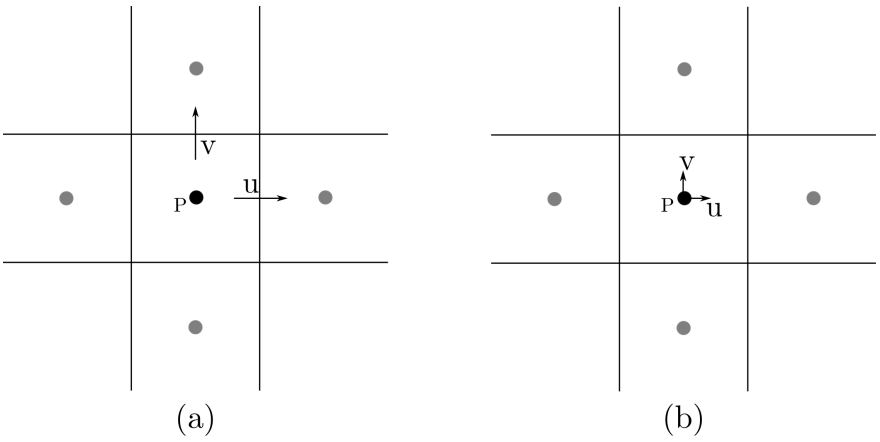


Figure 3.1: (a) Staggered grid (b) Collocated grid.

is used. This oscillating pressure field problem can be resolved by applying the Rhie-Chow interpolation [83]. Although this drawback is not seen on a staggered grid, it is difficult to use when non-orthogonal or unstructured meshes are employed.

The discretization of the convection term in the underlying partial differential equations in both procedures requires special treatment. A detailed discussion of the discretization procedure on a collocated grid is outlined in subsequent sections.

3.1 Governing equations

The immiscible two-phase fluid flow of a Newtonian and viscoelastic fluid is considered in this thesis. The flow is assumed to be incompressible and isothermal.

To capture the interface between both phases, the Volume of Fluid methododolgy

(VOF) was used. A scalar-valued function, ϕ that can take on only values in the range $[0, 1]$ is used to identify the two fluids, with $\phi \in (0, 1)$ signaling a cell that contains part of the interface while $\phi = 0$ and $\phi = 1$ connotes a control volume filled only with the continuous and dispersed phase respectively. The evolution of the scalar field, ϕ, is then typically governed by

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\boldsymbol{v}\phi) = 0. \tag{3.1}$$

To avoid numerical issues associated with the discretization of the advection term in Eq. (3.1), the Inter-gamma compressive scheme [84] is employed. With this scheme, the sharpness of the interface and monotonicity of the volume fraction field is maintained by introducing an artificial compression term into Eq. (3.1) to obtain

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\boldsymbol{v}\phi) + \nabla \cdot (\phi(1-\phi)\boldsymbol{v}_c) = 0 \tag{3.2}$$

where $\boldsymbol{v_c}$ is the difference between the velocity of the dispersed phase and the continuous phase. As is evident in Eq. (3.2), the artifical term is only active within the interface region.

To gain from the gradient smoothness around the interface of the level set method

while still eliminating mass conservation errors, the VOF method is coupled with the level set method. To begin, the level set function, ψ, is initialized with the advected volume fraction field:

$$\psi_o = \varsigma \cdot (2\phi - 1) \tag{3.3}$$

where $\varsigma = 0.75\delta x$ and δx is the cell size. Although the location of the interface matches with the iso-line, $\phi = 0.5$ and $\psi_o = 0$, the function, ψ_o, is irregular i.e. it does not satisfy ($|\nabla\psi| = 1$). Thus, it needs to be *reinitialized* by integrating the equation,

$$\psi_\tau = \text{sign}(\psi_o)(1 - |\nabla\psi|) \tag{3.4}$$

with initial state, $\psi(x, 0) = \psi_o(x)$, for only a short period of time to determine a new ψ that is regular and has the same zero level set as ψ_o [85]. To avoid abrupt changes during integration, it is recommended to use $\delta\tau = 0.1\delta x$ [86].

Next, we approximate the Dirac delta function, δ, which ensures the surface tension force is only active in a controllable neighborhood of the interface ($\epsilon = 1.5\delta x$ was used here), as

$$\delta(\psi) = \begin{cases} 0, & \text{if } |\psi| > \epsilon \\ \\ \frac{1}{2\epsilon}(1 + \cos(\pi\psi/\epsilon)), & \text{if } |\psi| \leq \epsilon \end{cases}$$

and finally the surface tension force is determined as

$$\boldsymbol{F}_s = \sigma\kappa\,\delta(\psi)\nabla\psi \tag{3.5}$$

where $\kappa = \nabla \cdot \boldsymbol{n}$ is the mean curvature, $\boldsymbol{n} = \nabla\psi/|\nabla\psi|$ is the unit normal at the interface and σ is the interfacial tension coefficient between both phases.

The conservation equations of the fluid system are given by the continuity equation:

$$\nabla \cdot \boldsymbol{v} = 0 \tag{3.6}$$

and the momentum equation:

$$\frac{\partial(\rho\boldsymbol{v})}{\partial t} + \nabla \cdot (\rho\boldsymbol{v}\boldsymbol{v}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \rho\boldsymbol{g} + \boldsymbol{F}_s \tag{3.7}$$

where $\boldsymbol{\tau} = \boldsymbol{\tau}_s + \boldsymbol{\tau}_p$ is the extra-stress tensor, $\boldsymbol{\tau}_s$ represents the solvent contribution to

stress and satisfies the Newtonian constitutive law:

$$\boldsymbol{\tau}_s = 2\eta_s \boldsymbol{D} \tag{3.8}$$

where η_s is the viscosity of the solvent, $\boldsymbol{D}$ denotes the deformation rate tensor, defined by

$$\boldsymbol{D} = \frac{1}{2}\left(\nabla \boldsymbol{v} + (\nabla \boldsymbol{v})^T\right) \tag{3.9}$$

and the polymer stress, $\boldsymbol{\tau}_p$, is governed by any of the stress constitutive law described in the previous chapter. For example, the Oldroyd-B constitutive equation is given by

$$\boldsymbol{\tau}_p + \lambda \overset{\triangledown}{\boldsymbol{\tau}}_p = 2\eta_p \boldsymbol{D} \tag{3.10}$$

and the Giesekus constitutive equation is given by

$$\boldsymbol{\tau}_p + \lambda \overset{\triangledown}{\boldsymbol{\tau}}_p + \alpha \frac{\lambda}{\eta_p}(\boldsymbol{\tau}_p \cdot \boldsymbol{\tau}_p) = 2\eta_p \boldsymbol{D} \tag{3.11}$$

where λ is the relaxation time, η_p denotes the zero-shear-rate polymeric viscosity, α is a parameter that accounts for the anisotropy of the drag on polymer molecules in fluid flow, and $\overset{\triangledown}{\boldsymbol{\tau}}_p$ represents the upper convected time derivative of $\boldsymbol{\tau}_p$ defined as

$$\overset{\triangledown}{\boldsymbol{\tau}}_p \equiv \frac{\partial \boldsymbol{\tau}_p}{\partial t} + \boldsymbol{v} \cdot \nabla \boldsymbol{\tau}_p - (\nabla v)^T . \boldsymbol{\tau}_p - \boldsymbol{\tau}_p \cdot \nabla \boldsymbol{v}. \tag{3.12}$$

In Eqs. (3.7) to (3.12), $\boldsymbol{v}$ represents the velocity, p is the pressure, and $\boldsymbol{g}$ is the gravitational acceleration. The physical properties of fluids used in the equations are obtained as

$$\zeta = \phi\zeta_d + (1 - \phi)\zeta_c, \tag{3.13}$$

where ζ_d and ζ_c represents a generic property of the dispersed and continuous phase respectively.

3.2 Finite-volume discretization

Suppose ψ and $\boldsymbol{\psi}$ are scalar and vector-valued functions defined on an arbitrary control volume with volume, $\mathbb{V}$. In the discretizations that follow, Gauss' divergence

theorem is used to convert the volume intervals to surface integrals, which are then approximated using the second-order Gauss one-point (centroidal) scheme on each face. For example,

$$\int_{\mathbb{V}} \nabla \cdot \boldsymbol{\psi} d\mathbb{V} \simeq \sum_f \boldsymbol{S}_f \cdot \boldsymbol{\psi}_f \qquad (\nabla \cdot \boldsymbol{\psi})_p \simeq \frac{1}{\mathbb{V}_p} \sum_f \boldsymbol{S}_f \cdot \boldsymbol{\psi}_f$$

$$\int_{\mathbb{V}} \nabla \psi d\mathbb{V} \simeq \sum_f \boldsymbol{S}_f \psi_f \qquad (\nabla \psi)_p \simeq \frac{1}{\mathbb{V}_p} \sum_f \boldsymbol{S}_f \psi_f.$$

Where p denotes the centroid of the control volume, f denotes the centroid of each cell face, $\boldsymbol{S}_f = \hat{\boldsymbol{n}} A$ is the face area vector that point outwards from the cell face, $\hat{\boldsymbol{n}}$ is the unit normal vector at the face pointing outwards, A is the area of the face, $\mathbb{V}_p$ is the volume of the cell with centroid at p and $\sum_f$ denotes the summation over all faces for a particular cell.

3.2.1 Momentum equation

To enhance the numerical stability of the momentum equation, the DEVSS [87] technique is applied. This involves the addition of an elliptic term to both sides of the momentum equation - the one on the left (-$\nabla \cdot (\epsilon \nabla \boldsymbol{v})$) contributes to the coefficient matrix once the equation is discretized and the other (-$\nabla \cdot (\epsilon \nabla \boldsymbol{v})$) to the source term. We obtain

$$\frac{\partial(\rho \boldsymbol{v})}{\partial t}+\nabla\cdot(\rho \boldsymbol{v}\boldsymbol{v})-\nabla\cdot((\epsilon+\eta_s)\nabla \boldsymbol{v}) = -\nabla p+\nabla \boldsymbol{v}\cdot\nabla\eta_s+\nabla\cdot\boldsymbol{\tau}_p+\rho \boldsymbol{g}+\sigma\kappa\delta\nabla\psi-\nabla\cdot(\epsilon\nabla \boldsymbol{v}) \tag{3.14}$$

where Eq. (3.8) and Eq. (3.9) have been used in Eq. (3.7) and $\kappa = \nabla \cdot (\nabla\psi/|\nabla\psi|)$.

Taking the integral over the control volume and time step, $\triangle t$, we obtain

$$\int_{\mathbb{T}}\int_{\mathbb{V}}\left[\frac{\partial(\rho \boldsymbol{v})}{\partial t}+\nabla\cdot(\rho \boldsymbol{v}\boldsymbol{v})-\nabla\cdot((\epsilon+\eta_s)\nabla \boldsymbol{v})\right]\mathrm{d}\mathbb{V}\mathrm{d}\mathbb{T} = \int_{\mathbb{T}}\int_{\mathbb{V}}\left[-\nabla p+\nabla \boldsymbol{v}\cdot\nabla\eta_s+\nabla\cdot\boldsymbol{\tau}_p+\rho \boldsymbol{g}+\sigma\kappa\delta\nabla\psi-\nabla\cdot(\epsilon\nabla \boldsymbol{v})\right]\mathrm{d}\mathbb{V}\mathrm{d}\mathbb{T} \tag{3.15}$$

$\mathbb{T} := [t, t+\triangle t]$.

Using $\epsilon = \eta_p$ [88], the momentum equation in semi-discrete form is then given by

$$\int_{\mathbb{T}}\left[\left(\frac{\partial\rho \boldsymbol{v}}{\partial t}\right)_p\mathbb{V}_p+\sum_f F\boldsymbol{v}_f-\sum_f(\eta_o)_f\boldsymbol{S}_f\cdot(\nabla \boldsymbol{v})_f\right]\mathrm{d}\mathbb{T} = \int_{\mathbb{T}}\Bigg[\,(-(\nabla p)_p+(\nabla \boldsymbol{v})_p\cdot(\nabla\eta_s)_p +\rho_p \boldsymbol{g}+\sigma\kappa_p\delta_p(\nabla\psi)_p)\,\mathbb{V}_p+\sum_f \boldsymbol{S}_f\cdot(\boldsymbol{\tau}_p)_f-\sum_f(\eta_p)_f\boldsymbol{S}_f\cdot(\nabla \boldsymbol{v})_f\Bigg]\mathrm{d}\mathbb{T} \tag{3.16}$$

where $F = \rho \boldsymbol{S}_f \cdot \boldsymbol{v}_f$.

3.2.2 Temporal discretization

On using the Euler implicit scheme for time discretization, we obtain the following equation for the (guessed) velocity, $\hat{\boldsymbol{v}}_p$, at time $t_{n+1} = t_n + \triangle t$

$$
\begin{aligned}
[\rho_p^{n+1}\hat{\boldsymbol{v}}_p + (\rho \boldsymbol{v})_p^n]\mathbb{V}_p + \left[\sum_f F\hat{\boldsymbol{v}}_f - \sum_f (\eta_o)_f^{n+1}\boldsymbol{S}_f \cdot (\nabla \hat{\boldsymbol{v}})_f\right]\triangle t = \Big[- (\nabla p)_p \\
+ (\rho_p \boldsymbol{g})^{n+1} + (\sigma \kappa_p \delta_p (\nabla \psi)_p)^{n+1} + \nabla \boldsymbol{v}_p^n \cdot \nabla (\eta_s)_p^{n+1}\Big]\mathbb{V}_p \triangle t \\
+ \left(\sum_f \boldsymbol{S}_f \cdot (\boldsymbol{\tau}_p)_f^n - \sum_f (\eta_p)_f^{n+1}\boldsymbol{S}_f \cdot (\nabla \boldsymbol{v})_f^n\right)\triangle t \quad (3.17)
\end{aligned}
$$

where $\zeta^n = \zeta(t)$ and $\zeta^{n+1} = \zeta(t + \triangle t)$. On multiplying both sides of Eqn. (3.17) by

$1/(\mathbb{V}_p \triangle t)$, we obtain

$$\frac{\rho_p^{n+1}\hat{\boldsymbol{v}}_p}{\triangle t} + \frac{1}{\mathbb{V}_p}\sum_f F\hat{\boldsymbol{v}}_f - \frac{1}{\mathbb{V}_p}\sum_f (\eta_o)_f^{n+1}\boldsymbol{S}_f \cdot (\nabla\hat{\boldsymbol{v}})_f = -(\nabla p)_p + \underbrace{\frac{(\rho\boldsymbol{v})_p^n}{\triangle t} + (\rho_p\boldsymbol{g})^{n+1} + (\sigma\kappa_p\delta_p(\nabla\psi)_p)^{n+1} + \nabla\boldsymbol{v}_p^n \cdot \nabla(\eta_s)_p^{n+1}}_{:=\Upsilon_1} + \underbrace{\frac{1}{\mathbb{V}_p}\left(\sum_f \boldsymbol{S}_f \cdot (\boldsymbol{\tau}_p)_f^n - \sum_f (\eta_p)_f^{n+1}\boldsymbol{S}_f \cdot (\nabla\boldsymbol{v})_f^n\right)}_{:=\Upsilon_2} \quad (3.18)$$

which is equivalent to

$$\mathtt{a}_p\hat{\boldsymbol{v}}_p = \sum_N \mathtt{a}_N\hat{\boldsymbol{v}}_N + \Upsilon - (\nabla p)_p \quad (3.19)$$

where we note that the quantities estimated at the face can be written in terms of its value in a neighborhood cell and the current cell using an appropriate scheme and $\Upsilon = \Upsilon_1 + \Upsilon_2$ involves quantities at times, t_n and t_{n+1}. Equation (3.19) is a linear system of equations for predictor velocity at time, t_{n+1}, and $\sum_N$ represents the summation over neighborhood cells.

3.2.3 Pressure equation

To derive the pressure equation, Eq. (3.19) is re-written as

$$\mathsf{a}_p \hat{\boldsymbol{v}}_p = \mathbb{H}(\boldsymbol{v}^i)_p - (\nabla p)_p \tag{3.20}$$

where $\mathbb{H}(\boldsymbol{v}^i) = \sum_N \mathsf{a}_N \hat{\boldsymbol{v}}_N + \Upsilon$ and the index, i, indicates the current value within the PISO [89] iteration. Therefore,

$$\hat{\boldsymbol{v}}_p = \frac{\mathbb{H}(\boldsymbol{v}^i)_p}{\mathsf{a}_p} - \frac{1}{\mathsf{a}_p}(\nabla p)_p \qquad \text{and} \tag{3.21}$$

$$\hat{\boldsymbol{v}}_f = \left(\frac{\mathbb{H}(\boldsymbol{v}^i)}{\mathsf{a}_p}\right)_f - \left(\frac{1}{\mathsf{a}_p}\right)_f (\nabla p)_f \tag{3.22}$$

Now, the finite volume discretization of Eq. (3.6) gives the discrete continuity equation,

$$\sum_f \boldsymbol{S}_f \cdot \hat{\boldsymbol{v}}_f = 0. \tag{3.23}$$

Assuming the continuity equation is satisfied by $\hat{\boldsymbol{v}}_p$ in Eq. (3.21), then using Eq. (3.22) in Eq. (3.23), we obtain the discretized pressure equation:

$$\sum_f \left(\frac{1}{\mathsf{a}_p}\right)_f \boldsymbol{S}_f \cdot (\nabla p)_f = \sum_f \boldsymbol{S}_f \cdot \left(\frac{\mathbb{H}(\boldsymbol{v}^i)}{\mathsf{a}_p}\right)_f \tag{3.24}$$

Equation (3.22) is also used to determine the volumetric flux,

$$F = \boldsymbol{S}_f \cdot \hat{\boldsymbol{v}}_f \tag{3.25}$$

$$= \boldsymbol{S}_f \cdot \left[\left(\frac{\mathbb{H}(\boldsymbol{v}^i)}{\mathtt{a}_p} \right)_f - \left(\frac{1}{\mathtt{a}_p} \right)_f (\nabla p)_f \right] \tag{3.26}$$

which is guaranteed to be conservative [90].

$$F = \boldsymbol{S}_f \cdot \left[\left(\frac{\mathbb{H}(\boldsymbol{v}^i)}{\mathtt{a}_p} \right)_f - \left(\frac{1}{\mathtt{a}_p} \right)_f (\nabla p)_f \right] \tag{3.27}$$

3.2.4 Discrete form of VOF equation

Starting from Eqn. (3.2), we take the integral over the control volume and time step, $\triangle t$, to obtain

$$\int_{\mathbb{T}} \int_{\mathbb{V}} \left(\frac{\partial \phi}{\partial t} \right) d\mathbb{V} d\mathbb{T} + \int_{\mathbb{T}} \int_{\mathbb{V}} \nabla \cdot (\boldsymbol{v}\phi)\, d\mathbb{V} d\mathbb{T} + \int_{\mathbb{T}} \int_{\mathbb{V}} \nabla \cdot (\phi(1-\phi)\boldsymbol{v}_c)\, d\mathbb{V} d\mathbb{T} = 0 \tag{3.28}$$

The semi-discrete form of the volume of fluid equation then gives

$$\int_{\mathbb{T}} \left(\frac{\partial \phi}{\partial t}\right)_p \mathbb{V}_p \, d\mathbb{T} + \int_{\mathbb{T}} \sum_f \boldsymbol{S}_f \cdot (\boldsymbol{v}\phi)_f \, d\mathbb{T} + \int_{\mathbb{T}} \sum_f \boldsymbol{S}_f \cdot (\phi(1-\phi)\boldsymbol{v}_c)_f \, d\mathbb{T} = 0 \quad (3.29)$$

or

$$\int_{\mathbb{T}} \left(\frac{\partial \phi}{\partial t}\right)_p \mathbb{V}_p \, d\mathbb{T} + \int_{\mathbb{T}} \sum_f F\phi_f \, d\mathbb{T} + \int_{\mathbb{T}} \sum_f F_c(\phi(1-\phi))_f \, d\mathbb{T} = 0 \quad (3.30)$$

where $F = \boldsymbol{S} \cdot \boldsymbol{v}_f$ is the face volume flux. To discretize the artificial compression term, the maximum of the velocity magnitude within the neighborhood of the interface and its direction is used in constructing the relative velocity at the cell face [91] and is obtained as

$$F_c = (\boldsymbol{n}_f \cdot \boldsymbol{S}_f) \min\left[C_\phi \frac{|F|}{|\boldsymbol{S}_f|}, \max\left(\frac{|F|}{\boldsymbol{S}_f}\right)\right] \quad (3.31)$$

where

$$\boldsymbol{n}_f = \frac{(\nabla\phi)_f}{|(\nabla\phi)_f + \delta_n|} \quad (3.32)$$

where δ_n is the stablilization parameter [91] which accounts for the non-uniformity of the grid. This parameter usually takes the value of 10^{-5}. The constant C_ϕ is a user-specified value that functions to regulate the smearing of the interface; $C_\phi = 1$

has been used in all simulations in this work. Finally, time discretization yields

$$\mathbb{V}_p(\phi_p^{n+1} - \phi_p^n) + \sum_f \Big[(1-\lambda)(F\phi_f)^{n+1} + \lambda(F\phi_f)^n\Big] + \sum_f \Big[(1-\lambda)(F_c[\phi(1-\phi)]_f)^{n+1} + \lambda(F_c[\phi(1-\phi)]_f)^n\Big]. \quad (3.33)$$

In OpenFOAM®, λ is set as 1 by default; $\lambda = 1$ and 0 connotes the Euler implicit and explicit scheme respectively.

3.2.5 Discrete form of re-initialization equation

The discretization of the re-initialization of the level set function in Eq. (3.4) is achieved by applying explicit Euler in time and approximating the gradient term using the Gauss centroidal scheme outlined earlier to obtain

$$\psi_p^{n+1} = \psi_p^n + \triangle t\left[\frac{\psi_o}{|\psi_o|}\left(1 - \frac{1}{\mathbb{V}_p}\left|\sum_f \boldsymbol{S}_f \psi_f^n\right|\right)\right]. \quad (3.34)$$

3.3 Adaptive local mesh refinement

The general methodology of the refinement/unrefinement process can be described as follows:

1. An initial computational grid is set up.
2. The governing equations are solved on the base grid (original grid) See Eqs. (3.2, 3.4, 3.6, 3.7, 3.10).
3. Cells of the base grid within some user-specified range of the volume fraction ϕ (around the interface) are selected.
4. Selected cells are then refined locally via cell splitting.
5. Cells that have volume fraction values outside the range in (3) may be coarsened.
6. The current numerical result is now mapped to the refined mesh as an initial guess for the next computation. Further explanation on these is given below.

3.3.1 Data structure

Modifications in 2d planar and axisymmetric geometry were done by Baniabedalruhman [27] and have been used in `clsVeInterDymFoam`.

The mesh adaptation procedure begins with a base grid. Throughout the simulation, nodes can be inserted or deleted from the original mesh and the refinement changes are stored using a hierarchical quad tree data structure *(see Figure 3.2)*. Here, each *parent* cell tagged for refinement is split isotropically into four *daughter* cells. These newly formed cells are then added to the quadtree at a position which is one-lower than the parent cell in the hierarchy. All cells in the mesh have a pointer to its parent cell (if it is a result of refinement) and pointers to its daughter cells (if it has undergone division). The unrefinement process involves the reconstruction of the parent cell from its daughter cells.

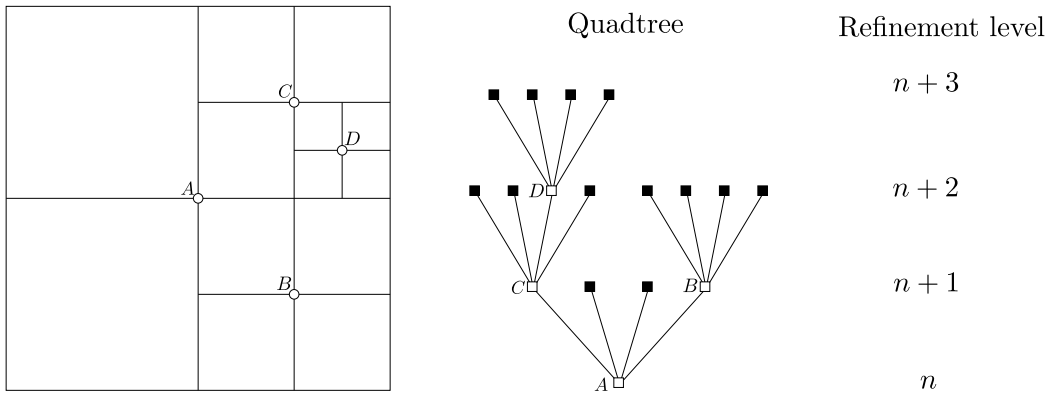


Figure 3.2: A two-dimensional grid and its quadtree representation.

3.3.2 Mesh refinement and unrefinement

Given an initial base grid, the refinement process starts by selection of candidate cells for refinement and unrefinement based on the volume fraction, ϕ.

The change in size of two neighboring cells should be maintained as either one or two to ensure smooth grading. For clarity, a node which is a vertex to all of its neighboring cells is called a 'regular node'; otherwise, it is 'irregular'. An n-irregular mesh is a mesh with a maximum of 'n' irregular nodes for all cells. Figure 3.3 shows examples of regular and irregular meshes. The limitation highlighted above is the acceptance of only 1-irregular meshes. In other words, each cell can have at most two neighbors over each of its faces; otherwise the cell is subdivided as shown in Figure 3.4.

A cell tagged for division becomes a parent cell and consequently, new nodes, edges and faces are added to its interior. The connectivity information is then updated and the daughter cells stored for later coarsening.

Cells added during the simulation are stored in a quad tree data structure. Thus unrefinement, merely involves the deletion of newly-added cells. This can also be thought of, literally, as an 'undo' operation.

The coarsening process begins with scanning of cells that have been selected for

unrefinement recursively in decreasing order of their refinement levels. Next, cell-pairs that satisfy the regularity condition discussed above are formed and merged.

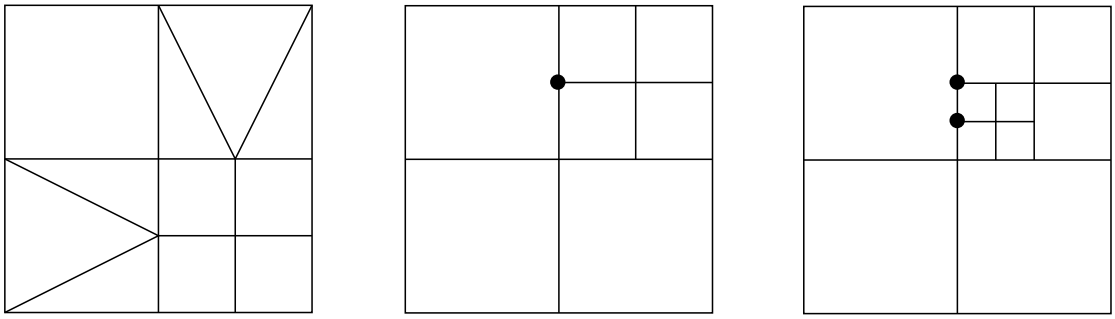


Figure 3.3: (a) Regular mesh (b) 1-irregular mesh (c) 2-irregular mesh

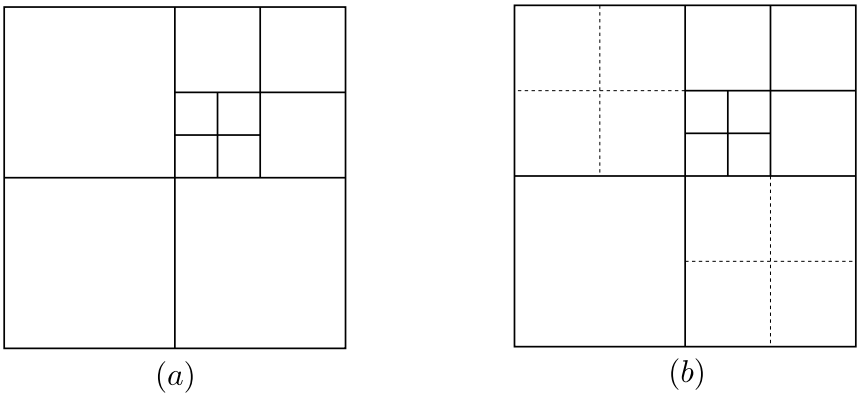


Figure 3.4: Irregular meshes (a) 2-irregular mesh (b) 2-irregular mesh showing refinements to be made to make the original mesh 1-irregular.

3.3.3 Solution mapping

After appropriate changes are made to the mesh due to the refinement procedure, the solution on the previous mesh is used as an initial guess on the refined mesh for the next calculation iterate. This is achieved on the assumption that the variation of the variables on each cell is linear. Thus, field values at the center of each cell of the refined mesh are obtained by finding the closest point, B, on the previous grid *(see Figure 3.5)* and applying the second-order linear approximation below:

$$\zeta_P = \zeta_B + (\boldsymbol{x}_P - \boldsymbol{x}_B) \cdot (\nabla\zeta)_B \tag{3.35}$$

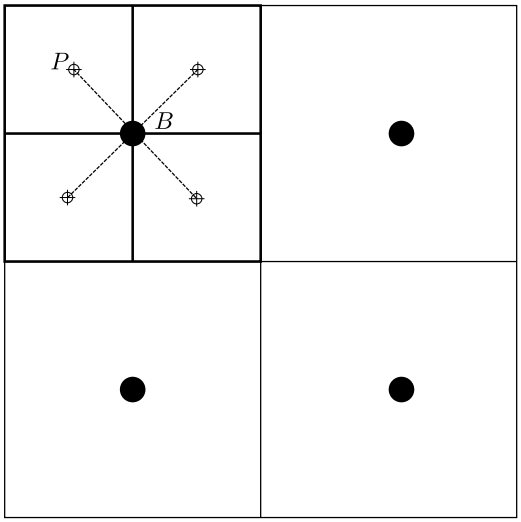


Figure 3.5: Mapping of variables between cells of different refinement level.

3.3.4 Treatment of fluxes

Appropriate treatment of the overlapping interface between a fine and coarse grid is required to ensure the continuity constraint (Eqn. (3.23)) is satisfied. As a result of the accumulation of errors during discretization, not all cells satisfy this constraint especially coarse cells next to a coarse-fine interface. The refinements of cells locally poses a challenge on the correct handling of the coupling coefficients on the coarse-fine interface. In addition to this, when the flux on the coarse grid is corrected, it could result in a flux imbalance.

To resolve the afore-mentioned issues, the approach adopted in OpenFOAM [90] is to solve the pressure equation on the new mesh and then recalculate the flux using the new pressure, before resuming the computation for the current time-step on the new mesh. Therefore, we solve

$$\nabla \cdot \left(\frac{1}{\mathtt{a}_p} \nabla p \right) = \sum_f \boldsymbol{S}_f \cdot \left(\frac{\mathbb{H}(\boldsymbol{v})}{\mathtt{a}_p} \right)_f \tag{3.36}$$

for pressure and the flux,

$$F = S \cdot \left[\left(\frac{\mathbb{H}(\boldsymbol{v})}{\mathtt{a}_p} \right)_f - \left(\frac{1}{\mathtt{a}_p} \right)_f (\nabla p)_f \right] \tag{3.37}$$

is evaluated again using the interpolated values of $\mathtt{a}_p$ and $\mathbb{H}(\boldsymbol{v})$ on the new mesh. Since Eq.(3.36) was obtained from the continuity equation, the fluxes computed with this new pressure are conservative.

3.4 Solution algorithm

The simulation begins with a base mesh and then continues from `step(2)`. For other time steps, the numerical algorithm begins at `step(1)`.

1‡ The mesh is updated based on the value of the volume fraction, ϕ, in each cell and the flux is recalculated (see Eq. (3.37)) to ensure it is conservative.

2. Given the initial values of ϕ and $\boldsymbol{v}$, a new volume fraction field, ϕ^{n+1}, is obtained by solving Eq. (3.33).

3‡ The level set function, ψ is initialized using the current volume fraction field, ϕ^{n+1} (see Eq. (3.3))

4‡ ψ is then re-initialized by solving Eq. (3.34).

5‡ The Dirac function and curvature are now evaluated and used to estimate the

volumetric surface tension force. (see Eq. 3.5)

6. The values of the physical properties in each cell, ρ and η are updated using the new ϕ^{n+1} (see Eq. (3.13)).

7. The momentum equation is solved implicitly to predict the velocity field, $\hat{\boldsymbol{v}}$, using values of velocity, $\boldsymbol{v}$, stress, $\boldsymbol{\tau}$ and pressure, p from the previous time step (see Eq. (3.21)).

8. $\mathbb{H}(\boldsymbol{v})$ is constructed using the new velocity, $\hat{\boldsymbol{v}}$, and a new pressure field, $\hat{p}$, is obtained by solving Eq. (3.24).

9. The fluxes, F, and $\hat{\boldsymbol{v}}$ are corrected using Eqs. (3.27) and (3.21) respectively.

10. `Steps(8)` and `(9)` are repeated for a fixed number of times.

11. The newly obtained conservative fluxes are then used to solve the stress equation to obtain the final $\hat{\boldsymbol{\tau}}$.

12. `Steps(2)`–`(11)` can be repeated as many times as desired before moving to the next time step. In this study, the loop was iterated only once.

The steps enumerated above with the symbol (‡) are the modifications made to the algorithmic process of the original solver, `viscoelasticInterFoam`.

3.5 Solution of system of linear equations

In general, the linear system of equations obtained after discretizing the transport equations can be written for each cell P as

$$a_P \zeta_P^{n+1} + \sum_N a_N \zeta_N^{n+1} = b_P.$$

In matrix form, we can drop the indices and summation and obtain the following system

$$A\zeta^{n+1} = b,$$

where ζ^{n+1} and b are N-dimensional vectors and A is a sparse $N \times N$-dimensional matrix. The sparseness of A is due to the contributions of its off-diagonal non-zero coefficients from only adjacent cells.. The system of equations is typically not solved with direct methods but iteratively. Iterative solvers are more efficient because they exploit the sparsity of A, thereby reducing memory requirements.

Throughout this work, the discrete volume fraction equation is solved using the multi-dimensionsal limiter for explicit solution (MULES) method of OpenFOAM®. Next, the first pressure equation is solved with a geometric-algebraic multi-grid (GAMG)

solver in conjunction with a GAMG preconditioner until either the absolute tolerance is below 10^{-8} or the relative tolerance is below 0.01. The GAMG solver with GAMG preconditioner was also used for solving the second pressure equation but an absolute tolerance of 10^{-8} is used to determine convergence. The GAMG solver was used here since it is known to be very fast in initially decreasing the residual. Lastly, the GaussSiedel method is used to solve the discrete stress and momentum equations. The absolute tolerance here was set to 10^{-6}. An example of how these settings are applied to the linear solvers in OpenFOAM is shown in Fig. 3.6.

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{
    "alpha.phase1.*"
    {
        nAlphaCorr      1;
        nAlphaSubCycles 3;
        cAlpha          1;
    }
    p_rgh
    {
        solver          GAMG;
        tolerance       1e-08;
        relTol          0.01;
        smoother        DIC;
        nPreSweeps      0;
        nPostSweeps     2;
        nFinestSweeps   2;
        cacheAgglomeration false;
        nCellsInCoarsestLevel 10;
        agglomerator    faceAreaPair;
        mergeLevels     1;
    }
    p_rghFinal
    {
        $p_rgh;
        relTol          0;
    }
    "pcorr.*"
    {
        $p_rghFinal;
        tolerance       0.0001;
    }
    U
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        tolerance       1e-06;
        relTol          0;
        nSweeps         1;
    }
    tau
      {
        solver          smoothSolver;
        smoother        GaussSeidel;
        tolerance       1e-06;
        relTol          0;
        nSweeps         1;
      }
}
PIMPLE
{
    momentumPredictor no;
    nCorrectors     3;
    nNonOrthogonalCorrectors 0;
    pdRefCell       0;
    pdRefValue      0;
    pRefCell        0;
    pRefValue       0;
}

// ************************************************************************* //
```

Figure 3.6: An example of parameter settings for linear solvers. This is an `fvSolution` file of OpenFOAM® (without header).

Chapter 4

Benchmark problems

In this chapter, we investigate the robustness and accuracy of the newly developed solver, `clsVeInterDymFoam`, by examining the following benchmark cases: Two dimensional drop under static conditions (i.e. static rod case); Drop deformation in simple shear flow for the cases where the drop is Newtonian and the continuous phase is viscoelastic, and vice versa and Rise of a Newtonian drop in a Giesekus fluid. The method is further validated by comparing simulation predictions of drop formation in a t-junction to results from experimental studies [1]. For the static rod case, the viscoelastic fluid is modeled by the Oldroyd-B constitutive equation while the viscoelastic fluid used in the other cases are modeled as a Giesekus fluid.

For easy referencing, the properties of all fluids considered in this chapter is delineated

in Table 4.1. N1, N2 and N3 represent Newtonian fluids, VE2, VE3 and VE6 signify Oldroyd-B fluids while VE1, VE4 and VE5 represent Giesekus fluids. The symbols η_s and η_p respectively connote the solvent and polymeric viscosity of the corresponding viscoelastic fluid.

Table 4.1
Material properties of the fluids used in all test cases.

Property	N1	N2	N3	N4	N5	VE1	VE2	VE3	VE4	VE5	VE6
Density (kg/m^3)	1	1000	984	1000		1000					
Viscosity (mPa·s.)	0.01	1	69.5	12.1	44.1	$\eta_s = 6.05$, $\eta_p = 6.05$			$\eta_s = 0.001$, $\eta_p = 1.049$	$\eta_s = 0.001$, $\eta_p = 5.919$	$\eta_s = 0.9$, $\eta_p = 0.1$
Relaxation time (s)	-	-	-	-	-	0.25	0.25	100	0.034	0.152	0.01
Mobility factor	-	-	-	-	-	0.003	-	-	0.003	0.003	-

4.1 2D Drop under static conditions

Brackbill's continuum surface force (CSF) technique [1] is commonly used to model surface tension force in multiphase flows, especially when the grid is Eulerian. This method employs the use of delta functions to reconstruct the surface tension forces as volume forces around the interface i.e. it is only active within the interface. A major weakness with the CSF method is that in surface tension-driven flows, it induces artificial velocities around the interface. This anomaly has been attributed to the numerical imbalance between the pressure gradient and the corresponding surface tension force [92].

There has been numerous recommendations in literature that concerns possible ways by which these spurious currents can be minimized. Some of these include improving the evaluation of curvature [93, 94] and enhancing the flow algorithm [95, 96].

In this section, the performance of the new method is assessed by considering the equilibrium rod problem. A similar example has been used by Albadawi et al. [86]. It involves the investigation of the pressure jump across the interface of a two-dimensional bubble that is initially fixed at the center of the domain neglecting gravity forces and the velocity in the whole domain is initialized as zero. For the bubble to remain static, an exact discrete balance between the pressure gradient and surface tension

force is expected.

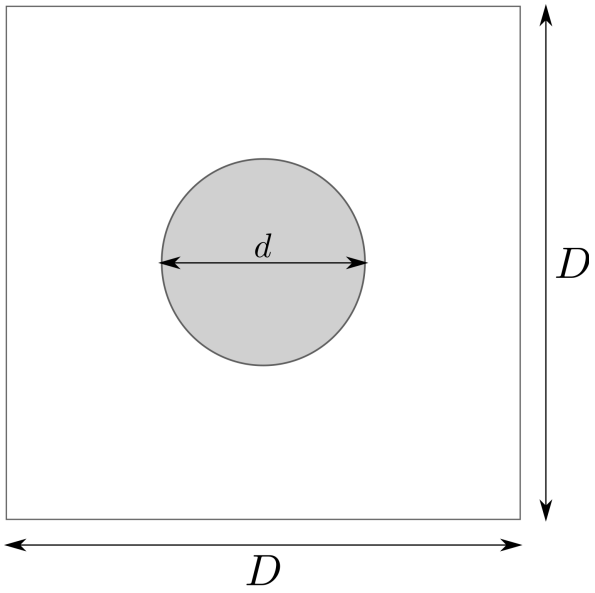


Figure 4.1: Schematic representation of static drop. $D = 0.1$m, $d = 0.01$m.

The numerical domain (as shown in Fig. 4.1) is initialized with a bubble (drop phase) of diameter, $d = 0.01m$ at the center and the surrounding is filled with water. The parameter values employed for both the bubble and continuous phase were adapted from Albadawi et al. [86]. The density of the bubble (fluid N1 in Table 4.1) and continuous phase (fluid N2 in Table 4.1) is given respectively by $\rho_d = 1\text{kg}/m^3$ and $\rho_c = 1000\text{kg}/m^3$; the dynamic viscosities are $\eta_d = 10^{-5}$Pa·s and $\eta_c = 10^{-3}$Pa·s. The interfacial tension was taken to be $\sigma = 0.01$N/m. Using the Laplace - Young law, we can obtain the analytical jump in pressure across the interface to be equal to $\triangle P_{anal} = 2\sigma/d = 2$Pa.

Table 4.2
Pressure jump across the interface.

Mesh	Number of cells	$\triangle x/d$	$\triangle P_{vof}$	$\triangle P_{lsvof2}$	$\triangle P_{lsvof4}$	$\triangle P_{ve}$	E_{vof}	E_{lsvof2}	E_{lsvof4}	E_{ve}
1	10000	0.1	1.779	1.953	2.066	2.064	0.111	0.023	0.033	0.032
2	40000	0.05	1.754	1.911	1.984	1.978	0.123	0.044	0.0082	0.011
3	160000	0.025	1.749	1.899	1.964	1.95	0.126	0.051	0.018	0.025

Table 4.2 shows a comparison among the pressure difference ($\triangle P_i$) obtained using the VOF and the LS-VOF method on three meshes. $\triangle P_{lsvof2}$ and $\triangle P_{lsvof4}$ connote the pressure difference obtained using second order Gaussian integration and fourth order least squares method respectively. Refined meshes are obtained from the base mesh *(mesh 1)* upon refinement in both x- and y- direction by a factor of 2. The cell size is indicated by $\triangle x/d$ in the table. The relative errors incurred with each method were determined using

$$E_i = \left| \frac{\triangle P - \triangle P_{anal}}{\triangle P_{anal}} \right|,$$

where $i(= vof, lsvof2, lsvof4, ve)$ is associated with the different cases considered as illustrated in Table 4.2. All results shown were taken at $t = 3.0$s when steady state was reached. We observe that the fourth order least squares method converges to a value closer to the exact solution that the second order Gaussian integration. The improved accuracy in the pressure jump seen with the new method can be attributed to refinement made in the evaluation of curvature via the level-set method using the fourth order least square method to evaluate $\nabla\psi$. We also note in Table 4.2 that for each case, the relative change in pressure drop between two consecutive meshes decreases with mesh refinement.

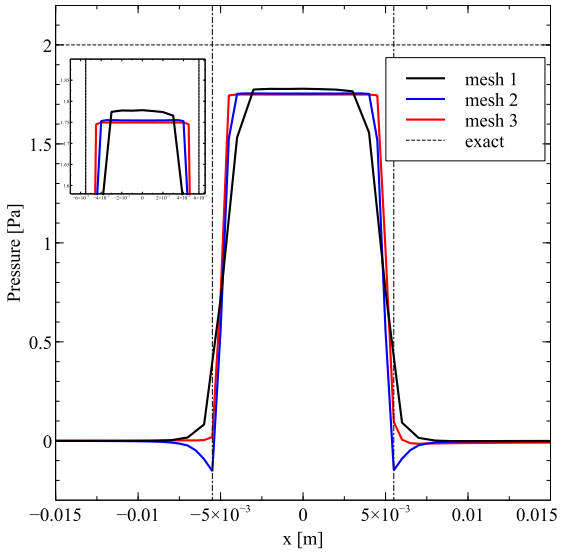


Figure 4.2: Computed pressure along the x-direction through the center of the static bubble with VOF model. The horizontal dotted line signifies the exact solution and the vertical dotted lines signify the boundary of the bubble on the $x-$axis.

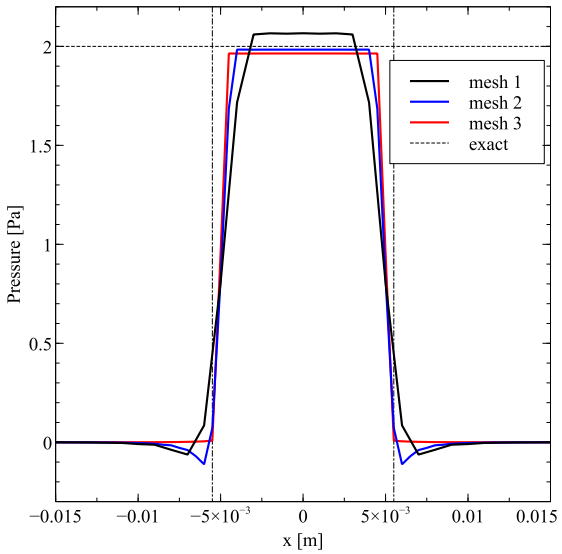


Figure 4.3: Computed pressure along the x-direction through the center of the static bubble with LS-VOF model using the fourth-order least squares method. The horizontal dotted line signifies the exact solution and the vertical dotted lines signify the boundary of the bubble on the $x-$axis.

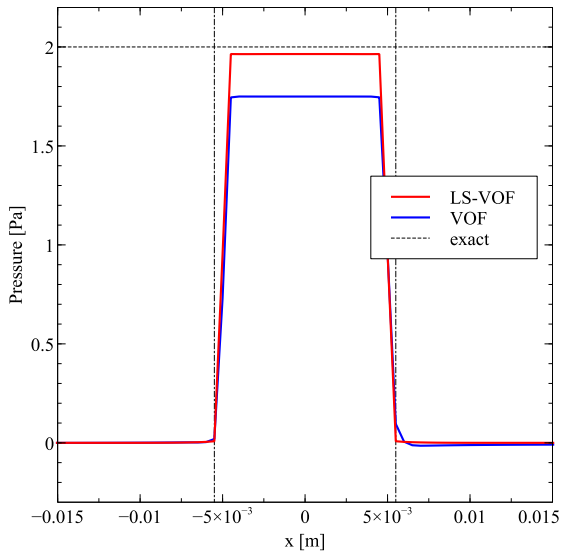


Figure 4.4: Comparison of pressure jump on *mesh 3*. The horizontal dotted line signifies the exact solution and the vertical dotted lines signify the boundary of the bubble on the $x-$axis.

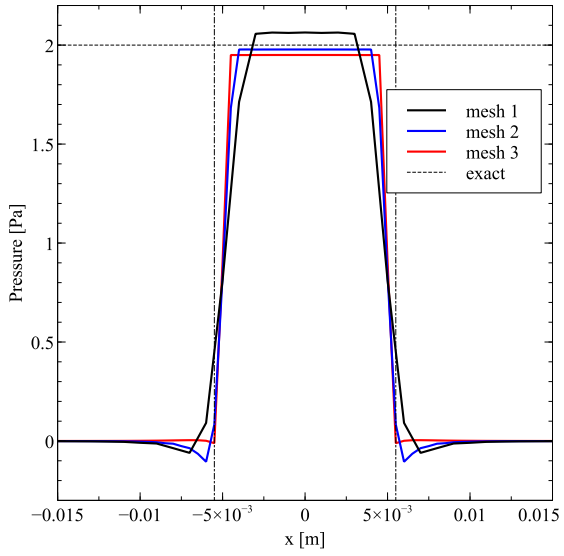


Figure 4.5: Computed pressure along the x-direction through the center of the static bubble in an Oldroyd-B continuous phase using the LS-VOF model using fourth-order least squares method. The horizontal dotted line signifies the exact solution and the vertical dotted lines signify the boundary of the bubble on the $x-$axis.

Figure 4.2 shows the pressure along a horizontal line through the center of the drop on *mesh1*, *mesh2* and *mesh3* using the VOF approach of `interFoam`. The dotted horizontal line indicates the analytical value of the pressure jump, $\triangle P_{anal} = 2$ Pa. As the mesh is refined, the jump in pressure, $\triangle P_{vof}$ converges to a value that is not the exact value. This can be further verified by perusing Table 4.2. For example, for *mesh1*, *mesh2* and *mesh3*, $\triangle P_{vof}$ is 1.779 Pa, 1.754 Pa and 1.749 Pa respectively. Results shown in Figure 4.3 are obtained using the new solver and adopting the fourth order least squares method for the evaluation of $\nabla\psi$. We observe a much better result in this case, since the jump in pressure seems to converge to a closer value to the analytical result.

A quantitative comparison of both methods on the most refined mesh is shown in Figure 4.4. Here, we clearly see that LS-VOF outweighs the VOF method in its prediction of the pressure jump.

We recall that the end goal was to assess the implementation of this new method in the calculation process of multiphase viscoelastic fluid flows, which is expected to be more effective in surface tension dominated flows. The relative impact of this newly added feature was tested by mimicking the above experiment for the case of an Oldroyd-B continuous phase (fluid VE6 in Table 4.1). The physical properties for the bubble remains the same while for the continuous phase we use the following parameter values: $\eta_s = 0.0009$Pa·s, $\eta_p = 0.0001$Pa·s, $\rho_c = 1000\text{kg}/m^3$ and $\lambda = 0.01s$.

The results obtained were close to the analytical value as indicated in Figure 4.5 and Table 4.2.

4.2 Drop deformation in shear flow

Investigating steady-state and transient shear flow dynamics of a drop is paramount for the development of emulsification technology. When a single drop undergoes sufficiently high shear stress, it stretches for some period of time before finally breaking up into small droplets. The sequence of events that result in the generation of daughter drops serves as an archetype that aids the fundamental understanding of the underlying mechanism of emulsification and mixing.

The new solver `clsVeInterDymFoam`, using fourth order least squares method to compute $\nabla\psi$, is further tested for efficiency and accuracy on a dynamic test case - drop deformation in shear flow. The fluid parameter values chosen for both the drop and continuous phase are similar to the one used in the experimental work of Li et al. [97]. For the Newtonian continuous phase (fluid N3 in Table 4.1), the density is $\rho_d = 984$ kg/m^3 and viscosity is $\eta_d = 69.5$ mPa·s; for the Giesekus drop phase (fluid VE1 in Table 4.1), the density is $\rho_c = 1000$ kg/m^3, the relaxation time, $\lambda = 0.25$ s, the solvent viscosity, $\eta_s = 6.05$ mPa· s and the polymeric viscosity, $\eta_p = 6.05$ mPa·s giving a zero-shear-rate viscosity of $\eta_o = 12.1m$Pa·s. Hence for this problem, the density ratio

(ρ_d/ρ_c) is 1.016 and viscosity ratio (η_d/η_c) at zero-shear-rate is 0.174. The interfacial tension between the two fluids is $\sigma = 0.102$ mN/m.

4.2.1 Computational parameters

The geometry used in this study is shown in Fig. 4.6. In Fig. 4.6, $H = 5$mm and $W = 10$mm. The initial diameter of the drop positioned midway between two parallel plates is $d = 1$mm. The matrix liquid is then subjected to simple shear with intensity depending on the velocity of the plates. For all simulations, we set the velocity at the upper wall as $u = 5$mm/s, at the lower wall, we set $u = -5$mm/s and at the inlet and outlet boundary, the gradient of the velocity was set to zero. Zero Neumann boundary condition was specified for the volume fraction, α, pressure, p and polymeric stress, $\boldsymbol{\tau}_p$ at all boundaries.

The non-dimensional parameters considered are the Reynolds number, $Re = \rho_c u H/\eta_c$, capillary number, $Ca = \eta_c \dot{\gamma} a/\sigma$ and Deborah number, $De = \lambda\dot{\gamma}$; where ρ_c and η_c is the density and viscosity of the matrix phase respectively, $\dot{\gamma} = 2u/H$ is the shear rate, u is the velocity of the moving wall (see Fig. 4.6) and a is the radius of the drop. The continuous phase is Newtonian and the dispersed phase is the Giesekus fluid.

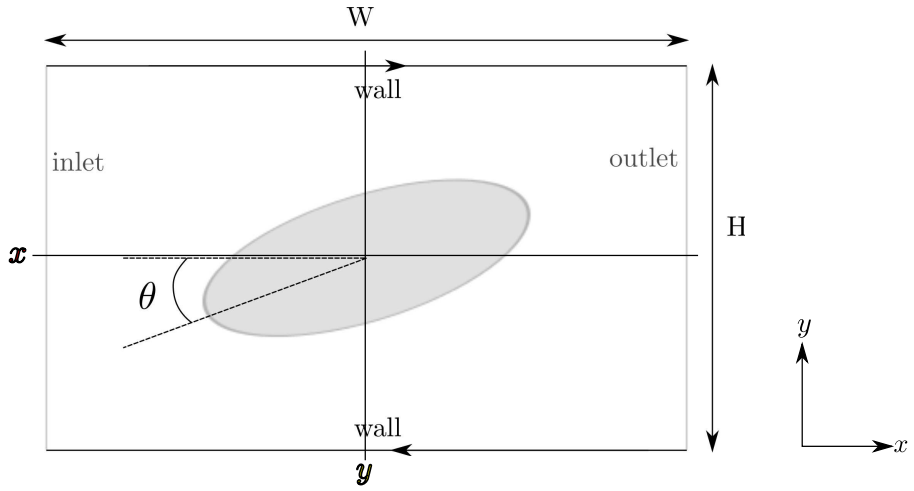


Figure 4.6: Computational domain for drop deformation in shear flow.

4.2.2 Parameter study for adaptive mesh refinement

To enable the use of automatic refinement with a two-dimensional simulation case in OpenFoam, the library `dynamicRefineFvMesh2D` of Baniabedalruhman [27] (see previous chapter) is called during run-time. This is made possible by setting the keyword, `dynamicFvMesh`, in the dynamicMeshDict[1] to `dynamicRefineFvMesh2D`. Figure 4.7 shows the parameter settings used for one of the dynamic mesh cases. Line 18 specifies the library to call during run-time. The variable values to use for the functions defined in `dynamicRefineFvMesh2D` are determined in line 22 - line 48. In line 22, the number of timestep(s) after which refinement can occur is given. Line 24 specifies the field to base the refinement upon. In lines 26, 27 and 32, we assign values to the variables `lowerRefineLevel, upperRefineLevel` and `nBufferLayers`.

[1]dynamicMeshDict is a dictionary file that controls deformation and morphing of the mesh during a simulation. It is only useful on solvers that invoke mesh motion

Any cell with values between `lowerRefineLevel` and `upperRefineLevel` is refined. Cells that are within `nBufferLayers` of points marked for refinement are unrefined. `nBufferLayersR` determines the number of buffer layers to be extended for refinement [27]. The maximum refinement level which starts at 1 is given in line 34. Line 36 specifies the maximum cell limit above which the refinement process should be terminated. Lastly, line 48 determines whether to write the level of refinement for each cell as a field or not.

Our goal is to investigate the best parameter settings to be used with the dynamicMeshDict for optimal efficiency and accuracy. To begin, a grid convergence study is carried out on three static meshes - mesh 1, mesh 2 and mesh 3. Next, we conduct similar simulations on several cases with dynamic mesh functionality enabled. The dynamic mesh cases can be grouped into two - one using mesh 1 as the base mesh (amrIJ) and the other sets used mesh 2 as the base mesh (amrIJB). I and J in amrIJ and amrIJB denote respectively the level of mesh refinement (maxRefinement in Table 4.3) and specific range of values of the volume fraction (refineRange in Table 4.3). We considered the cases when I=$1, 2, 3$ and 4, and the following range of values for J: $1 \sim [0.1, 0.9]$, $2 \sim [0.01, 0.99]$, $3 \sim [0.001, 0.999]$, $4 \sim [0.1, 1]$ and $5 \sim [0.01, 1]$. A summary of the properties of each case can be found in Table 4.3. All results were taken at time, $t = 20s$ when steady state was attained by all cases. We remark that the number of cells shown in Table 4.3 is the number of cells in the computational domain at the end of simulation.

```
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      dynamicMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dynamicFvMesh   dynamicRefineFvMesh2D;

dynamicRefineFvMesh2DCoeffs
{
    refineInterval  1;

    field           alpha.phase1;

    lowerRefineLevel 0.01;
    upperRefineLevel 0.99;

    unrefineLevel   10;

    nBufferLayersR   1;
    nBufferLayers   2;

    maxRefinement   2;

    maxCells        200000;

    correctFluxes
    (
     (phiAlpha none)
     (phi none)
     (nHatf none)
     (rhoPhi none)
     (ghf none)
    );

    dumpLevel       true;
}


// ************************************************************************* //
```

Figure 4.7: Summary of `dynamicMeshDict` (without header) parameters for drop deformation in shear flow (*amr22 case*).

Table 4.3
Summary of mesh properties for all test cases considered for drop deformation in shear flow. $\triangle\hat{x} = \triangle x_{min}/d$, where d is the diameter of the drop. amrIJ uses mesh1 as the base mesh and amrIJB uses mesh2 as the base mesh; CPU times shown are for a single processor.

Cases	maxRefinement	refineRange	$\triangle\hat{x}$	Number of cells	CPUTime (s)
mesh1	-	-	0.05	20000	3130.91
mesh2	-	-	0.025	80000	30333.6
mesh3	-	-	0.0125	320000	73045
amr11	1	0.1 – 0.9	0.025	20452	5133.7
amr12		0.01 – 0.99		20676	5821.4
amr13		0.001 – 0.999		22089	7516.6
amr14		0.1 – 1		21189	5030.2
amr15		0.01 – 1		21298	5571.4
amr21	2	0.1 – 0.9	0.0125	21560	7707.8
amr22		0.01 – 0.99		22220	8877.4
amr23		0.001 – 0.999		32790	12079.4
amr24		0.01 – 1		25827	7806.7
amr31	3	0.1 – 0.9	0.00625	24114	10334.7
amr32		0.01 – 0.99		27645	13141.5
amr33		0.001 – 0.999		81254	32103.4
amr41	4	0.1 – 0.9	0.003125	34180	22692.6
amr42		0.01 – 0.99		80717	49110.8
amr43		0.001 – 0.999		221025	119658
amr11B	1	0.1 – 0.9	0.0125	80934	29561.9
amr12B		0.01 – 0.99		81551	32672.6
amr13B		0.001 – 0.999		96456	39803.4
amr14B		0.1 – 1		84230	29658
amr15B		0.01 – 1		84579	31183.2
amr22B	2	0.01 – 0.99	0.00625	94280	40188.8

In Table 4.3, as expected on the uniform meshes, the CPU time increased as the mesh size increased. The same is true when adaptive meshing is used, except when the upperRefineLevel is 1. In other words, the CPU time increases as the number of cells at the end of calculation increases. This, in turn, occurs when the refineRange increases. The most likely reason for this oddity is that out of the total simulation time, more time was spent on updating the mesh within each time step for the cases amr12, amr22 and amr12B (see Fig. 4.8 and 4.9). For the cases with an upperRefineLevel as 1, an increase in the number of cells and CPU time is observed as the refineRange increases. We also note that setting the upperRefineLevel as 1 forces all the cells within the drop to be refined the maximum number of times.

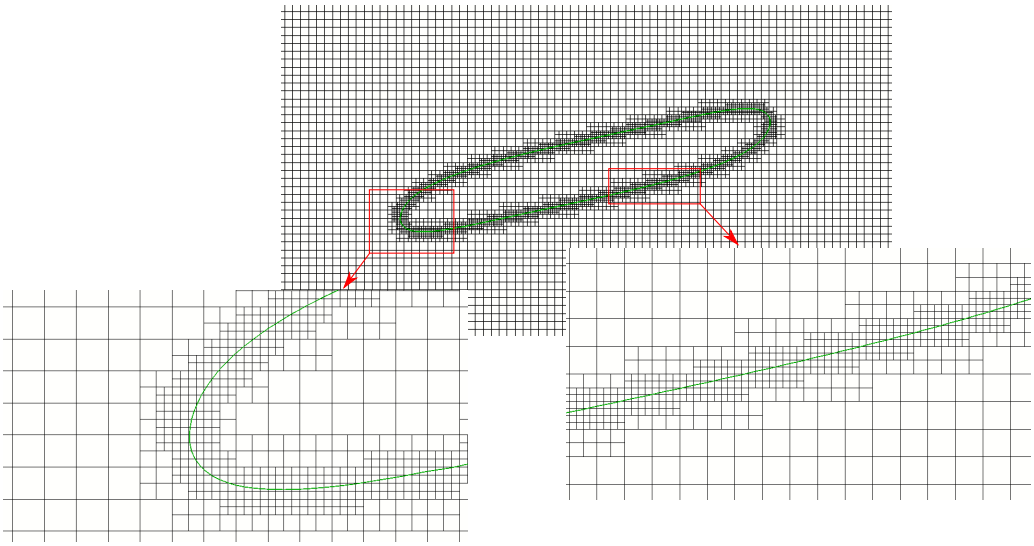


Figure 4.8: Dynamic mesh showing refined cells around the interface for drop deformation in shear flow at time, $t = 20s$ *(amr22 case)*.

In Fig. 4.10, the contour lines of the volume fraction field at $\phi = 0.5$ for the cases amr31, amr32 and amr33 are compared with mesh1, mesh2 and mesh3. We observe

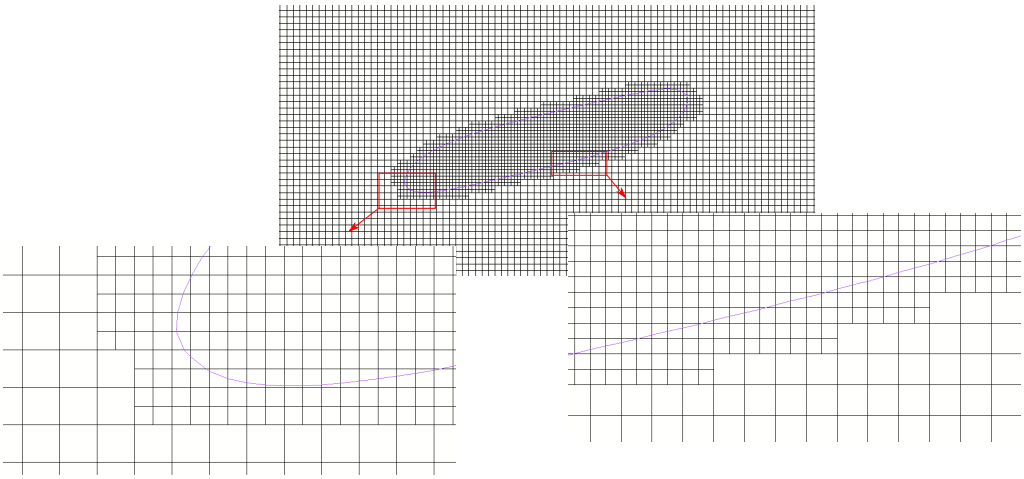


Figure 4.9: Dynamic mesh showing refined cells around the interface for drop deformation in shear flow at time, $t = 20s$ *(amr15 case).*

that all dynamic mesh (amr) cases shown predicted a greater drop elongation length with amr32 predicting the shortest length. Similar behavior was seen in amr41, amr42 and amr43. Hence, no further investigation was done for these cases.

Next, we compare the contour lines of the cases amr21,amr22 ,amr23 and amr24 with mesh1, mesh2 and mesh3 in Fig. 4.11. Again, we observe a longer deformed steady drop as in the amr cases with three and four levels of mesh refinement. However, the orientation of the drops are better aligned.

The best result for the amr cases using mesh 1 as the base grid were those ran with one level of mesh refinement (amr1*). They are faster and more accurate. As shown in Fig. 4.12, amr12, amr13, amr14 and amr15 are mesh independent with respect to the contour lines. In addition, the contour lines for these cases were closer to the contours of mesh 3 than mesh 2. Considering the pressure profile along the x-axis,

on the other hand, amr15 gave the best result relative to all amr cases with mesh 1 as the base grid. This is delineated in Fig. 4.15. We remark here that the run time for these cases are about one-fifth that of mesh 2.

For the cases with mesh 2 as the base mesh, the contour lines for amr11B, amr12B, amr13B and amr15B, and amr22B are compared mesh1, mesh2 and mesh3 in Fig. 4.13 and Fig. 4.14 respectively. The pressure profile along the x-axis was also compared with mesh1, mesh2 and mesh3. We include only the best two cases in the results shown in Fig. 4.16. We observe in Fig. 4.16 that amr15B is not mesh independent. Although, the case amr22B agrees more with mesh 3, the run time is a lot higher - about 3 hrs more (see Table 4.3).

We thus recommend using mesh 1 as the base mesh and setting the upperRefineLevel to 1 because it is more efficient in terms of speed and accuracy.

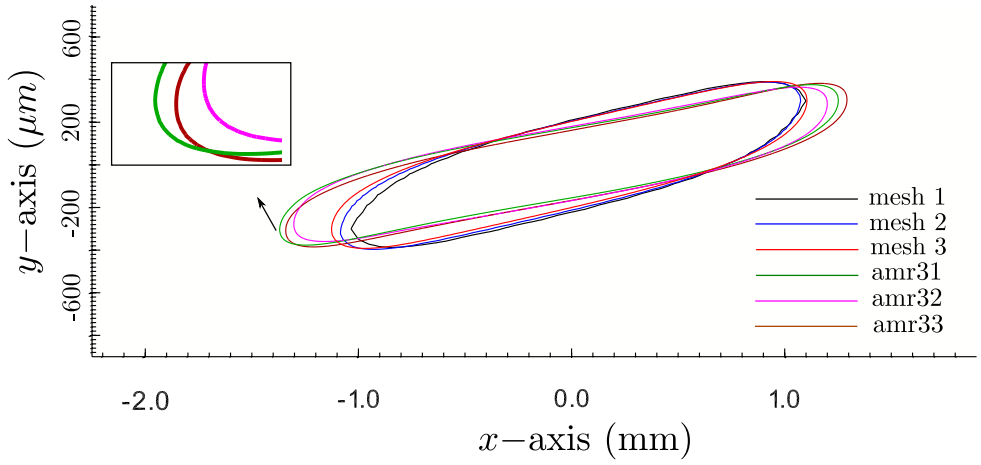


Figure 4.10: Comparison of contour lines at $\alpha = 0.5$ of a steady deformed shape of Giesekus drop in a Newtonian fluid for mesh1, mesh2, mesh3, amr31, amr32 and amr33. (Base mesh is mesh1, maxRefinement = 3).

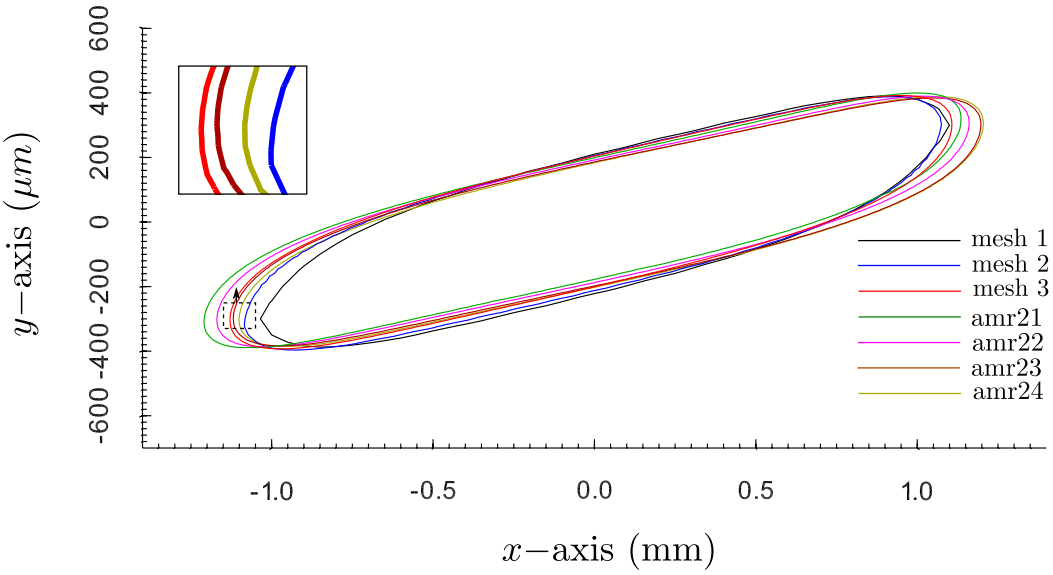


Figure 4.11: Comparison of contour lines at $\alpha = 0.5$ of a steady deformed shape of Giesekus drop in a Newtonian fluid for mesh1, mesh2, mesh3, amr21, amr22, amr23 and amr24. (Base mesh is mesh1, maxRefinement = 2)

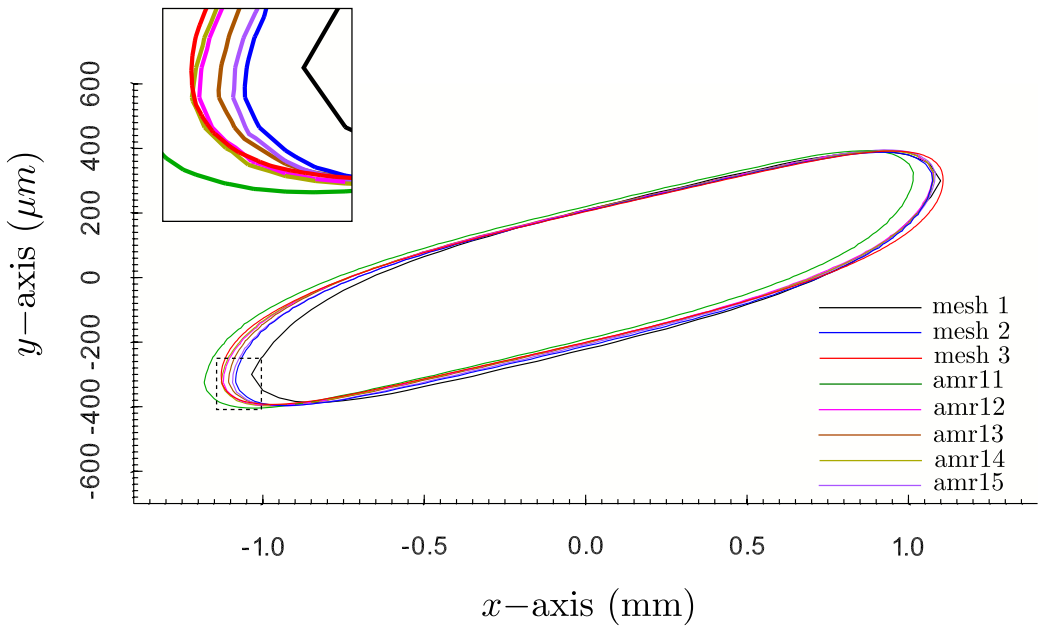


Figure 4.12: Comparison of contour lines at $\alpha = 0.5$ of a steady deformed shape of Giesekus drop in a Newtonian fluid for mesh1, mesh2, mesh3, amr11, amr12, amr13, amr14 and amr15. (Base mesh is mesh1, maxRefinement = 1)

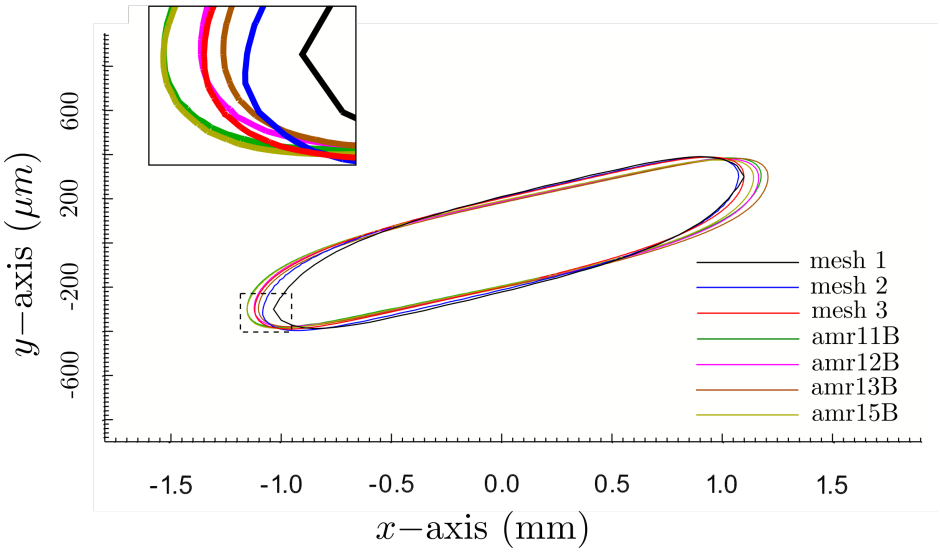
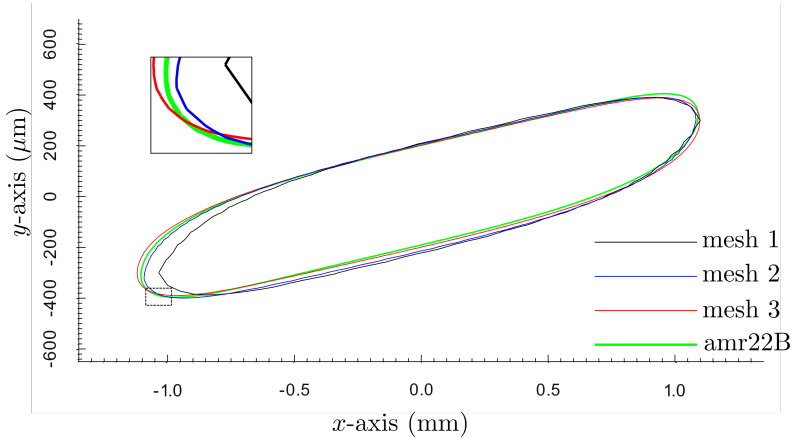


Figure 4.13: Comparison of contour lines at $\alpha = 0.5$ of a steady deformed shape of Giesekus drop in a Newtonian fluid for mesh1, mesh2, mesh3, amr11B, amr12B, amr13B and amr15B. (Base mesh is mesh2, maxRefinement = 1)



(a)

Figure 4.14: Comparison of contour lines at $\alpha = 0.5$ of a steady deformed shape of Giesekus drop in a Newtonian fluid for mesh1, mesh2, mesh3 and amr22B. (Base mesh is mesh2, maxRefinement = 2)

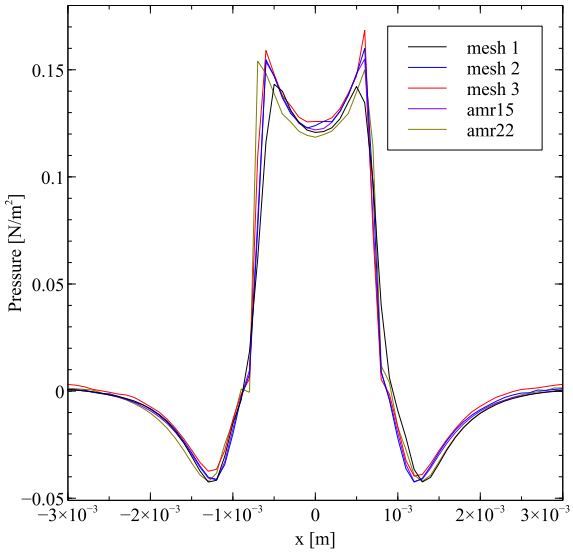


Figure 4.15: Pressure profile along the horizontal line $y = 0$ at time $t = 20s$ for mesh1, mesh2, mesh3, amr15 and amr22.

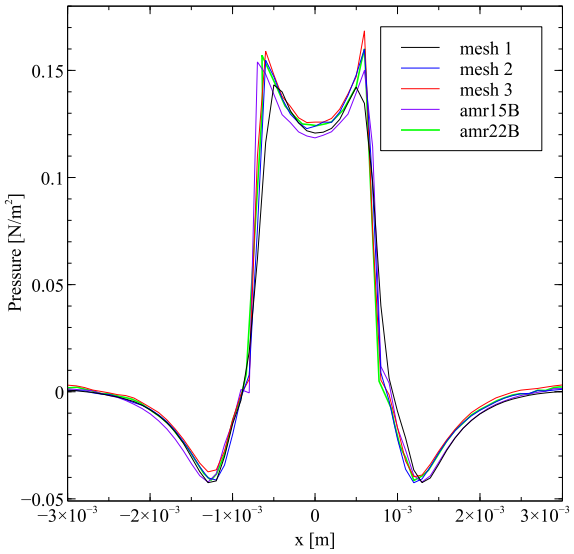


Figure 4.16: Pressure profile along the horizontal line $y = 0$ at time $t = 20s$ for mesh1, mesh2, mesh3, amr15B and amr22B.

4.2.3 Effect of viscoelastic model on drop phase

Here, we investigate the differences in the prediction of two viscoelastic models used in characterizing the drop phase for the same Newtonian continuous phase using static mesh 2 with no AMR. The Newtonian fluid, N3, and the viscoelastic fluids, VE1 and VE2 were selected for this study (see Table 4.1). We note that VE1 and VE2 represents a Giesekus and Oldroyd-B model respectively.

At steady state, no remarkable difference was seen between the volume fraction field for both cases (see Fig. 4.17(a)). We also compared the steady state shape of a Giesekus drop (VE1) in a Newtonian matrix (N3) and a Newtonian drop (N3) in a Giesekus matrix (VE2). As delineated in Fig. 4.17(b), we observe that the droplet is more deformed when it is viscoelastic and the continuous phase is Newtonian. To understand why this occurs, we recall that the capillary number, Ca, gives an indication of the relative strength of the viscous stretching force to the resistive force due to interfacial tension. Thus, the higher deformation of the viscoelastic droplet is expected since in this case, the drop is subjected to a higher shear stress ($Ca = 0.34$) in comparison to the Newtonian drop ($Ca = 0.059$) for the same resistive force.

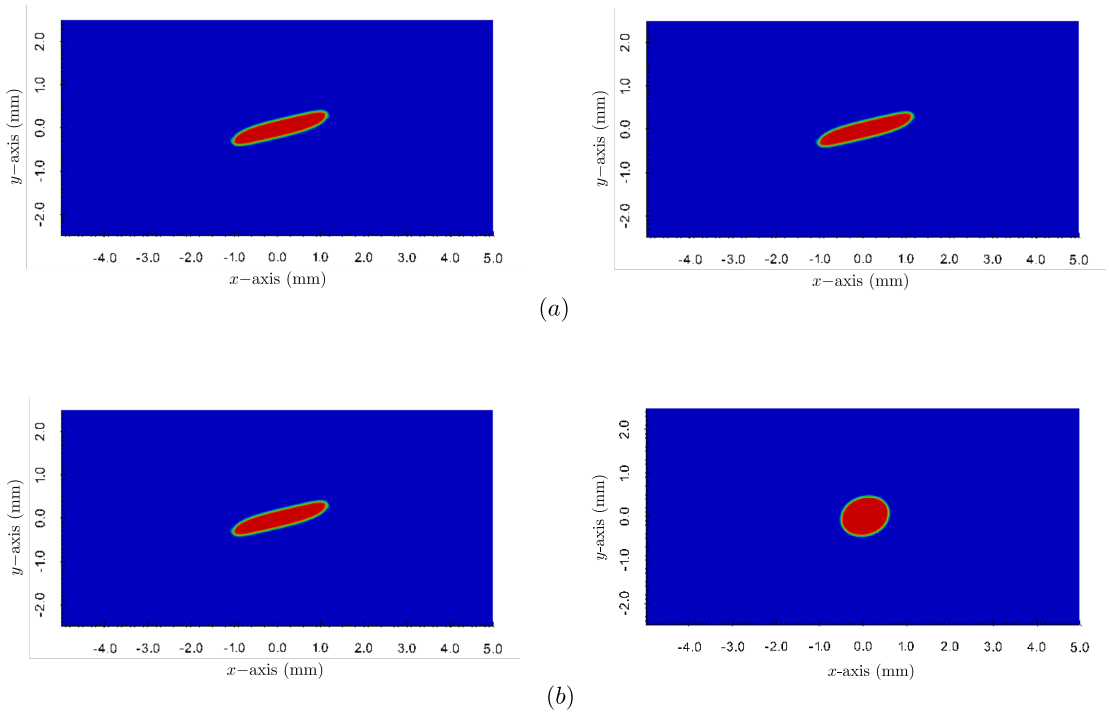


Figure 4.17: Comparison of steady deformed droplet shape (a) left: Giesekus drop in a Newtonian matrix; right: Oldroyd-B drop in a Newtonian matrix (b) left: Giesekus drop in a Newtonian matrix ($Ca = 0.34, \dot{\gamma} = 2s^{-1}$, viscosity ratio=0.174); right: Newtonian drop in a Giesekus matrix ($Ca = 0.059, \dot{\gamma} = 2s^{-1}$, viscosity ratio=5.74).

Finally, a comparison of the profile of the components of polymeric stress, $\tau_{p,xx}, \tau_{p,xy}, \tau_{p,yy}$, pressure, p, and $x-$component of velocity, u were taken along the horizontal line $y = 0$. The results are shown in Fig. 4.18. In Fig. 4.18, *ls-Oldroyd-B* and *ls-Giesekus* represent results obtained with `clsVeInterDymFoam` while other cases (*Oldroyd-B* and *Giesekus*) were obtained using `viscoelasticInterFoam`. We observe that the only remarkable difference is in the stress distribution. The Oldroyd-B model predicts a larger stress value within the drop for all components. In contrast to the Oldroyd-B fluid, the viscosity of the Giesekus fluid decreases as the shear rate increases which results in a reduction in the magnitude of stress within the drop.

Furthermore, the results predicted with `clsVeInterDymFoam` are lesser for all profiles. The stress field for all the four cases are shown in Fig. 4.19. Figure 4.19 (a) and (c) were obtained using `viscoelasticInterFoam` while Fig. 4.19 (b) and (d) were obtained with `clsVeInterDymFoam`. As shown, the Oldroyd-B drop has a higher concentration of stress than the Giesekus drop.

4.2.4 Effect of elasticity on drop phase

The role elasticity of drop plays in its deformation when subjected to simple shear flow is examined here. Three cases - N3–N3, VE2–N3, VE3–N3 were studied using mesh 2. All cases had the same Newtonian fluid, N3, as the continuous phase. For the cases N3–N3, VE2–N3 and VE3–N3, the drop phase is the fluid N3, VE2 and VE3 respectively. Also for the three cases, N3–N3, VE2–N3 and VE3–N3, $Ca = 0.34$; $De = 0, 0.5, 200$ and viscosity ratio $= 1, 0.174, 0.174$ respectively. The material properties of these fluids can be found in Table 4.1.

Figure 4.20 illustrates the steady deformed shapes of the drop for the three cases. It can be seen that the drop was most deformed in the Newtonian case and had the least amount of deformation in the case with highest elasticity. In addition, we also observed in Fig. 4.6 that as elasticity of drop increases, drop deformation decreases and drop alignment decreases.

In an attempt to explain this behavior, the strain rate profile was investigated. The distribution of the strain rate across the whole domain for all three cases is shown in Fig. 4.21. For all cases, the strain rate is highest at the tip of the deforming droplet. However, the N3–N3 case has the most even distribution of strain rate. Consequently, the strain rate transmitted from the walls is evenly distributed across the drop and this gives rise to the high drop alignment and elongation. In the VE3-N3 case on the other hand, most of the strain rate is absorbed at the tip.

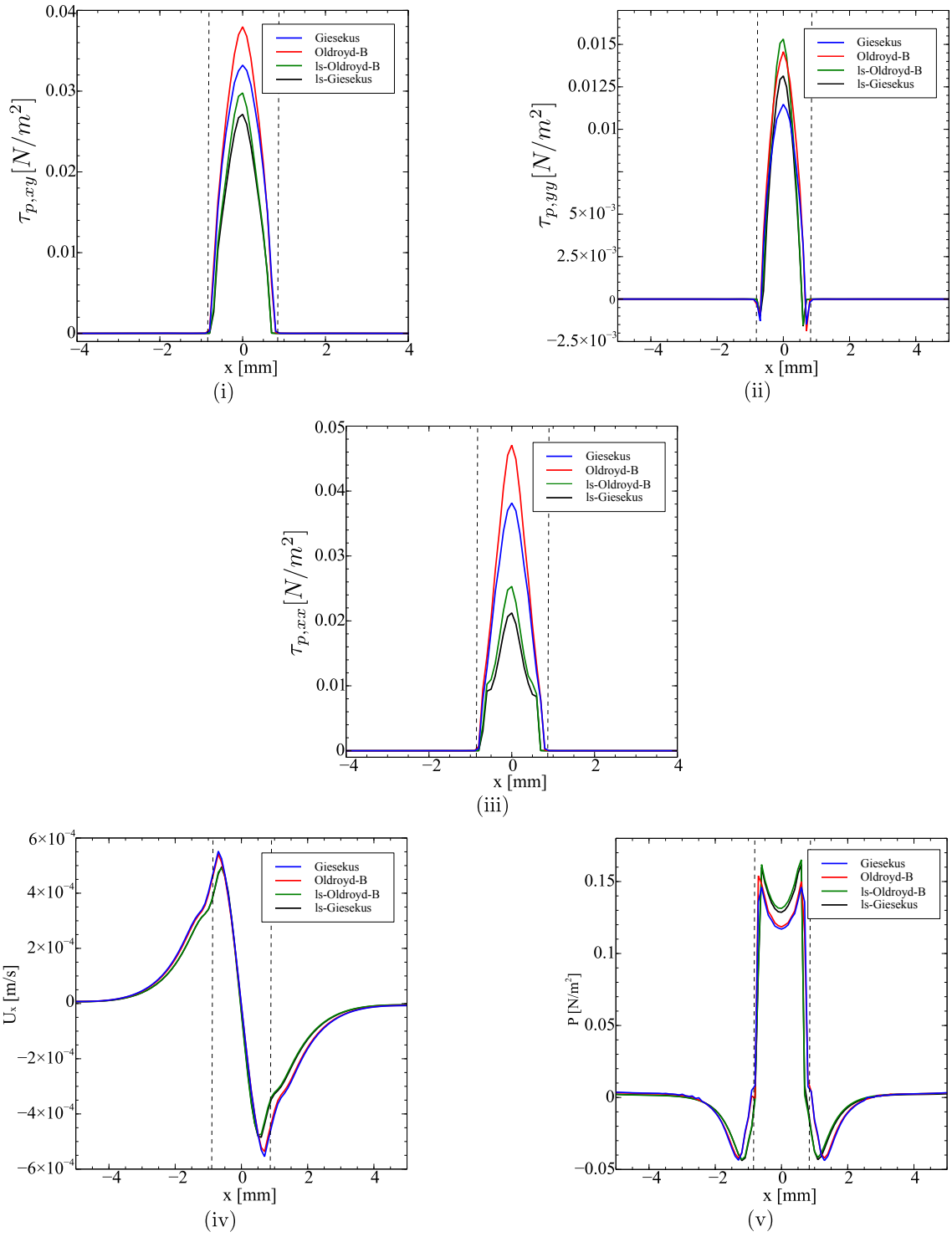


Figure 4.18: Comparison of profiles along the x-axis: viscoelastic drop in a Newtonian matrix.

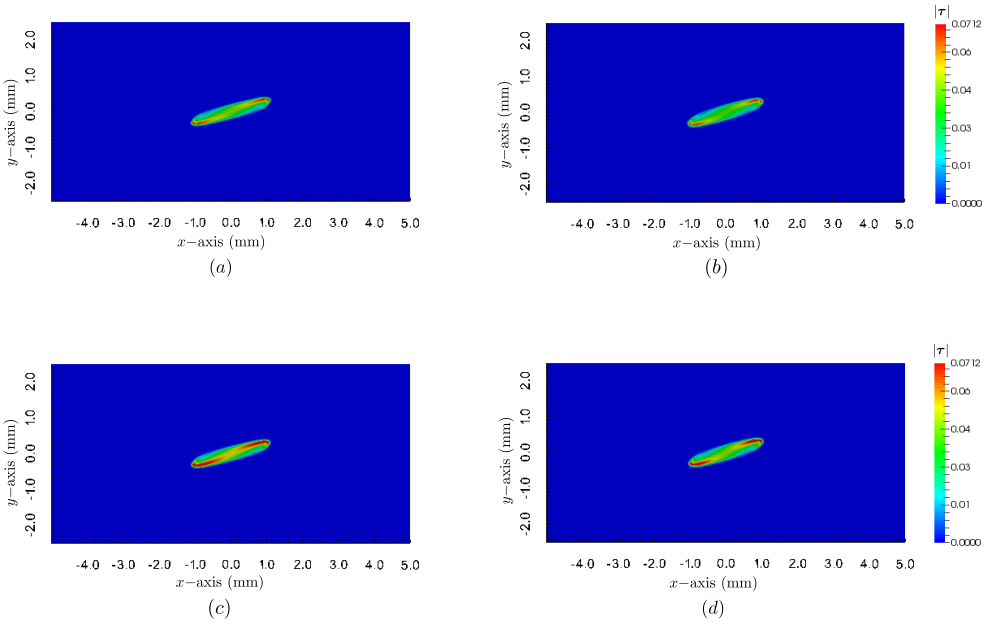


Figure 4.19: Polymer stress magnitude: viscoelastic drop in a Newtonian matrix using static mesh 2 (a) Giesekus drop (`viscoelasticInterFoam`) (b) Giesekus drop (`clsVeInterDymFoam`) (c) Oldroyd-B (`viscoelasticInterFoam`) (d) Oldroyd-B (`clsVeInterDymFoam`).

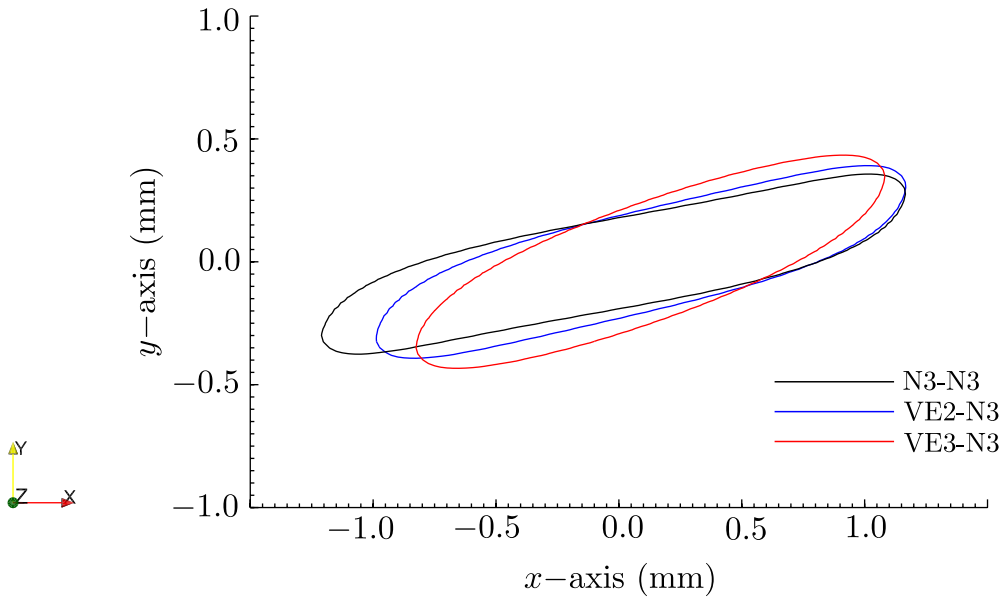


Figure 4.20: Effect of elasticity for drop deformation in shear flow.

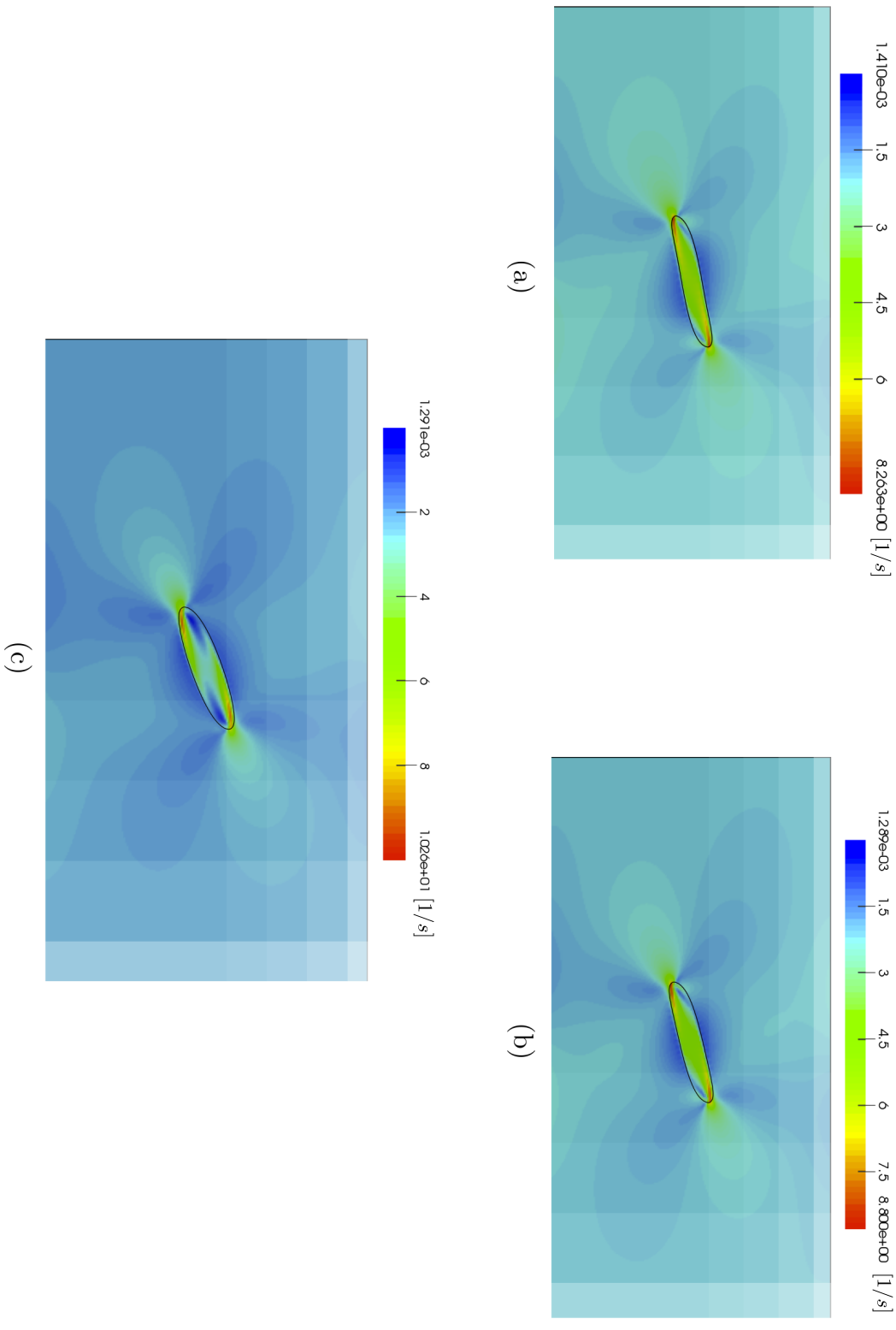


Figure 4.21: Strain rate distribution for drop deformation in shear flow (a) N3-N3; $De = 0$, $Ca = 0.34$ (b) VE2-N3; $De = 0.5$, $Ca = 0.34$ (c) VE3-N3; $De = 200$, $Ca = 0.34$.

4.3 Rising of a Newtonian drop in a liquid

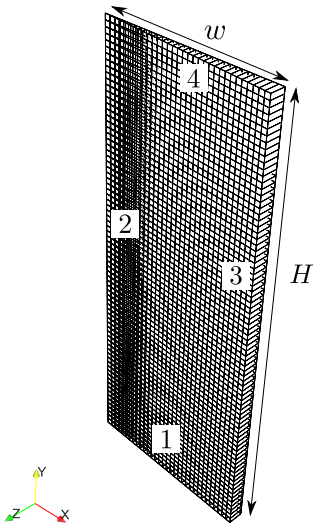
To test the capability of `clsVeInterDymFoam` in handling an axisymmetric geometry, the rise of a Newtonian drop in a viscoelastic matrix (Giesekus fluid) is considered. The geometry is a cylinder of radius $r = 0.04m$ and height $H = 0.1m$. The bottom of the cylinder is a wall and the top is open. The computational domain used has the shape of a wedge (see Figure 4.22). The density of the drop (fluid N3 in Table 4.1) and continuous phase (fluid VE1 in Table 4.1) is 984kg/m^3 and 1000kg/m^3 respectively while the dynamic viscosity of the drop and continuous phase is 69.5mPa·s and 12.1mPa·s; The relaxation time is $\lambda = 0.25s$ and the interfacial tension between both phases is given by σ =0.102mN/m. A summary of the fluid properties used is given in Table 4.4.

Table 4.4
Fluid properties of drop and continuous phase for the rising drop case.

Property	Drop phase (Newtonian)	Continuous phase (Giesekus fluid)
Density (kg/m^3)	984	1000
Viscosity (mPa·s.)	69.5	$\eta_s = 6.05$, $\eta_p = 6.05$
Relaxation time (s)	-	0.25
Mobility factor	-	0.003

A drop with radius, 1 cm, was placed initially at rest close to the bottom of the geometry and left to rise freely as a result of buoyancy. Acceleration due to gravity is $g = -9.8\text{m}/s^2$ in the $y-$direction. At time, $t = 0$, the velocity, $\boldsymbol{v}$, pressure, p, and

stress, $\boldsymbol{\tau}_p$ field was set to zero. The volume fraction, ϕ was initialized to 1 in the drop and 0 outside the drop (continuous phase). A summary of the boundary conditions prescribed for all variables are delineated in Fig. 4.22.



(a)

▷ Boundary conditions

	$\boldsymbol{v}$	ϕ	p	$\boldsymbol{\tau}_p$
1	no-slip	zero gradient	zero gradient	zero gradient
2	symmetry	symmetry	symmetry	symmetry
3	no-slip	zero gradient	zero gradient	zero gradient
4	pressureInletOutletVelocity	inletOutlet	totalPressure	zero gradient

Figure 4.22: (a) Geometry for the rising drop case. $H = 0.1$m and $w = 0.04$m.

Table 4.5 shows the meshes and the total run time for each mesh used for convergence study. The nomenclature is similar to that in the previous section. As expected for the static meshes, the run time increases as the number of cells increases but a lesser

number of cells is used on the dynamic mesh for an almost same accuracy with the most refined mesh. A summary of the parameters employed for the AMR case is shown in Fig. 4.23. In Fig. 4.23, `dynamicRefineFvMeshAxi` in line 18 is a library called during runtime to enable AMR on an axisymmetric grid. `dynamicRefineFvMeshAxi` was developed by Baniabedalruhman [27]. Selective snapshots of the volume fraction field at level set, $\phi = 0.5$, are shown in Figure 4.24. As shown, AMR matches closely with mesh 3 and was used for subsequent simulations.

Table 4.5
Summary of mesh information. $\triangle\hat{x} = 2\triangle x_{min}/d$, where d is the diameter of the drop.

Meshes	$\triangle\hat{x}$	Number of cells	CPUTime (s)
1	0.25	600	22.25
2	0.125	2480	69
3	0.0625	10080	332
AMR	0.0625	3626	175.14

In the rest of this section, we investigate the changes in the shear rate field around a drop (fluid N3) rising in a viscoelastic fluid (fluid VE1) that shear-thins (Giesekus fluid). It is very important to determine the behavior of drops in various flow conditions that include a Newtonian and non-Newtonian matrix phase. This would help elucidate necessary details for bringing well-suited process operation and model to reality.

Figure 4.25 shows the shear rate distribution (shear rate $= \sqrt{2D : D}$, $D = (\nabla \boldsymbol{v} +$

```
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      dynamicMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dynamicFvMesh   dynamicRefineFvMeshAxi;

dynamicRefineFvMeshAxiCoeffs
{
    refineInterval  1;

    field           alpha.phase1;
    axis 2;
    axisVal 0;

    lowerRefineLevel 0.001;
    upperRefineLevel 0.999;

    unrefineLevel   10;
    nBufferLayersR   1;
    nBufferLayers   2;

    maxRefinement   2;
    maxCells        200000;

    correctFluxes
    (
     (phiAlpha none)
     (phi none)
     (nHatf none)
     (rhoPhi none)
     (ghf none)
    );
    dumpLevel       true;
}
```

Figure 4.23: Summary of `dynamicMeshDict` parameters for rising drop case.

$(\nabla \boldsymbol{v})^T)/2)$ and shape of a drop (fluid N3) rising in a Newtonian (fluid N4) and Giesekus fluid (fluid VE1). A wider region of greater local shear rate is observed at the drop nose in the Giesekus fluid than the Newtonian. To help elucidate how the viscosity of the Giesekus fluid is locally distributed, we consider its velocity field. Figure 4.25 (bottom) shows the flow field around the drop after 1.2s from rest position.

It could be seen that as the drop rises, the liquid close to its head is propelled upward, some of which finds its way back into the wake of the bubble. This leads to the formation of a vortex on the periphery of the drop. Although, the velocity distribution in both cases seems almost indistinguishable, an increased vortex is seen to appear as the influence of the shear-thinning becomes more pronounced. Due to the increased shear-rate zone around the front line of the drop, a remarkable velocity gradient is seen in that area. For the shear-thinning Giesekus fluid, this causes a reduction in its viscosity in that region which consequently results in an increase in the rising velocity of the drop.

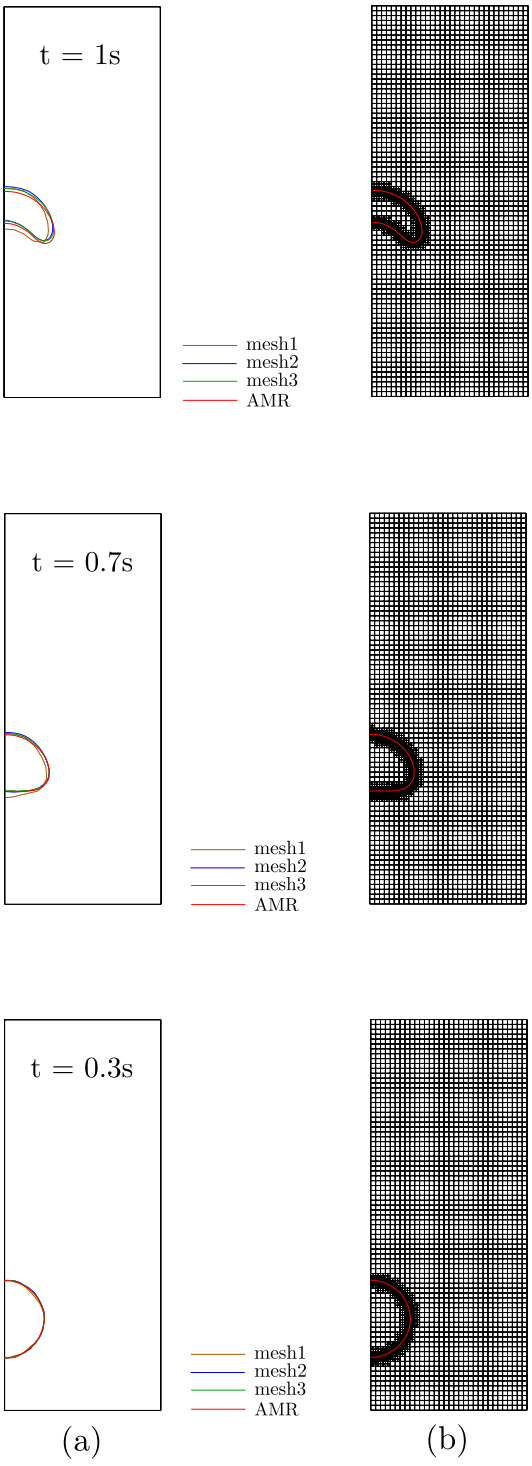


Figure 4.24: Rise of a Newtonian drop in a Giesekus fluid on three meshes (a) $\alpha = 0.5$ contour plot (b) Contour plot on a dynamically refined mesh (AMR).

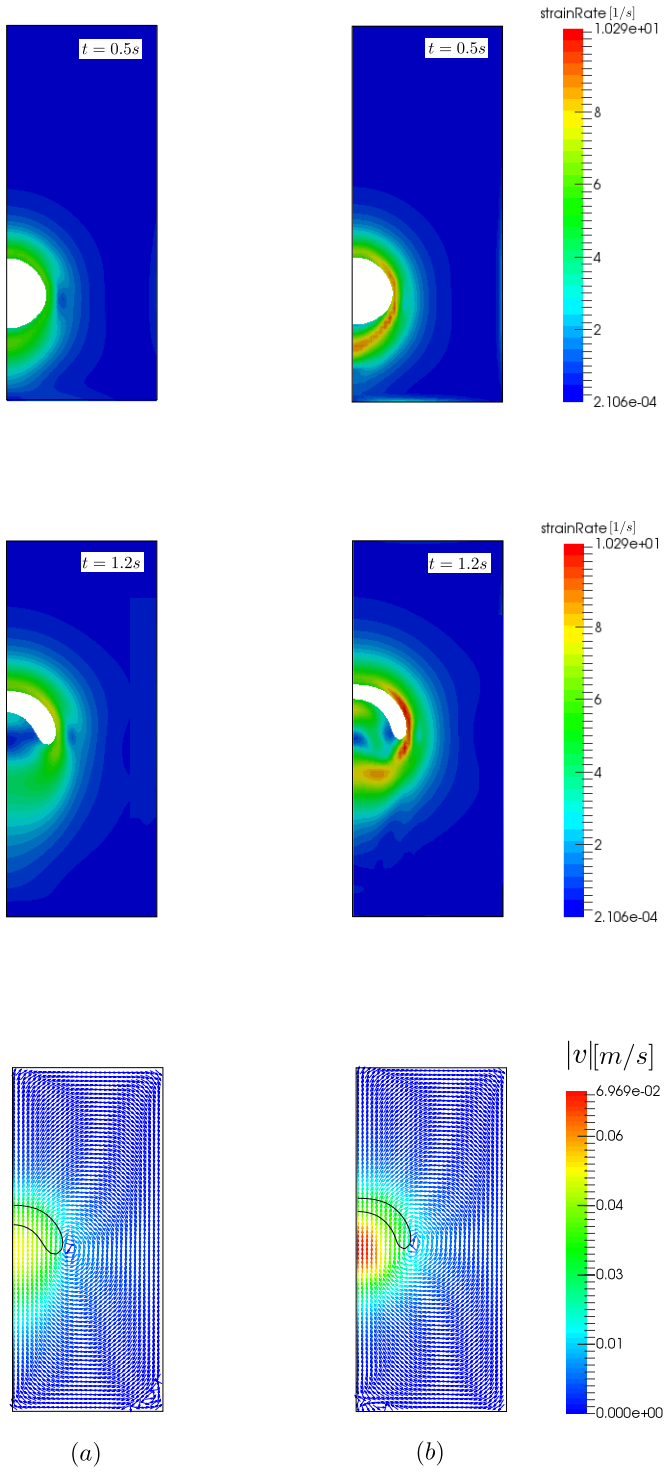


Figure 4.25: Top and Middle: Shear rate distribution and shape of a drop (fluid N3) rising in a Newtonian and viscoelastic fluid; Bottom: Velocity field at time, $t = 1.2s$. (a) Newtonian (fluid N4) (b) Viscoelastic matrix phase (fluid VE1).

4.4 Experimental validation

In this section, `clsVeInterDymFoam` is further validated by comparing the results of a two-dimensional computer simulation with experiments performed by Li et al. [1] of the formation process of a viscoelastic droplet in a three-dimensional microchannel.

For the experiment, an aqueous solution of cetyltrimethylammonium chloride (CTAC) and Sodium salicylate (NaSal), which was used as the disperse phase, is introduced into a flowing stream of silicone oil (continuous phase). The dispersed phase is eventually pinched at some point near the mouth of its entrance channel and plugs are formed. These plugs are then transported downstream by the continuous phase.

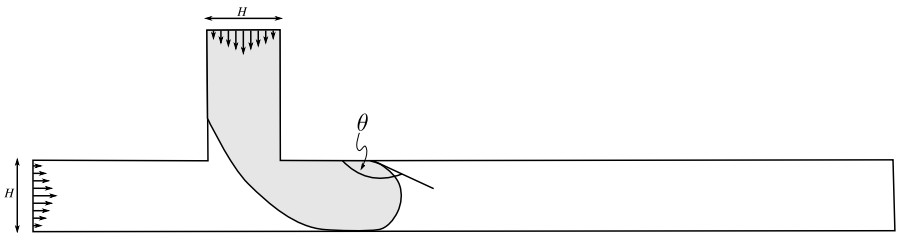


Figure 4.26: Schematic illustration of the viscoelastic two-phase system. $H = 100\,\mu$m.

The flow problem is shown schematically in Figure 4.26. The width of both inlet channels is $H = 100\,\mu$m and θ represents the contact angle. The base mesh of this

case has a size of $\triangle x/H = 0.0962$. Using two levels of refinement, the size of the smallest cell i.e. around the interface is given by $\triangle x/H = 0.02405$. A summary of the parameter settings for dynamic mesh refinement used with all cases is outlined in Fig. 4.27. At the walls, the no-slip condition was set. A zero normal gradient was specified at both inlets for p and $\boldsymbol{\tau}_p$, at the walls for p and $\boldsymbol{\tau}_p$, and at the outlet for $\boldsymbol{v}, \boldsymbol{\tau}_p$ and ϕ. A value of zero was assumed at the outlet for p. The equilibrium contact angle used is 180^o. At both inlets, a fully developed velocity profile is prescribed so as to conduct the numerical simulations with a channel length smaller than that used in the experiment. The procedure for setting up a fully developed profile at both inlets is the same, so an explanation for one of the inlet is given. We consider a channel with the same width as any of the inlet e.g 100μm, and length that is long enough (20 channel diameters was used in the current study). A single phase calculation is then carried out using the same parameters as the fluid entering the corresponding inlet and solved until the velocity profile along a cross-section of channel doesn't change beyond the entrance region. This indicates the flow is fully-developed. The next step is to map this fully-developed profile to the inlet as a boundary condition. To do this, we employ the `mapFields` utility in the OpenFOAM library.

A summary of the parameter setting for all linear solvers used in this chapter is shown in Fig. 4.28. The volume fraction equation (alpha.phase1) is solved using the multi-dimensionsal limiter for explicit solution (MULES) method by default. Next,

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      dynamicMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
dynamicFvMesh   dynamicRefineFvMesh2D;


dynamicRefineFvMesh2DCoeffs
{

    refineInterval  1;

    field           alpha.phase1;

    lowerRefineLevel 0.001;
    upperRefineLevel 0.999;

    unrefineLevel   10;
    nBufferLayersR   1;
    nBufferLayers   2;

    maxRefinement   2;
    maxCells        200000;

    correctFluxes
    (
     (phiAlpha none)
     (phi none)
     (nHatf none)
     (rhoPhi none)
     (ghf none)
    );

    dumpLevel       true;
}


// ************************************************************************* //
```

Figure 4.27: Summary of dynamic mesh parameters for experimental validation case.

the first pressure equation is solved with a geometric-algebraic multi-grid (GAMG) solver in conjunction with a GAMG preconditioner until either the absolute tolerance is below 10^{-8} or the relative tolerance is below 0.01. The GAMG solver with

GAMG preconditioner was also used for solving the second pressure equation but an absolute tolerance of 10^{-8} is used to determine convergence. We use the GAMG solver here since it is known to be very fast in initially decreasing the residual. Lastly, the GaussSiedel method is used to solve the stress and momentum equations. The absolute tolerance here was set to 10^{-6}.

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{
    "alpha.phase1.*"
    {
        nAlphaCorr      1;
        nAlphaSubCycles 3;
        cAlpha          1;
    }
    p_rgh
    {
        solver          GAMG;
        tolerance       1e-08;
        relTol          0.01;
        smoother        DIC;
        nPreSweeps      0;
        nPostSweeps     2;
        nFinestSweeps   2;
        cacheAgglomeration false;
        nCellsInCoarsestLevel 10;
        agglomerator    faceAreaPair;
        mergeLevels     1;
    }
    p_rghFinal
    {
        $p_rgh;
        relTol          0;
    }
    "pcorr.*"
    {
        $p_rghFinal;
        tolerance       0.0001;
    }
    U
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        tolerance       1e-06;
        relTol          0;
        nSweeps         1;
    }
    tau
      {
        solver          smoothSolver;
        smoother        GaussSeidel;
        tolerance       1e-06;
        relTol          0;
        nSweeps         1;
      }
}
PIMPLE
{
    momentumPredictor no;
    nCorrectors     3;
    nNonOrthogonalCorrectors 0;
    pdRefCell       0;
    pdRefValue      0;
    pRefCell        0;
    pRefValue       0;
}

// ************************************************************************* //
```

Figure 4.28: Summary of fvSolution settings (without header) for drop formation case.

The density of the dispersed and continuous phase is given by $\rho_d = 1000$ kg/m^3 and $\rho_c = 980$ kg/m^3 respectively. An interfacial tension of $\sigma = 22.9$ mN/m was used. Important non-dimensional parameters include the Reynolds number, $Re = \rho_c v_c H/\eta_c$,

the capillary number, $Ca = \eta_c v_c/\sigma$, and the Weissenberg number, $Wi = \lambda\dot{\gamma}$. We note here that the continuous phase corresponds to fluid N3 in Table 4.1 while the drop phase using $\lambda = 0.034s$ and $\lambda = 0.152$s corresponds to fluid VE4 and fluid VE5 respectively.

Table 4.6
Fluid properties of the drop and continuous phase for the drop formation case.

Property	Continuous phase (Newtonian)	Drop phase (Giesekus fluid)
Density (kg/m^3)	984	1000
Viscosity (mPa·s.)	69.5	$\eta_s = 6.05$, $\eta_p = 6.05$
Average velocity (mm/s)	1.197, 2.395	1.197, 2.395
Relaxation time (s)	-	0.034, 0.152
Mobility factor	-	0.003

The parameters, $\rho_d, \rho_c, \eta_c, \eta_d, \boldsymbol{v}_c, \boldsymbol{v}_d, \sigma, \lambda$, employed for the numerical setup were obtained from the experiment and reported in [1]. A summary of the fluid parameters used is delineated in Table 4.6. At both entrances, the same average velocity was specified so that $v_c/v_d = 1$. The Weissenberg number, $Wi = 0.41$, 3.64, was obtained using a relaxation time of $0.034s$ and $0.152s$ respectively.

A qualitative comparison is made between experiment and numerical result in Figs. 4.29 and 4.30. On increasing the capillary number from $Ca = 0.00231$ in Fig. 4.29 to $Ca = 0.00461$ in Fig. 4.30, the formation process changed from squeezing to jetting. A very good agreement was observed between experiment and simulation in this study. In particular both squeezing and jetting behavior were reproduced. The

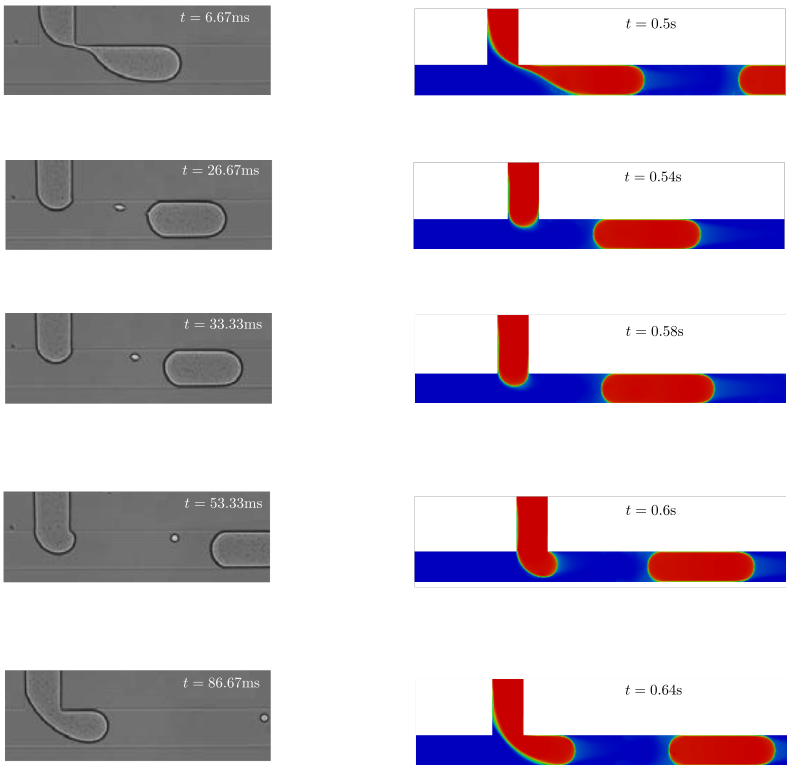


Figure 4.29: Droplet formation process in a T-junction. (a) Experiment: width of both inlets is 100μm and depth is 58μm. (b) Numerical simulation. Parameters: $\eta_c = 44.1$ mPa.s, $\eta_d = 1.05$ mPa.s, $v_c = v_d = 1.197$ mm/s; $Re = 0.00266$, $Ca = 0.00231$, $Wi = 0.41$. Reprinted from "Formation of Uniform Plugs and Monodispersed Droplets for Viscoelastic Fluid," by Li, Xiao-Bin, et al. [1], 2010, Earth and Space, p. 2211. Copyright 2010, with permission from ASCE. See documentation in Appendix E

drop sizes, however, do not match exactly. This is largely due to the two-dimensional approximation of the experiment which was conducted in a three-dimensional planar geometry. Also, a surfactant-laden dispersed phase (CTAC/NaSal aqueous solution) was used in the experiment whereas the numerical simulations were conducted using CTAC/NaSal solution without a surfactant.

In flow conditions, the interfacial stresses between different fluid phases is transformed

by surfactants and this modification is determined by how much surfactant is assimilated and the dynamics of surfactant mass transfer [98]. For this reason, the formation of droplets in a T-junction can be quickened or hindered by surfactants. To gain a deeper understanding of the role a surfactant plays in the droplet formation process, we consider an emerging drop from the injection channel. It grows into a bloated shape that detaches with the aid of the rapid formation and pinching of a neck. The speed with which the neck shrinks is highest when surfactants are not present [99]. Using a dispersed phase with surfactants results in the aggregation of surfactants in the neck region, thereby modifying the dynamic behavior by weakening the surface tension that is majorly responsible for the shrinkage. This ultimately decelerates the thinning process, modifies the structure of the neck and in some cases, inhibit neck formation [99, 100, 101].

(a) (b)

Figure 4.30: Jetting in formation mechanism. (a) Experiment (b) Numerical simulation. Parameters: $\eta_c = 44.1$ mPa.s, $\eta_d = 5.92$ mPa.s, $v_c = v_d = 2.395$ mm/s; $Re = 0.00532$, $Ca = 0.00461$, $Wi = 3.64$. Reprinted from "Formation of Uniform Plugs and Monodispersed Droplets for Viscoelastic Fluid," by Li, Xiao-Bin, et al. [1], 2010, Earth and Space, p. 2211. Copyright 2010, with permission from ASCE. See documentation in Appendix E.

4.4.1 Effect of contact angle

The hydrophobicity of microchannels has been reported to play some role on the formation process of droplets, including their size [102]. Typical range of contact angles commonly published for the aqueous solution is between 105^o and 180^o [97]. The influence of contact angle on droplet size is delineated in Fig. 4.31. As shown, the size of droplets increases as the contact angle rises.

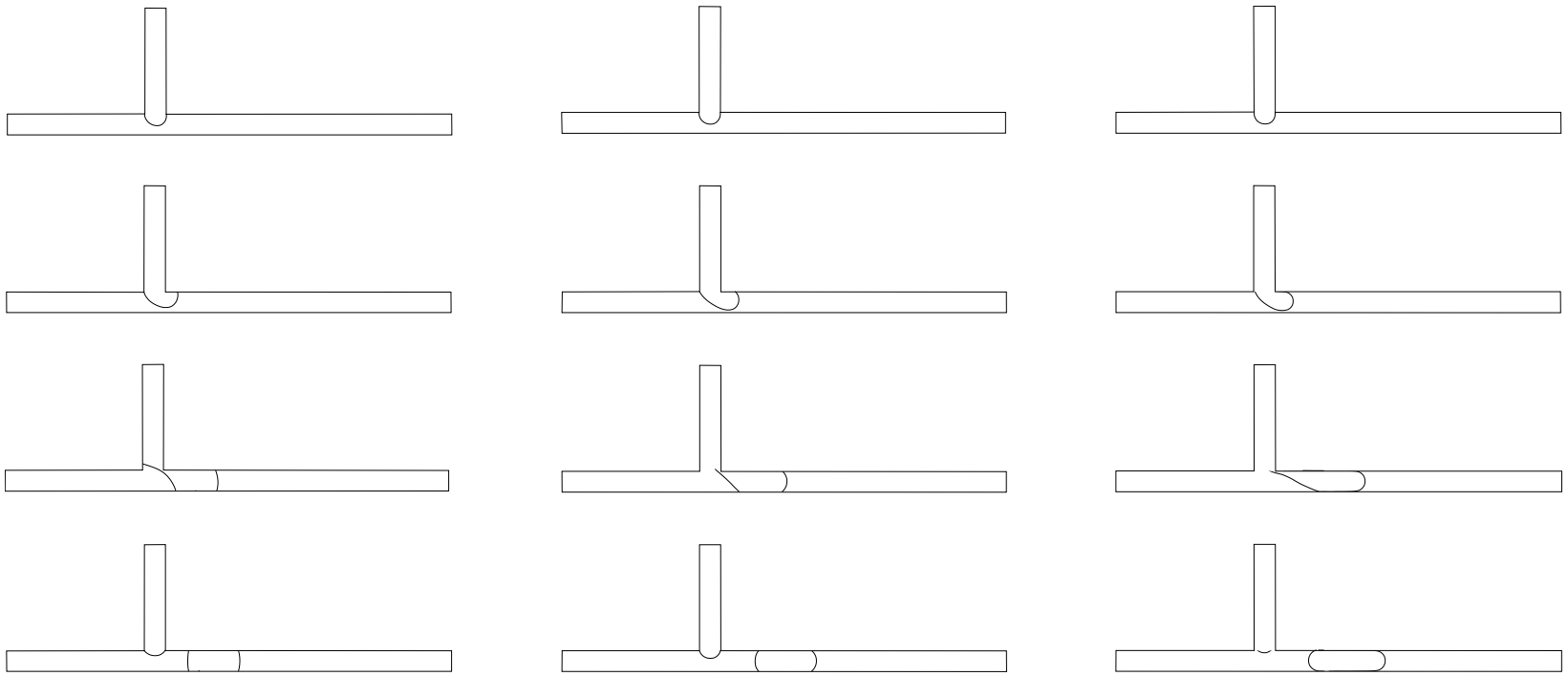


Figure 4.31: Droplet formation patterns for different static contact angles, θ_s.(a) $\theta_s = 105^o$ (b) $\theta_s = 140^o$ (c) $\theta_s = 180^o$. Parameters: $\eta_c = 44.1$ mPa.s, $\eta_d = 1.05$ mPa.s, $v_c = v_d = 1.197$ mm/s; $Re = 0.00266$, $Ca = 0.00231$, $Wi = 0.41$.

4.5 Conclusions

The newly developed solver, `clsVeInterDymFoam`, has been validated by a series of two-dimensional test problems: (1) drop under static conditions; (2) drop deformation in simple shear flow for the cases where the drop is Newtonian and the continuous phase is viscoelastic, and vice versa; (3) rise of a Newtonian drop in a Giesekus fluid; and (4) drop formation in a T-junction under experimental conditions considered by Li et al. [1].

The study revealed that `clsVeInterDymFoam` is superior to `viscoelasticInterFoam` in the static drop problem analysis. The performance of the solver in simulating axisymmetric viscoelastic problems was also evaluated by comparing the results obtained on a dynamic mesh with those from uniform grid calculations for a rising Newtonian drop in a Giesekus fluid. The AMR cases were shown to provide a significant saving in grid size and computation time. Similar results were obtained for the drop deformation in shear flow cases. Based on the result obtained from the parameter study for AMR on drop deformation in shear flow, we recommend using mesh1 as the base mesh and setting the refineRange as $[0.01, 1]$. In addition, our simulations for drop deformation revealed that when a Newtonian drop is subjected to a viscoelastic simple shear flow, the drop deformation is smaller than that for a viscoelastic drop in a Newtonian matrix.

Finally, good qualitative agreement was found for the drop formation in a T-junction case between numerical results and their experimental counterpart.

Chapter 5

Viscoelastic Droplet Formation in a Microfluidic T-junction

Major emulsification methods used to produce uniformly-sized microdroplets include membrane emulsification, microchannel emulsification and microfluidic processes. In membrane emulsification, droplets are generated either as a result of the decomposition of a coarse emulsion after being forced through a membrane channel or shearing of the pure injection source by the continuous phase. The membrane used in membrane emulsification devices could be either fixed or dynamic, where the rotation/ vibration of the membrane also aids in the pinch-off of droplets from the membrane surface [51, 52]. In general, microfluidic devices are categorized as either flow focusing or microfluidic junctions. Of all microfluidic junctions, the T-junction is easiest to

construct [49, 53]. Hitherto, only a few numerical studies have been conducted on the characterization of viscoelastic droplet formation in a Newtonian stream. For example, viscoelastic drop formation at an aperture [66, 67] and in a flow-focusing channel [68]. To the best of our knowledge, no numerical study on the formation of viscoelastic droplets in a T-junction has been considered. A very good understanding of droplet formation mechanism enables the determination of the feasibility and boundary of the use of membrane emulsification in different kinds of applications.

This chapter focuses on the use of two-dimensional T-shaped microchannels to study the formation and detachment of viscoelastic drops in a Newtonian matrix. The role of imposed flow type, channel height, wall shear rate, interfacial tension and elasticity on formation process of viscoelastic droplets in a Newtonian fluid is critically examined.

A schematic diagram of the microchannel used in the present study is illustrated in Fig. 5.1. The width of the continuous phase channel and the dispersed phase channel are respectively denoted by H and D. The dispersed phase, which is characterized as a Giesekus fluid, is injected at normal direction into a flowing stream of the continuous phase, droplets then detach from the tip of the injection source as a result of several factors including the accumulation of pressure upstream of the growing droplet and drag from the continuous phase. Emulsification process with a fixed membrane is captured in the microchannel as the case when the flow of the continuous phase is Poiseuille (P-flow) and for a rotating membrane, the flow of the continuous phase is

plane Couette (C- flow).

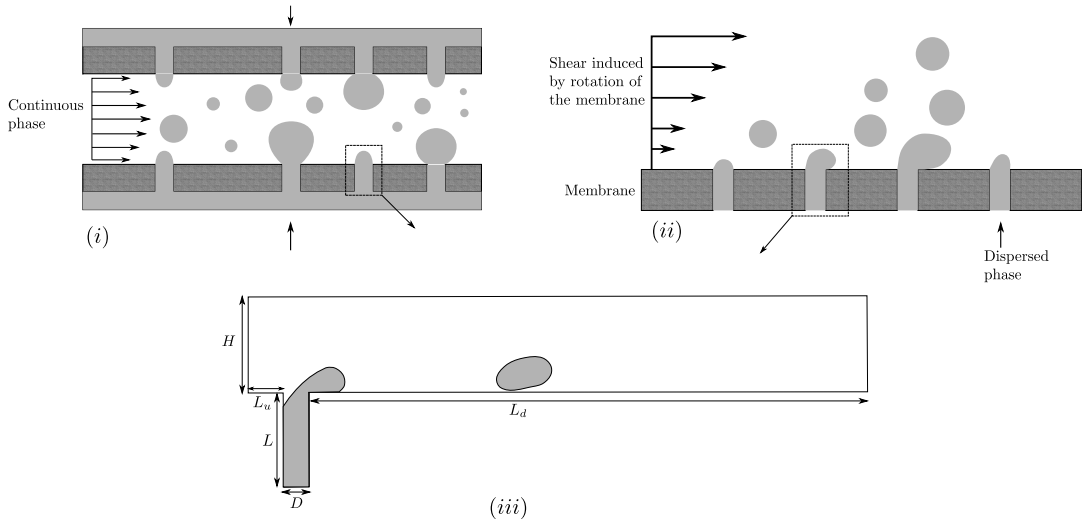


Figure 5.1: (i) Droplet formation in a membrane emulsification device with a fixed membrane (ii) Transverse section of a device similar to rotating membrane device (iii) Schematic representation of droplet formation in a T-cell. H and D denotes the width of the main channel and the dispersed phase channel respectively. $L_u = 975\mu$m, $L = 200\mu$m and $L_d = 2975\mu$m.

Two sets of numerical experiments were conducted. For the first set, $H = 500\mu m$ and $D = 50\mu m$ was used while $H = 100\mu m$ and $D = 50\mu m$ was used for the second set (Section 5.7). The solver, `viscoelasticInterFoam` of Favero [26], was employed for the first set of experiments. Next, selected cases in the first set were run with the modified solver, `clsVeInterDymFoam`, to investigate any difference between the predictions of both solver. For the second group, only the solver `clsVeInterDymFoam` was used. In all simulations, the continuous phase average imposed velocity v_c was varied while the dispersed phase average imposed velocity, v_d remained fixed. Essential dimensionless parameters used in characterizing the observed behavior of droplets formed include the Reynolds number, $Re_c = \rho_c v_c H/\eta_c$, capillary number, $Ca_c = \eta_c v_c/\sigma$,

Deborah number, $De_c = \lambda\dot{\gamma}_{ac}$, $De_d = \lambda\dot{\gamma}_{ad}$ and velocity ratio, v_c/v_d, where ρ_c is the density of the continuous phase; for P-flow, v_c is the same as the uniform velocity prescribed at its inlet, for C-flow, $v_c = v_w/2$, where v_w is the velocity of the upper channel wall; η_c is the continuous phase viscosity; σ is the interfacial tension between both phases; $\dot{\gamma}_{ac}(= 3v_c/H)$ [103] denotes the average imposed shear rate in the continuous phase channel and $\dot{\gamma}_{ad}(= 3v_d/D)$ [103] is the average imposed shear rate in the dispersed phase channel.

The interfacial tension, $\sigma = 0.0415\,N/m$, the density of the continuous phase, $\rho_c = 960\,kg/m^3$, the continuous phase viscosity, $\eta_c = 0.106\,Pa.s$, and the density of the dispersed phase, $\rho_d = 803.87\,kg/m^3$ were fixed for all simulations in this section. The relaxation parameter was set to $\lambda = 5ms$. A summary of these parameter values can be found in Table 5.1. For comparison purposes, the case with these parameter values shall be identified as the base case.

Table 5.1
Fluid properties for drop formation in a t-junction.

Property	Continuous phase	Dispersed phase
Density (kg/m^3)	960	803.87
Viscosity (Pa·s.)	0.106	$\eta_s = 0.002$, $\eta_p = 1.2$
Relaxation time (s)	-	0.005
Mobility factor	-	0.05

The range of values for v_c/v_d, Re_c, Ca_c, De_c, De_d and the channel wall shear rate,

$\dot{\gamma}_w$, considered in this study are given in Table 5.2. For P-flow, the wall shear rate was computed as $\dot{\gamma}_w = 6v_c/H$ while for C-flow, $\dot{\gamma}_w = 2v_c/H$.

Table 5.2
Range of parameter values for P-flow and C-flow.

Type	$\dot{\gamma}_w(1/s)$	v_c/v_d	Re_c	Ca_c	De_c	De_d
$H/D = 2$						
P-flow	3000 – 6000	45.45 – 90.91	0.045 – 0.091	0.13 – 0.25	0 – 15	0 – 0.66
$H/D = 10$						
P-flow	600 – 3000	45.45 – 227.27	0.23 – 1.13	0.13 – 1.33	0 – 12	0 – 0.66
C-flow	600 – 3000	136.36 – 681.81	0.68 – 3.4	0.38 – 4.0	0 – 15	0 – 0.66

At time, $t = 0$, the velocity, $\boldsymbol{v}$, pressure, p, and stress, $\boldsymbol{\tau}_p$ field were set to zero. The volume fraction, ϕ was initialized to 0 in the continuous phase channel and 1 in the channel of the dispersed phase.

For P-flow, $\boldsymbol{v} = 0$ holds at all walls of the computational domain. A zero normal gradient was specified at both inlets for p and $\boldsymbol{\tau}_p$, at the walls for p, $\boldsymbol{\tau}_p$, and at the outlet for $\boldsymbol{v}$ and ϕ. A value of zero was assumed at the outlet for the polymeric stress, $\boldsymbol{\tau}_p$, and pressure, p. The equilibrium contact angle used is 180^o. The boundary conditions for the C-flow case differed from the P-flow case only at the upper wall and inlet of the continuous phase channel where a non-zero fixed value was set and zero normal gradient was specified for $\boldsymbol{v}$ respectively.

5.1 Rheological properties of the dispersed phase (Giesekus fluid)

In steady shear flow, the exact solution for the shear viscosity, η, and first normal stress coefficient, Ψ_1, of the Giesekus model is obtained, respectively, as [104]

$$\eta = \eta_o \left(\frac{\lambda_2}{\lambda} + \left(1 - \frac{\lambda_2}{\lambda} \right) \frac{(1-f)^2}{1+(1-2\alpha)f} \right) \tag{5.1}$$

and

$$\Psi_1 = 2\eta_o(\lambda - \lambda_2)\frac{f(1-\alpha f)}{(\lambda\dot{\gamma})^2\alpha(1-f)}, \tag{5.2}$$

where

$$\eta_o = \eta_s + \eta_p, \tag{5.3a}$$

$$\lambda_2 = \lambda\frac{\eta_s}{\eta_s + \eta_p}, \tag{5.3b}$$

$$f = \frac{1-\chi}{1+(1-2\alpha)\chi}, \tag{5.3c}$$

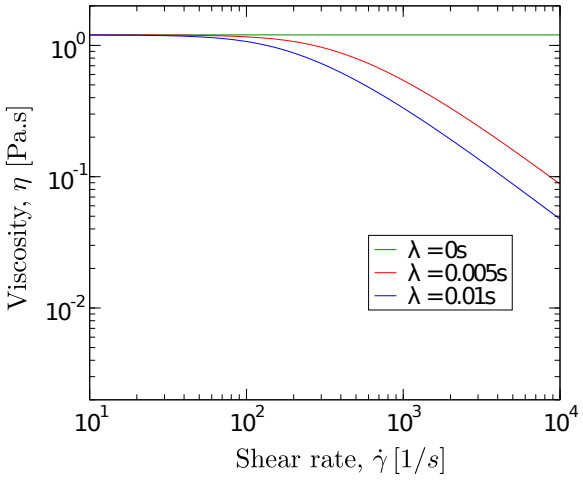
$$\chi^2 = \frac{(1+16\alpha(1-\alpha)(\lambda\dot{\gamma})^2)^{\frac{1}{2}}-1}{8\alpha(1-\alpha)(\lambda\dot{\gamma})^2}, \tag{5.3d}$$

In Eqs. (5.1) to (5.3), λ_2 represents the retardation time and $\dot{\gamma}$ denotes the shear rate.

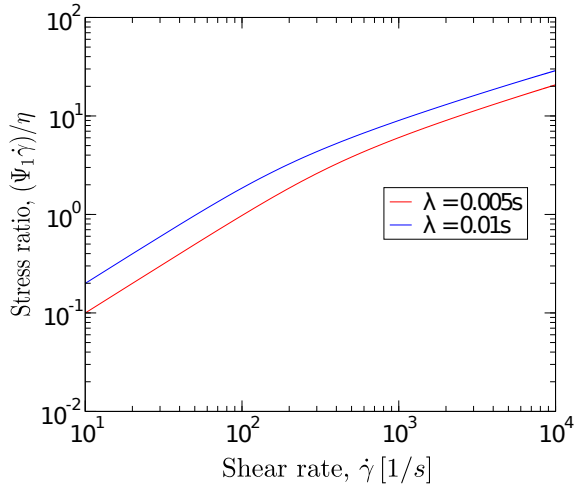
The graphs of viscosity and stress ratio ($\frac{\Psi_1\dot{\gamma}}{\eta}$) against shear rate, obtained using Eqs. (5.1) and (5.2), are shown in Fig. 5.2 for Giesekus fluid in Table 5.1. In (i), the effect of the relaxation time on the shear-thinning property of a fluid is delineated. As the relaxation time increases, the shear rate at which the fluid begins to shear thin drops. As expected, when $\lambda = 0$, the viscosity is independent of shear rate. The stress ratio provides a measure of elasticity in simple shear flow. It can be seen in (ii) that the elasticity of the fluid flow increases as the relaxation time, λ, increases.

5.2 Grid Independence Study

To confirm grid independence of the results obtained in this study using `viscoelasticInterFoam`, numerical simulations were performed on three different meshes [105]. For descriptive purposes, the meshes are identified as *Mesh 1*, *Mesh 2*



(i)



(ii)

Figure 5.2: (i) Viscosity as a function of shear rate. (ii) Stress ratio as a function of shear rate. The following parameter values have been used: $\eta_s = 0.002\,Pa.s$, $\eta_p = 1.2\,Pa.s$ and $\alpha = 0.05$ (see Table 5.1). In (ii), the stress ratio when $\lambda = 0$ (Newtonian) is zero.

and *Mesh 3* in order of fineness with *Mesh 1* being the coarsest. A summary of the characteristics of all meshes are shown in Table 5.3.

Table 5.3
Grid properties for droplet formation in a T-junction.

Mesh	$\triangle x_{min}/D$	$\triangle y_{min}/D$	Number of cells
1	0.091	0.058	55,577
2	0.063	0.038	124,960
3	0.042	0.026	281,160

Mesh 1 is a two dimensional mesh that comprises non-uniform hexahedral cells. The cells around the mouth of the dispersed phase channel are more refined to accurately predict droplet detachment. Cell sizes in this area were $\triangle x/D = 0.091$ and $\triangle y/D = 0.058$, where D is the width of the dispersed phase channel (see Fig. 5.1). On refining *Mesh 1* by a factor of 1.5 in both x- and y- directions, *Mesh 2* was obtained. In the same way, *Mesh 3* was obtained from *Mesh 2*. The effect of wall shear rate on droplet sizes are shown in Fig. 5.3 for all three meshes. We remark here that the size of droplet reported in Fig. 5.3 and subsequent figures in this chapter are the mean of the size of five droplets formed in the main channel.

To give a quantitative account of the discretization error, the Normalized Percent Error (NPE) defined as

$$\text{NPE} = \max_{k=1}^{n} \left(\frac{|x_k^i - x_k^{ref}|}{\max(\boldsymbol{x}^{ref})} \right) \times 100 \tag{5.4}$$

was computed for the result obtained in P-flow and C-flow. In Eq. (5.4), n denotes the number of wall shear rates considered for each mesh (See Fig. 5.3), $i = 1, 2, 3$

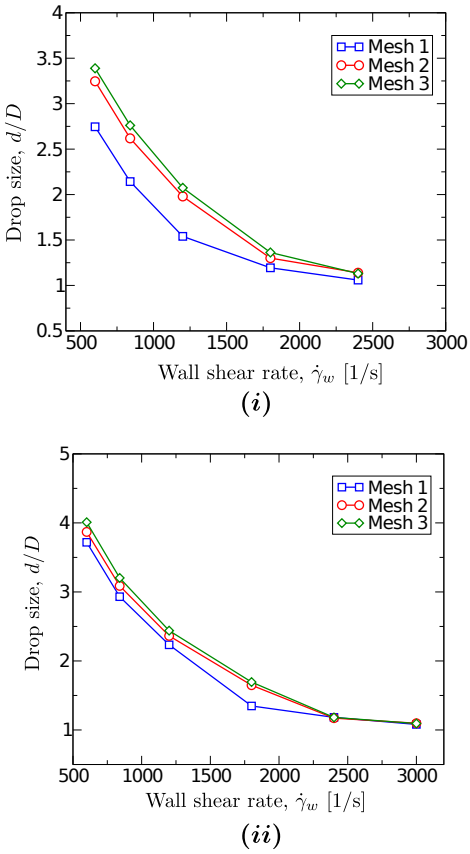


Figure 5.3: Drop size as a function of wall shear rate (i) C-flow (ii) P-flow. For C-flow, the wall shear rate was computed as $\dot{\gamma}_w = \frac{2\boldsymbol{v}_c}{H}$ while for P-flow, $\dot{\gamma}_w = \frac{6\boldsymbol{v}_c}{H}$

identifies the mesh and $ref = 3$ (i.e. *Mesh 3* was chosen as the reference mesh). In P-flow, the NPE for the normalized drop size with *Mesh 2* was obtained as 4% while in C-flow, it was obtained as 5% . Hence *Mesh 2* was used in the remaining simulations with `viscoelasticInterFoam`.

5.3 Influence of flow type

To investigate the effect of imposed channel flow type, two sets of experiments were conducted. The first set was performed in a pressure-driven flow (P-flow) while the second was conducted in a plane Couette flow (C-flow).

The influence of the imposed flow type on the size of droplet is depicted in Fig. 5.4. The case when both flow types have the same velocity ratio, $v_c/v_d = 136.4$, have also been included for comparison. As shown, the droplet size formed in C-flow is larger. This is due to the larger $\dot{\gamma}_{ac}$ in P-flow at the same v_c. Also, as expected for each flow, the droplet size decreases as Ca_c increases. A similar result was reported by Husny and Cooper-White [64] - they investigated the role of elasticity on the droplet formation process and size of droplet formed within a T-junction geometry. This qualitative behavior was also shown in the numerical study conducted by Feigl et. al. [103] for a Newtonian-Newtonian fluid system.

Different behavior was observed concerning the role of the average imposed velocity of the continuous phase, $\boldsymbol{v}_c$ on droplet size. In Fig. 5.5, the leftmost point on each curve connotes the threshold flow rate for which droplet detachment occurs and the right end-point indicates the flow rate above which iterative convergence could not be attained. In principle, the iterative schemes employed diverge beyond some critical

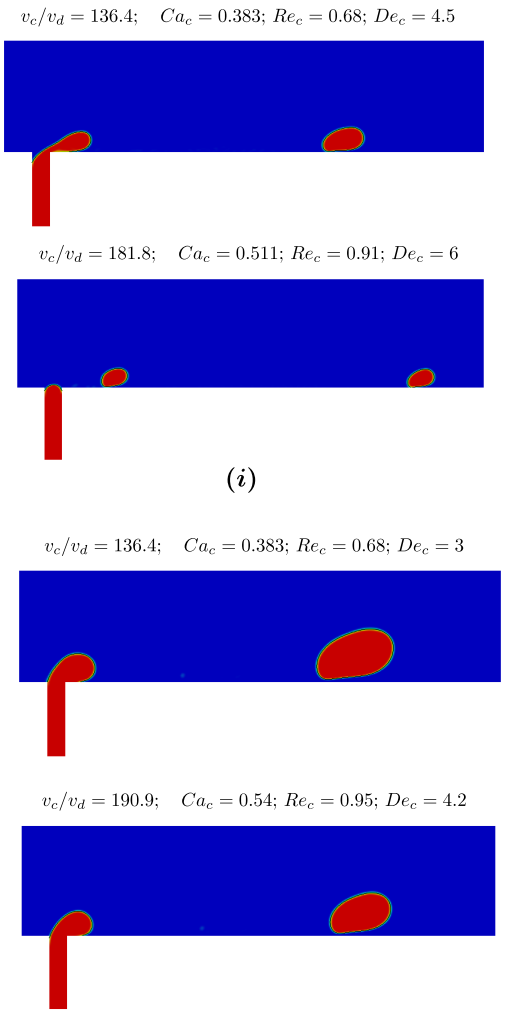


Figure 5.4: Effect of flow type on droplet size. (i) P-flow (ii) C-flow.

value of the Weissenberg number which is due mainly by the hyperbolic nature of the constitutive equations [106]. As a result, the variables solved for e.g. pressure, velocity and stress, begin to grow exponentially in time. All droplet detachment showed a dripping behavior i.e. drops detach directly from the mouth of the injection channel and are transported downstream by the flow of the continuous phase. As shown in Fig. 5.5, which is also evident in Fig. 5.4, a decrease in droplet size is seen as $\boldsymbol{v}_c$

increases. Also noticed is that for each continuous phase flow rate, the size of droplets generated in the C-flow is greater than that in the P-flow. This is expected since the applied shear rates are larger in P-flow than in C-flow for a given v_c. For additional understanding of the differences seen in the droplet size between the flow types, the shear rate just above the mouth of the dispersed phase channel was monitored from time, $t = 0s$ to $t = 1s$. The average imposed velocity of the continuous phase and dispersed phase was kept fixed at $v_c = 0.15\,m/s$ and $v_d = 0.0011\,m/s$ respectively. As shown in Fig. 5.6, just after the initial time, the shear rate in the P-flow had risen more than twice that in the C-flow. This behavior is seen almost throughout the simulations. Consequently, droplets detached in P-flow are smaller at a given $\boldsymbol{v}_c$ due to the presence of higher shear stress. This is also evident by comparing the number of detachments for both cases in Fig. 5.6 represented by open symbols - P-flow detaches about four times as many drops as C-flow.

The relationship between the drop size and the main channel's Reynolds number, capillary number and wall shear rate is depicted in Fig. 5.7. For C-flow, the drop size decreases approximately exponentially with Re, Ca and $\dot{\gamma}_w$ at the same rate, the index of the power law being about -0.744. This value is close to the power law index obtained by Feigl et al. [103] i.e. -0.8. The geometric paramters used in the work of Feigl et al. [103] and the current work are the same, except that Feigl et al. [103] considered a Newtonian-Newtonian fluid system. Also, it could be inferred from Fig. 5.7(iii) that for a given wall shear rate, the drop size produced in P-flow is

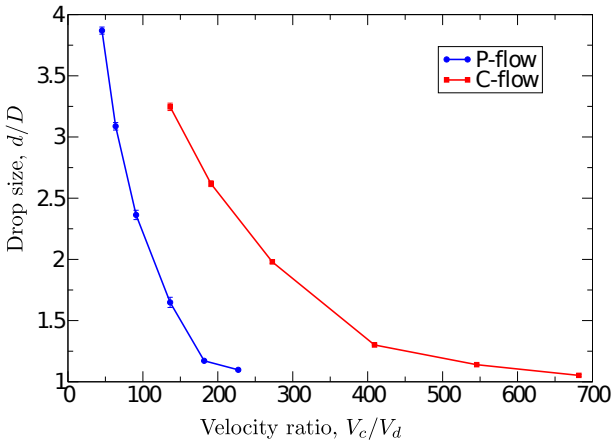


Figure 5.5: Influence of flow type on droplet sizes. V_c is the average velocity for both flow types; The average velocity of the dispersed phase,V_d = 0.0011m/s, η_c = 0.106 Pa·s, η_d = 1.202 Pa·s and $\sigma = 0.0415 N/m$. On the horizontal axis, the droplet size, d, is normalized by the width of the dispersed phase channel, D. The error bars indicate the standard deviation from the mean size, computed from $n = 5$ droplets formed in the channel.

larger than that in C-flow.

5.4 Influence of interfacial tension

Interfacial tension plays a significant role in chemical processes that involves the mixture of two or more immiscible fluids. For example, it influences the likelihood of phases to detach in the production of emulsions. It is also invaluable in the case of flooding during oil production. The use of emulsifiers to reduce interfacial tension aids in assembling of the organic phase after being inundated with water. To account for the effect of interfacial tension on droplet size and formation mechanism, different

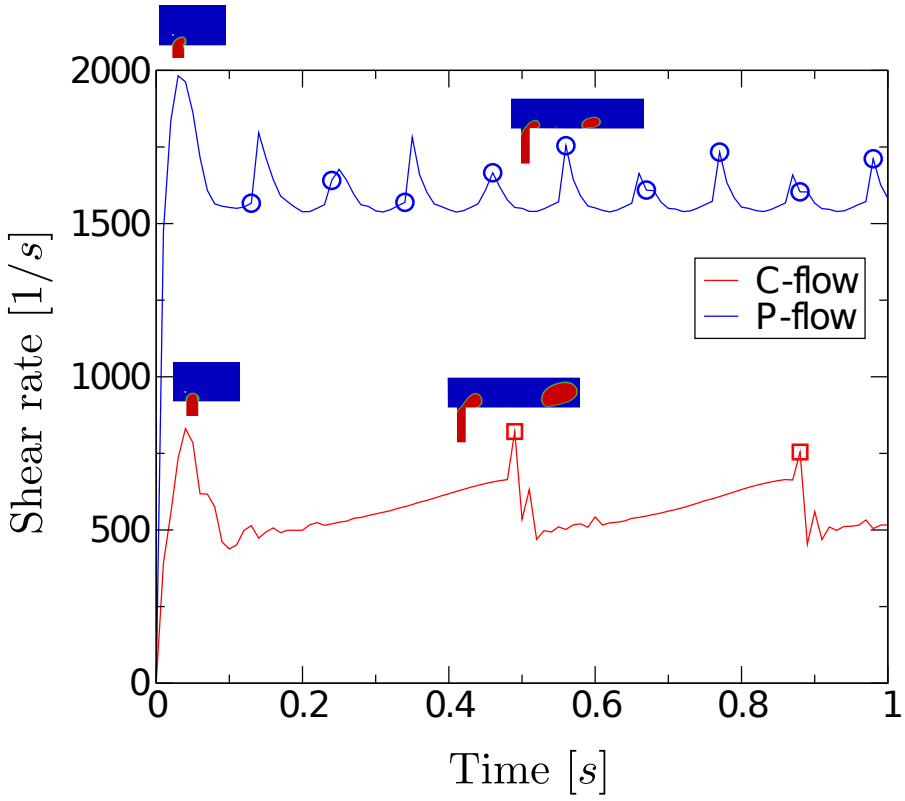
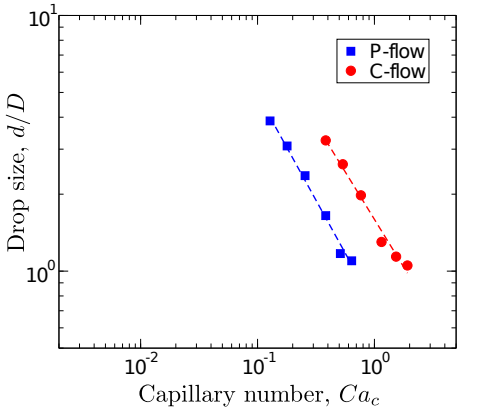


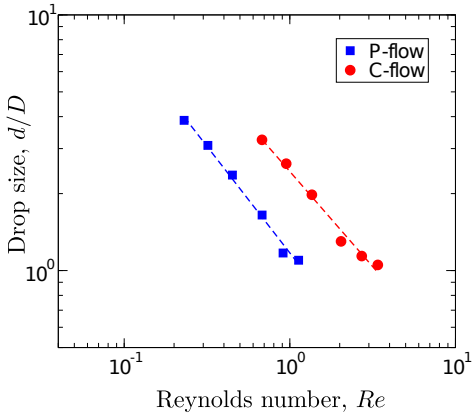
Figure 5.6: Evolution of shear rate at the point,$(-1.84 \times 10^{-5}m, 4.75 \times 10^{-5}m, 0)$, just upstream the t-junction. Open symbols indicate the time when droplet detachment occurs.

cases were set up with the parameters kept fixed as the base case while the interfacial tension coefficient was varied from $0.02N/m$ to $0.0415N/m$ for both P-flow and C-flow.

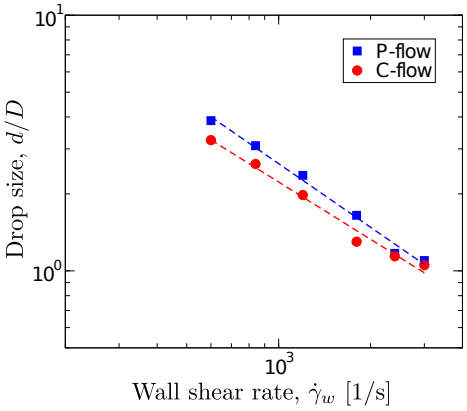
The effect of interfacial tension on the size of droplet is depicted in Fig. 5.8. For a fixed average imposed velocity of the continuous phase, $\boldsymbol{v}_c$, a direct relationship is seen between the drop size and interfacial tension. Droplet sizes smaller than the width of the dispersed phase channel were obtained at high $\boldsymbol{v}_c$ for $\sigma = 0.02N/m$ and



(*i*)



(*ii*)



(*iii*)

Figure 5.7: Drop size as a function of (i) Capillary number (ii) Reynolds number (iii) Wall shear rate. The dotted lines in each figure represent the line of best fit for each flow. The slopes of the blue and red dotted lines in (i), (ii) and (iii) are -0.827 and -0.744, -0.834 and -0.744, and -0.826 and -0.744 respectively.

$0.03N/m$.

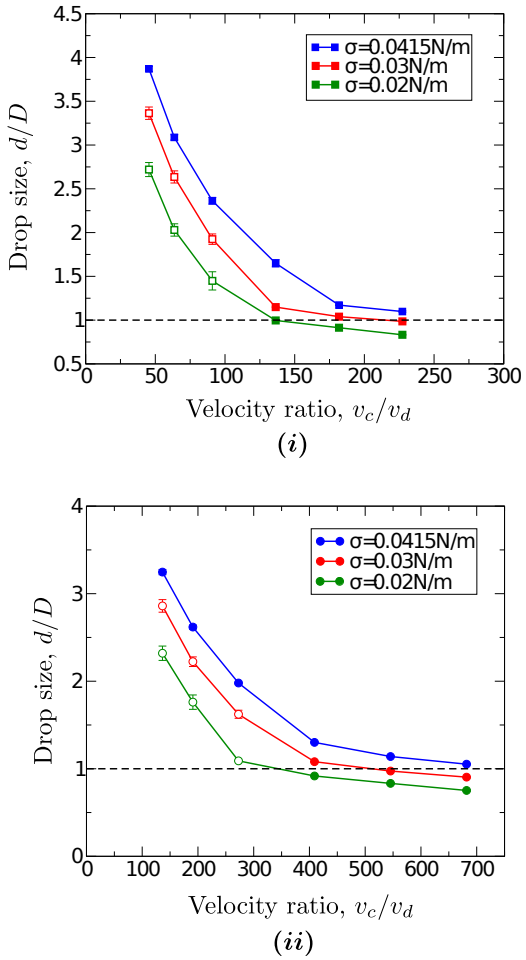


Figure 5.8: Effect of interfacial tension on drop size (i) P-flow (ii) C-flow. The symbols (■, •) and (□, ○) connotes dripping and jetting regime respectively.

The interfacial tension had a significant effect on droplet detachment behaviour. Two pinch-off regimes were observed as the interfacial tension was varied; dripping regime, where drops were generated at the tip of the pore channel and jetting regime, where there is an incomplete draw back of the neck after droplet pinch off from the tip

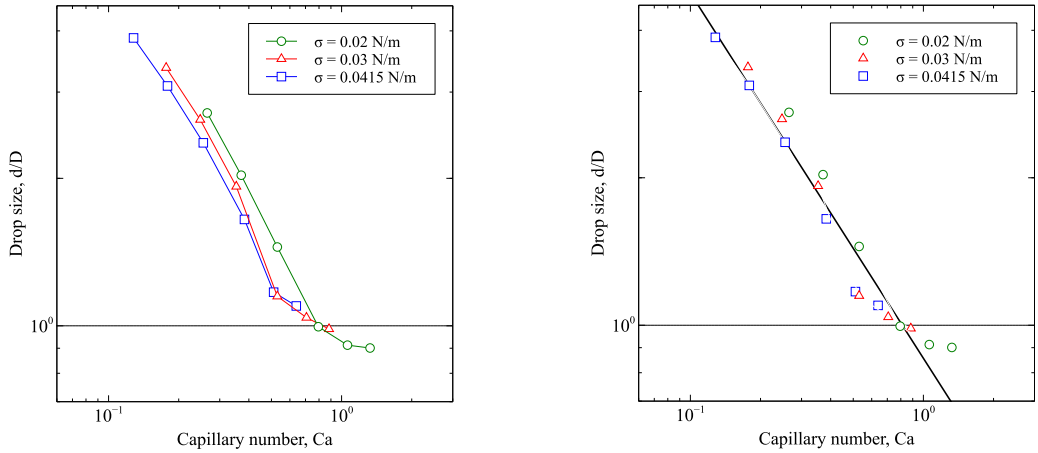


Figure 5.9: Log-log plot of drop size against capillary number (a) P-flow (b) C-flow.

of the filament. For the case with $\sigma = 0.0415N/m$ (base case), all droplets were formed in the dripping regime. With a reduced interfacial tension i.e. $\sigma = 0.03N/m$ and $0.02N/m$, a transition from jetting to dripping was seen as the cross-flow shear increased. Although not shown in Fig. 5.8, we observe that the critical average velocity of the continuous phase, v_c, at which the transition occurred is greater when

$\sigma = 0.02N/m$. The log-log plot of drop size against the capillary number for P-flow and C-flow is compared in Fig. 5.9. We observe in Fig. 5.9 that the predictions from C-flow appear to match a power law better than P-flow. The slopes of the lines of best fit for P-flow and C-flow reported in (a) and (b) are $-0.7455(\chi^2 = 86.35)$ and $-0.6932(\chi^2 = 60.50)$ respectively. Fig. 5.10 compares the snapshot of the volume fraction field when $v_c = 0.07m/s$ for $\sigma = 0.0415N/m$ and $\sigma = 0.02N/m$ immediately after droplet detaches from the parent source. As shown, dripping and jetting is seen in $\sigma = 0.0415N/m$ and $\sigma = 0.02N/m$ respectively. To gain a deeper understanding of this transition, we focus on the mechanism of drop formation in a T-junction. Before droplets are created, the tip of the dispersed phase fluid intrudes into the main channel, and then the tip grows under the balance of interfacial tension, shearing force and the pressure drop between the front and rear of the emerging droplet. Reducing the interfacial tension beyond a critical number results in the inertial force of the fluid emerging from the injection source exceeding the interfacial tension force, leading to a transition to jetting behavior. The jetting behavior can also be explained in general terms by the competition of different time scales. When the pinch-off time scale is longer than the time scale to form a blob of the dispersed phase, the jetting behavior is seen.

The impact of interfacial tension on the monodispersity of droplets formed was also considered in this study. Fig. 5.8 shows the standard error based on the size of the first five droplets formed in the microchannel. As discussed in the previous paragraph,

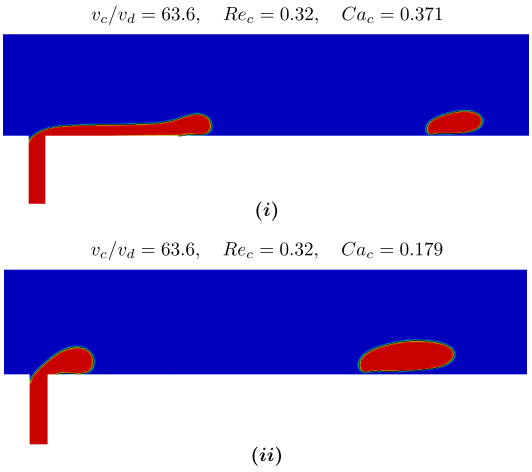


Figure 5.10: Transition from jetting to dripping as interfacial tension increases in P-flow. (i) $\sigma = 0.02N/m$ (ii) $\sigma = 0.0415N/m$

the droplet formation mechanism changes as σ is varied. When $\sigma = 0.0415N/m$, all droplets formed showed dripping behavior for the ranges of v_c considered and monodispersity was high. However, as σ reduced to $0.03N/m$ and $0.02N/m$, both dripping and jetting behavior was seen. Although the monodispersity is high in the dripping regime for all cases, even though small secondary drops may appear (as shown in Fig. 5.11, $\sigma = 0.02N/m$), this worsened as σ is reduced in the jetting regime. In P-flow, the maximum coefficient of variation, CV_{max}, based on 5 detached droplets for $\sigma = 0.0415, 0.03$ and 0.02N/m was 2.5%, 3.1% and 7.2% respectively while in C-flow, CV_{max} for $\sigma = 0.0415$, 0.03 and 0.02 N/m were obtained as $0.93\%, 2.83\%$ and 4.67% respectively. At a fixed wall shear rate, the droplets generated in C-flow were thus found to be more monodisperse.

Satellite droplets are drops formed along with the primary drop as a result of the non-linear behavior of the fluid motion near the pinch-off point [107], hence also known as secondary droplets. In many applications, the occurrence of satellite droplets is an undesirable phenomena. With the same test cases, the formation of satellite drops was seen at high velocity ($v_c = 0.25m/s$) for $\sigma = 0.02N/m$ and $0.03N/m$ only in the P-flow but not in C-flow. This corresponds to a fixed $Re_c = 1.13$, and $Ca_c = 1.33$ and 0.71 respectively. For $\sigma = 0.02N/m$, satellite droplets were formed immediately after every primary drop generated at the T-junction. On the other hand, for $\sigma = 0.03N/m$, the first secondary droplet was formed after several primary drop had been formed and this was at $t = 0.19s$. Fig. 5.11 compares the droplet generation process for three cases with the same parameters but only differing in interfacial tension. For the top, $\sigma = 0.02N/m$, the middle, $\sigma = 0.03N/m$ and the bottom, $\sigma = 0.0415N/m$. Fig. 5.12 shows a zoomed image illustrating the mesh resolution around a newly formed satellite drop. The number of cells in satellite drop indicated that this drop is not a numeric artifact. The Weber number of the dispersed phase, We_d, for the cases shown in Fig. 5.11 were $O(10^{-6})$. Hence the inertia force from the dispersed phase could be neglected. It can be inferred that reducing interfacial tension precipitates satellite droplet formation and this occurs above a critical Ca_c.

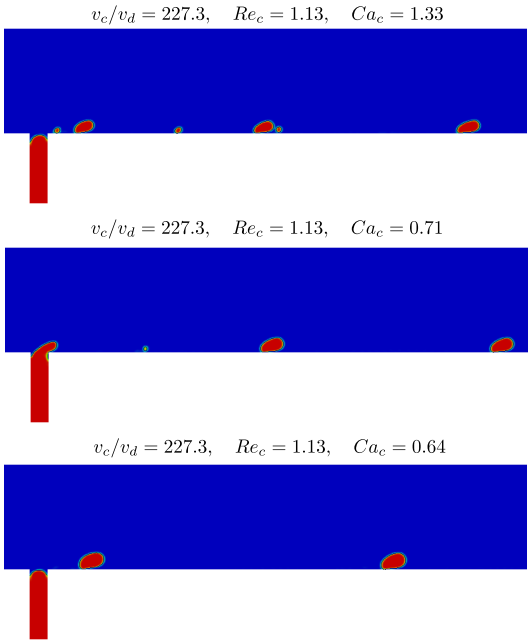


Figure 5.11: Satellite droplet formation in P-flow. Top: $\sigma = 0.02N/m$, Middle: $\sigma = 0.03N/m$, Bottom: $\sigma = 0.0415N/m$.

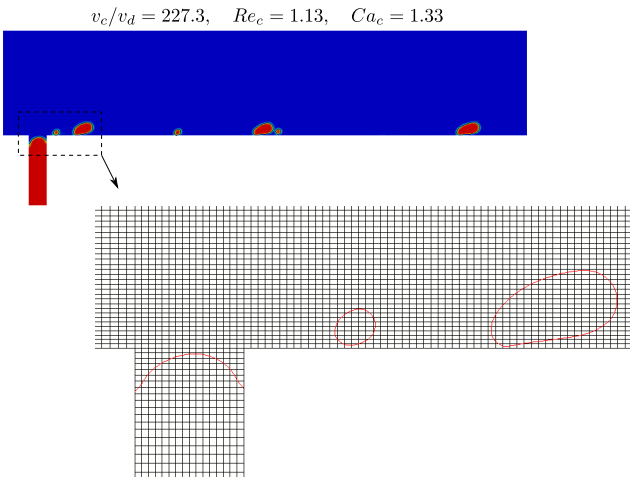


Figure 5.12: Zoomed image of satellite drop for P-flow with $\sigma = 0.02N/m$.

5.5 Influence of elasticity

In this section, the effect of elasticity on droplet size, detachment and filament dynamics is investigated. The elasticity, which was measured via the Deborah number, was raised by increasing the relaxation time, λ, while keeping the average inlet velocity of the dispersed phase, v_d, fixed. The relaxation time was varied between $0s$ and $0.01s$ for each fixed v_c as in the base case.

The snapshot shown in Fig. 5.13 depicts the droplet formation process in a P-flow for two cases with similar parameters but differed only in the amount of elasticity - a Newtonian dispersed phase (N) and a viscoelastic dispersed phase (V). As both fluids enter the continuous phase stream, droplet pinch-off is seen to occur further downstream in (N) and thus has a longer filament. When the droplet detaches, the front of the Newtonian fluid retracts to a position further downstream than the viscoelastic fluid. Also seen in Fig. 5.13 is that the interface at the pinch-off point has higher curvature in (V) than (N).

At low shear rates, the viscoelastic fluid had similar behavior to the Newtonian; increasing λ appears to have no effect on the drop size. On increasing v_c, the effect of elasticity heightened. Fig. 5.14 is a plot of drop size as a function of relaxation time for a case with low velocity ($v_c = 0.07m/s$) and another with high velocity

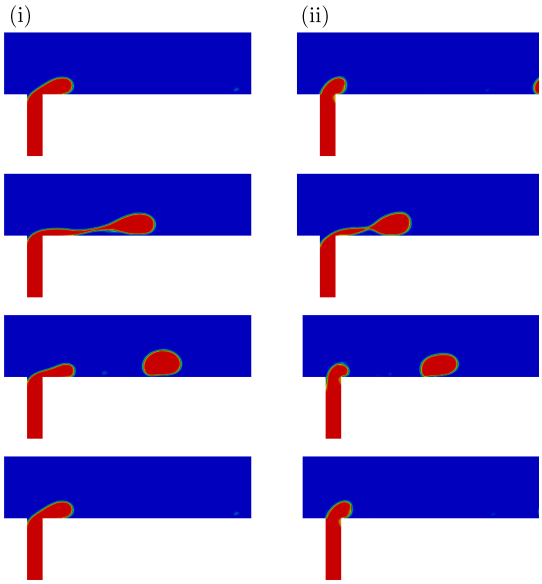


Figure 5.13: The effect of elasticity in the droplet generation process at a fixed dispersed phase flow rate ($v_d = 0.0011m/s$), viscosity ratio, $\eta_d/\eta_c = 11.34$ and $v_c/v_d = 181.8$. (i) Newtonian ($De_c = 0$) (ii) Viscoelastic ($De_c = 12$).

($v_c = 0.2m/s$). A slight decrease in drop size is seen for the case with high v_c as λ increases. Husny and Cooper-White [64] studied the influence of elasticity on the pinch-off dynamics and size of droplet formed within a T-junction geometry, using silicone oil as the continuous phase and both a Newtonian and Boger fluid as the dispersed phase. They reported that the presence of elasticity had no effect on the droplet formation time and concluded that although elasticity had a strong impact on the necking behavior of the injection source, its effect on the resultant droplet size is minimal.

Finally, we investigate the growth pattern of the filament. All calculations were taken after the first droplet had detached from the injection source. The evolution of the

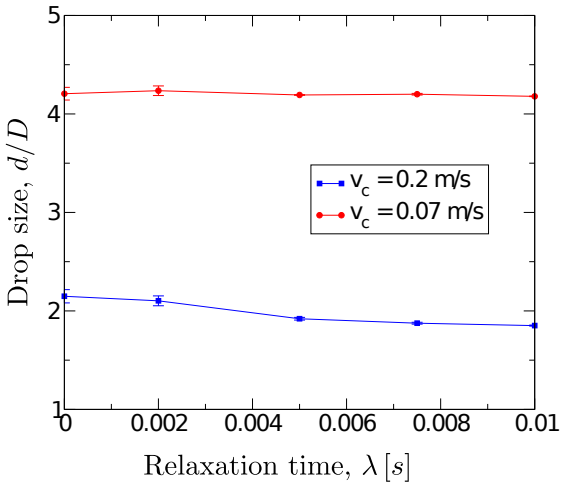


Figure 5.14: Droplet size as a function of relaxation time (elasticity) in P-flow. The effect of elasticity becomes more pronounced as $\boldsymbol{v}_c$ increases.

dispersed phase front (or filament length) for different relaxation times at a low and high v_c is delineated in Fig. 5.15. The increase in sparsity of the symbols as time proceeds indicates the non-linearity of fluid motion close to time of breakup. The growth rate was found to be almost the same for all cases. In particular, when v_c is low, elasticity does not appear to have any effect on the growth rate of the filament. Although, the drop formation time reduces as elasticity increases.

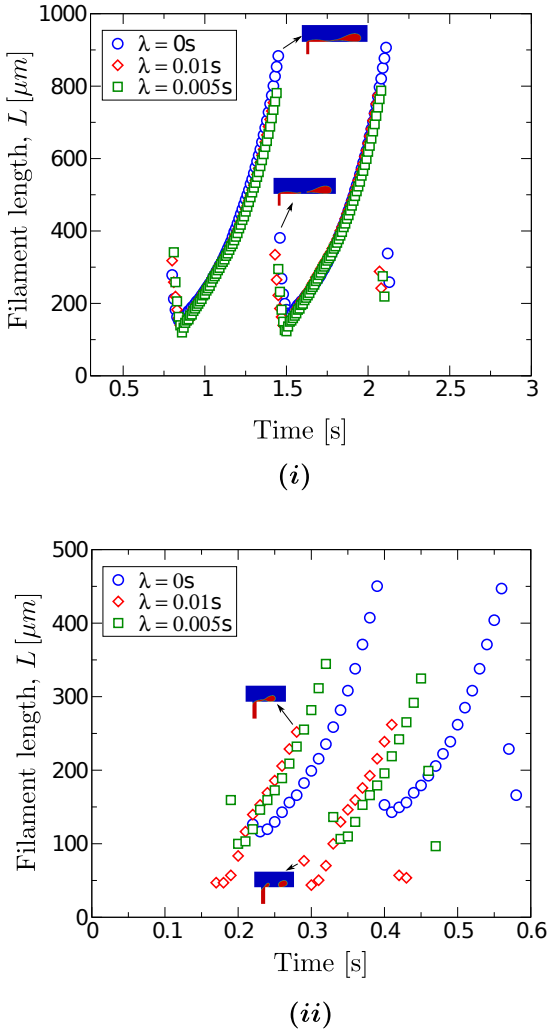


Figure 5.15: Effect of elasticity on droplet growth dynamics. (i) $v_c = 0.07\,m/s$ (low shear) (ii) $v_c = 0.2\,m/s$ (high shear).

5.6 Comparisons between `viscoelasticInterFoam` and `clsVeInterDymFoam` solver

In this section, we explore the differences in the prediction of drop sizes by the old solver, `viscoelasticInterFoam`, and the newly developed solver, `clsVeInterDymFoam`, for the P-flow case. With the new solver, the curvature is now computed using the level-set method in addition to refining the mesh automatically in regions where the volume fraction field, ϕ, falls between 0.01 and 0.99. The same fluid parameters as for the previous sections were used for continuous and dispersed phase for all cases in this section. Details of the material properties for each phase can be found in Table 5.1. Also, the initial and boundary conditions were no different from the case with $H = 500\mu$m (see section 5.1). Next, `clsVeInterDymFoam` was used to simulate droplet formation in a similar T-junction with the same width for the dispersed phase inlet, $D = 50\mu$m, but a smaller channel height, $H = 100\mu$m. The range of values for v_c/v_d, Re_c, Ca_c, De_c, De_d and $\dot{\gamma}_w$ can be found in Table 5.2.

The current case has a base mesh with size, $\triangle x/D = 0.08$. Using two levels of refinement, the size of the smallest cell i.e. around the interface is given by $\triangle x/D = 0.02$. A summary of the parameter settings for dynamic mesh refinement used with all cases is outlined in Fig. 5.16. For a smooth run of this setup, a summary of

the settings used for the linear solvers (in PISO mode) is delineated in Fig. 5.17. A detailed explanation of what each keyword in Figs. 5.16 and 5.17 connote can be found in the previous chapter.

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      dynamicMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dynamicFvMesh   dynamicRefineFvMesh2D;

dynamicRefineFvMesh2DCoeffs
{
    refineInterval  1;

    field           alpha.phase1;

    lowerRefineLevel 0.01;
    upperRefineLevel 0.99;

    unrefineLevel   10;

    nBufferLayers   2;

    maxRefinement   2;

    maxCells        200000;

    correctFluxes
    (
     (phiAlpha none)
     (phi none)
     (nHatf none)
     (rhoPhi none)
     (ghf none)
    );

    dumpLevel       true;
}


// ************************************************************************* //
```

Figure 5.16: Summary of dynamic mesh parameters for P-flow case.

```
solvers
{
    "alpha.phase1.*"
    {
        nAlphaCorr      1;
        nAlphaSubCycles 3;
        cAlpha          1;
    }

    p_rgh
    {
        solver          GAMG;
        tolerance       1e-08;
        relTol          0.01;
        smoother        DIC;
        nPreSweeps      0;
        nPostSweeps     2;
        nFinestSweeps   2;
        cacheAgglomeration false;
        nCellsInCoarsestLevel 10;
        agglomerator    faceAreaPair;
        mergeLevels     1;
    }

    p_rghFinal
    {
        $p_rgh;
        relTol          0;
    }

    "pcorr.*"
    {
        $p_rghFinal;
        tolerance       0.0001;
    }

    U
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        tolerance       1e-06;
        relTol          0;
        nSweeps         1;
    }

    tau
      {
        solver          smoothSolver;
        smoother        GaussSeidel;
        tolerance       1e-06;
        relTol          0;
        nSweeps         1;
      }
}

PIMPLE
{
    momentumPredictor no;
    nCorrectors     3;
    nNonOrthogonalCorrectors 0;
    pdRefCell       0;
    pdRefValue      0;
    pRefCell        0;
    pRefValue       0;
}
```

Figure 5.17: Summary of linear solver settings for P-flow case.

A comparison of the influence of interfacial tension on drop size using `viscoelasticInterFoam` and `clsVeInterDymFoam` is illustrated in Fig. 5.18. The graph obtained using both solvers are qualitatively similar. For each velocity ratio, the drop size increases as interfacial tension increases. For most cases, the prediction of drop sizes by `clsVeInterDymFoam` is greater than that of `viscoelasticInterFoam`.

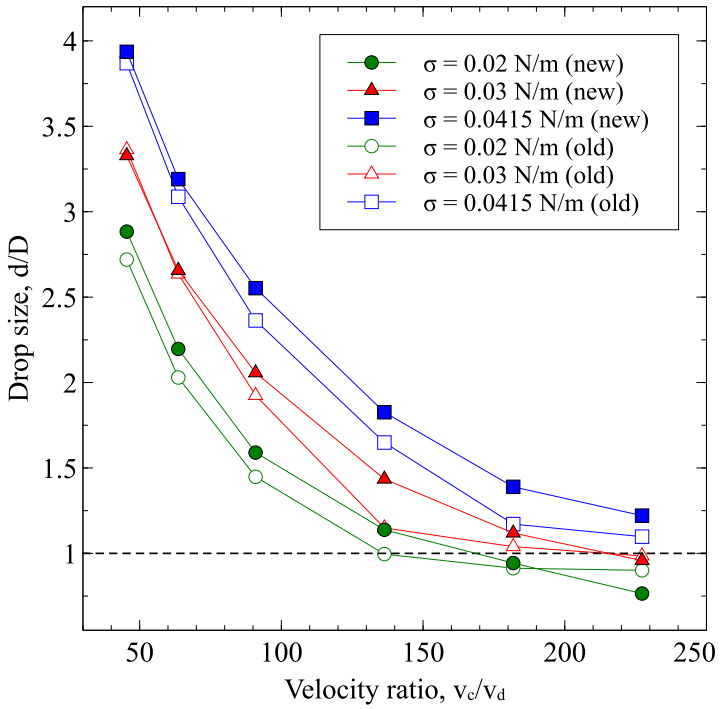


Figure 5.18: Effect of interfacial tension on drop size for P-flow: (old)`viscoelasticInterFoam` and (new)`clsVeInterDymFoam` .

Figure 5.19 compares the drop size against wall shear rate as obtained by the old and new solver. Each plot was created for three cases: (a) $\sigma = 0.02$N/m, (b) $\sigma = 0.03$N/m and (c) $\sigma = 0.0415$N/m. In all cases, the drop size predicted by both solver look qualitatively similar. However, the drop size prediction of the new solver is slightly

larger than that of the old solver in most cases.

The log-log plot of drop size against the capillary number for the two solvers is compared in Fig. 5.20. For both solvers, the curves for $\sigma = 0.03$N/m and $\sigma = 0.0415$N/m almost match. Also seen in Fig. 5.20 is that the predictions from `clsVeInterDymFoam` appear to match a power law better than `viscoelasticInterFoam`. The slopes of the lines of best fit for `viscoelasticInterFoam` and `clsVeInterDymFoam` reported in (a) and (b) are $-0.7455(\chi^2 = 86.35)$ and $-0.7373(\chi^2 = 30.34)$ respectively.

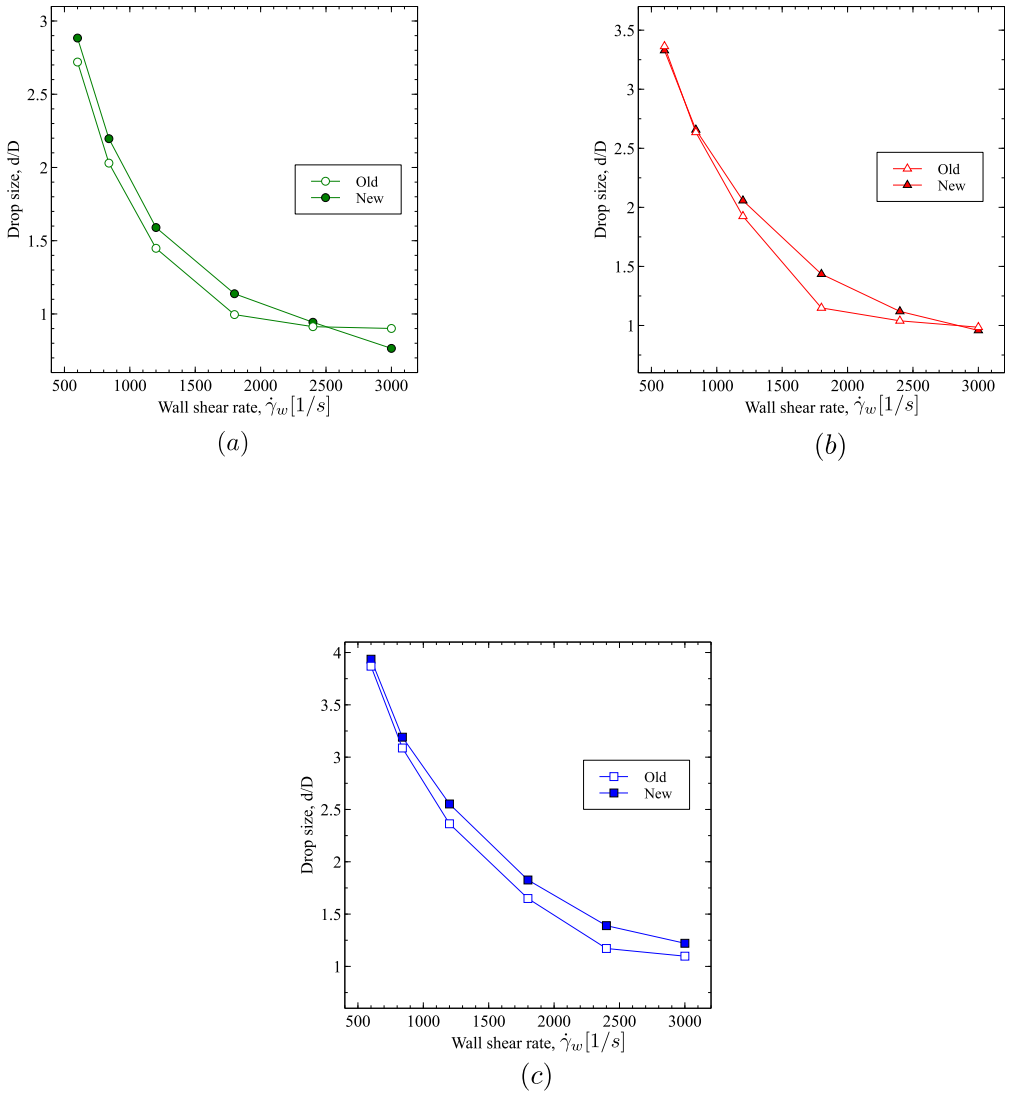
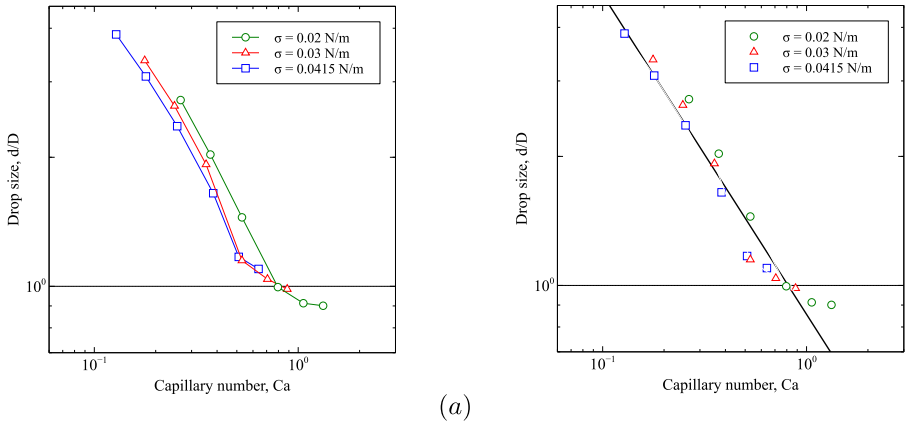
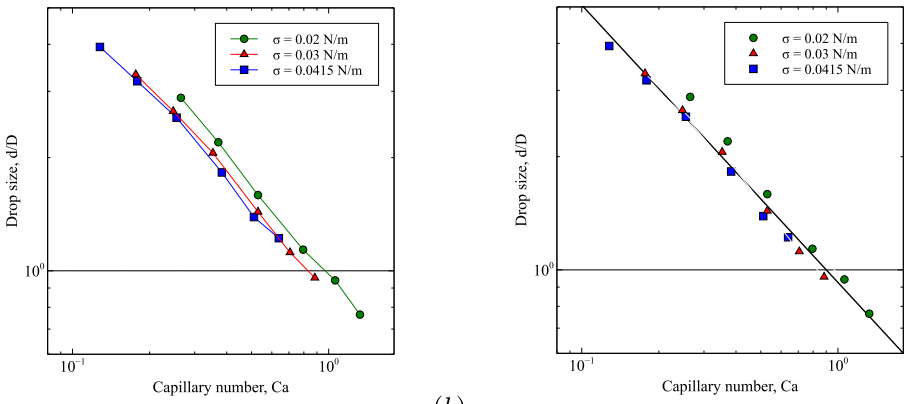


Figure 5.19: Comparison of drop size against wall shear rate using `viscoelasticInterFoam` (Old) and `clsVeInterDymFoam` (New) solver. (a) $\sigma = 0.02$N/m (b) $\sigma = 0.03$N/m (c) $\sigma = 0.0415$N/m.



(*a*)



(*b*)

Figure 5.20: Log-log plot of drop size against capillary number using (a) `viscoelasticInterFoam` (Old) and (b) `clsVeInterDymFoam` (New) solver.

Figures 5.21 and 5.22 are selective snapshots of the droplet formation processes when $v_c/v_d \approx 45.45$ and $v_c/v_d \approx 227.27$ respectively. The detachment behavior seen for all simulation cases were not all the same for both solvers. Jetting behavior for `viscoelasticInterFoam` still remained jetting with `clsVeInterDymFoam` but dripping behavior seen in `viscoelasticInterFoam` for large $\boldsymbol{v}_c$ changed when ran with `clsVeInterDymFoam`. For example, Fig. 5.23 compares the formation mechanism for the case with velocity ratio, $v_c/v_d = 181.82$ and interfacial tension, $\sigma = 0.0415$N/m. Not only is the size of drop formed with `clsVeInterDymFoam` greater, the detachment of droplets is in the transition regime for `clsVeInterDymFoam` while it shows dripping with `viscoelasticInterFoam`. The reason for the differences is not known.

Of particular interest in this study was to determine if the satellite droplet which was obtained using `viscoelasticInterFoam` (see Fig. 5.11) could also be reproduced using `clsVeInterDymFoam`. As shown in Fig. 5.24, satellite droplets was also generated in a similar case ran with `clsVeInterDymFoam`. We note, however, that the satellite droplet were not formed in a regular fashion as obtained with `viscoelasticInterFoam`.

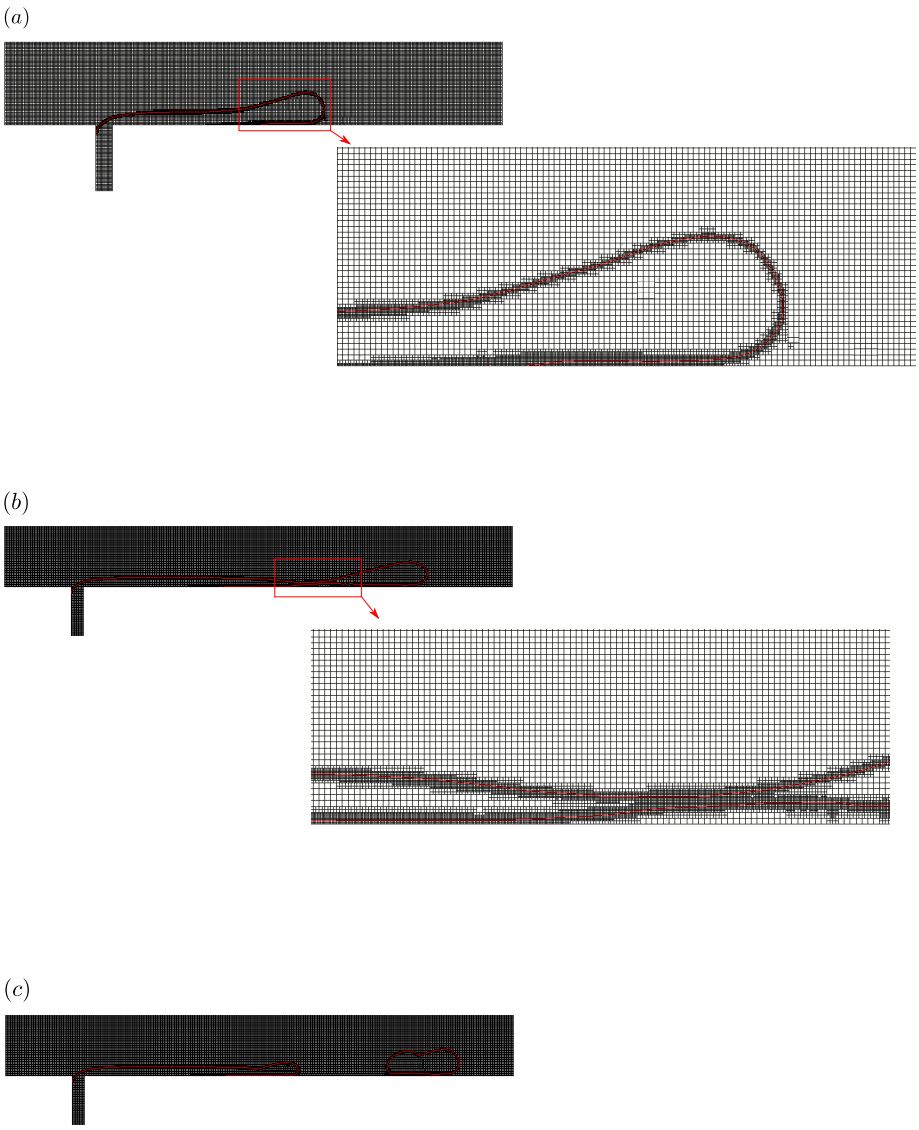


Figure 5.21: Droplet formation process in a T-junction using `clsVeInterDymFoam` for $v_c/v_d \approx 45.45, \sigma = 0.02$N/m. (a) $t = 0.62$s (b) $t = 0.76$s (c) $t = 0.82$s. The size of the smallest cell i.e. around the interface is given by $\triangle x/D = 0.02$.

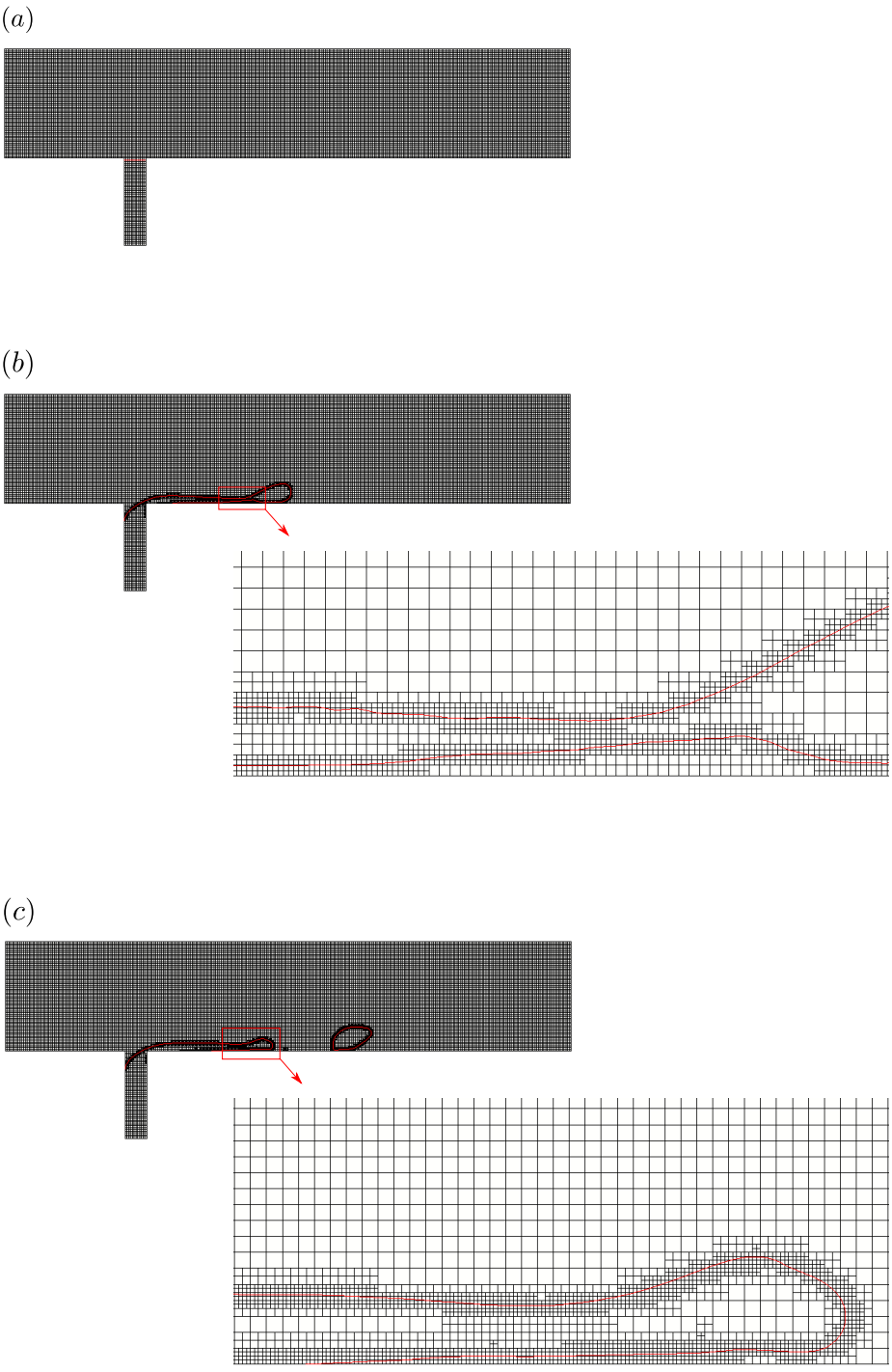


Figure 5.22: Droplet formation process in a T-junction using `clsVeInterDymFoam` for $v_c/v_d \approx 227.27$, $\sigma = 0.0415$N/m. (a) $t = 0$s (b) $t = 0.1$s (c) $t = 0.2$ s

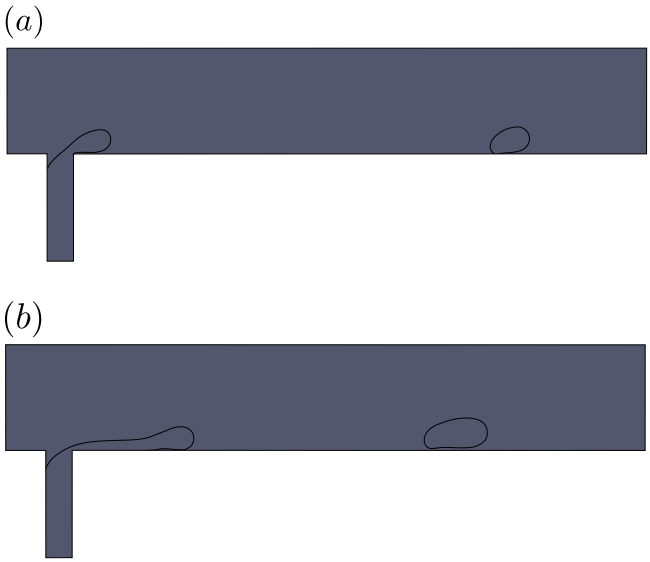


Figure 5.23: Comparison of droplet formation for the case, $v_c/v_d \approx 181.82$, $\sigma = 0.0415$N/m (a) `viscoelasticInterFoam` (b) `clsVeInterDymFoam`.

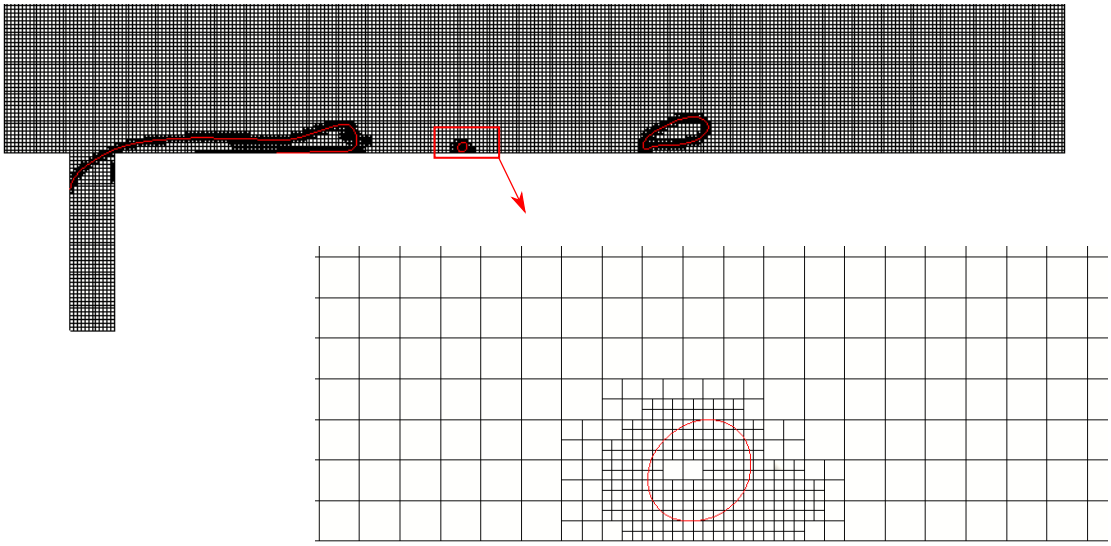


Figure 5.24: Formation of satellite droplet for the case, $v_c/v_d \approx 227.27$, $\sigma = 0.02$N/m using `clsVeInterDymFoam`.

5.7 Influence of channel height

In this section, we study the effect of reducing the channel height on drop size and detachment behavior while keeping other parameters fixed. In addition, we also examined, with the reduced channel height, the effect of elasticity on droplet formation mechanism. All simulations were ran using `clsVeInterDymFoam`.

Figure 5.25 compares the graph of drop size against velocity ratio for two channel heights, $H = 500\mu m$ and $100\mu m$. For the case when $H/D = 2$ ($H = 100\mu m$ and $D = 50\mu m$), the left end of the curve relates to the lowest v_c for which drop detachment occurs in the main channel. In other words, jetting with no detachment was seen at lower $\boldsymbol{v}_c$, as shown in Fig. 5.27. The right end coincides with the $\boldsymbol{v}_c$ above which iterative convergence could not be achieved ($De_c = 15$). Dripping behavior was seen when the velocity ratio, v_c/v_d is 45.45 and 63.64 while jetting behavior was seen when v_c/v_d is 81.82 and 90.91.

We observe from Fig. 5.25 and 5.28 that for a given v_c, the size of droplets generated in the main channel increases as the channel height, H, increases. The reason is evident from Fig. 5.26 where we observe that the droplet detachment process in the smaller channel is subjected to a higher shear rate at a given v_c. Consequently, the drops don't have enough time to grow and they detach earlier.

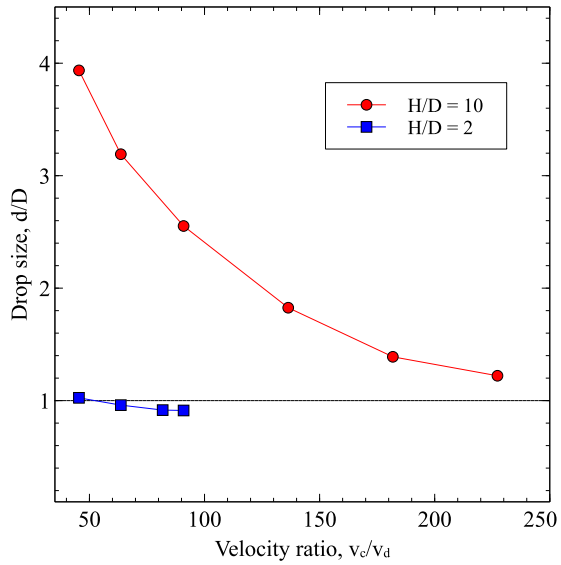


Figure 5.25: Comparison of drop sizes as a function of velocity ratio between the cases with channel height, $H = 500\mu m$ and $100\mu m$.

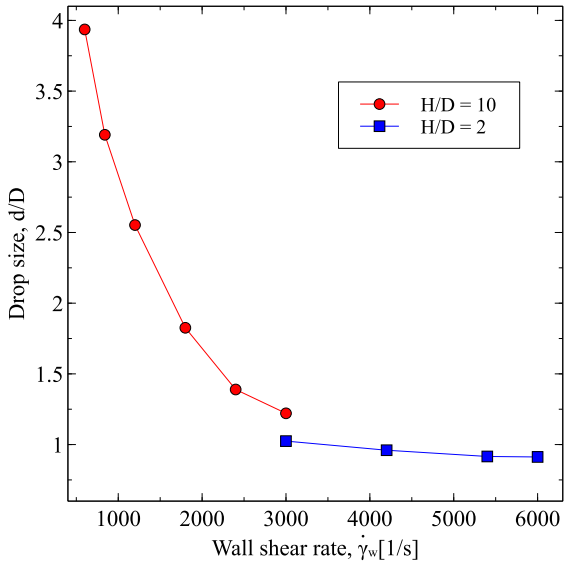


Figure 5.26: Comparison of drop sizes as a function of wall shear rate between the cases with channel height, $H = 500\mu m$ and $100\mu m$.

Figure 5.27: Parallel flow for $v_c/v_d \approx 27.27, \sigma = 0.0415\text{N/m}$.

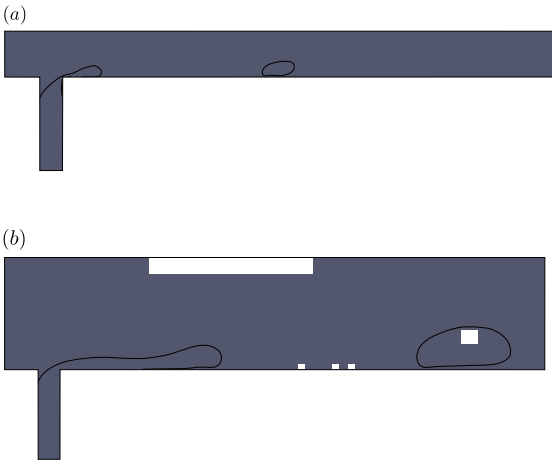


Figure 5.28: Comparison of drop size for P-flow; $v_c/v_d \approx 90.91, \sigma = 0.0415\text{N/m}$. (a) $H/D = 2$ (b) $H/D = 10$.

The influence of elasticity on drop size was also investigated. The elasticity of the dispersed phase was raised by increasing the relaxation time, λ. For the Newtonian case, $\lambda = 0$. As shown in Figs. 5.29 and 5.30, increasing the elasticity of the dispersed phase resulted in an increase in size of droplet formed in the main channel. We also noticed a stronger dependence of droplet size on elasticity as the average velocity of the continuous phase, v_c, is increased. A different behavior was observed for the case when $H = 500\mu$m - a minimal reduction in drop size was seen as elasticity increased.

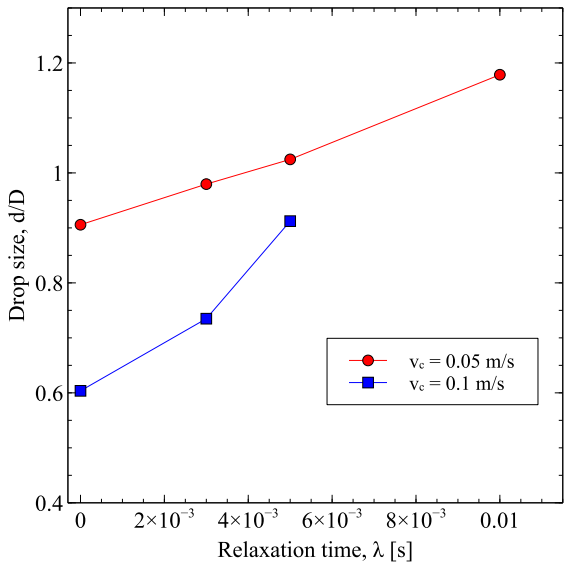


Figure 5.29: Effect of elasticity on drop size for P-flow ($H = 100\mu$m).

The droplet generation process is also different as elasticity of dispersed phase is increased. This behavior is clearly delineated in Fig. 5.30. We compare the droplet formation process for a Newtonian dispersed phase and a viscoelastic dispersed phase. The droplet pinch-off is seen to occur further downstream for the Newtonian case than the viscoelastic case.

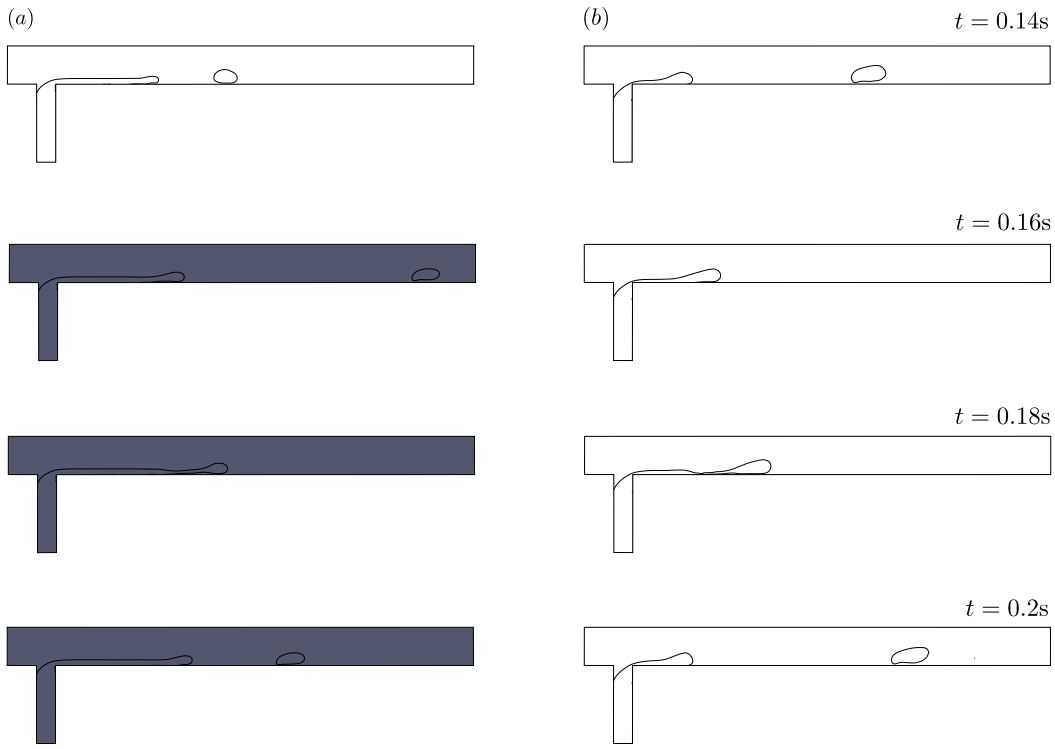


Figure 5.30: Effect of elasticity on droplet formation; $v_c/v_d = 90.91$. (a) Dispersed phase is Newtonian ($De_c = 0$) (b) Dispersed phase is viscoelastic ($De_c = 15$).

To obtain a clearer understanding on the role elasticity plays on final drop size, we examine the polymer tensile stress, τ_{yy}, just before breakup. This is clearly depicted in Fig. 5.31. The highest value of τ_{yy} is achieved in the neck region. The major role is to slow down the drop detachment process. This case is analogous to the visco-elasto-capillary thinning and breakup behavior of a polymeric thread where it has been shown that the detachment of the thread was delayed as a result of increased tensile stress within the neck neighborhood [108]. This gives the drop extra time to expand from taking in fluid through the neck.

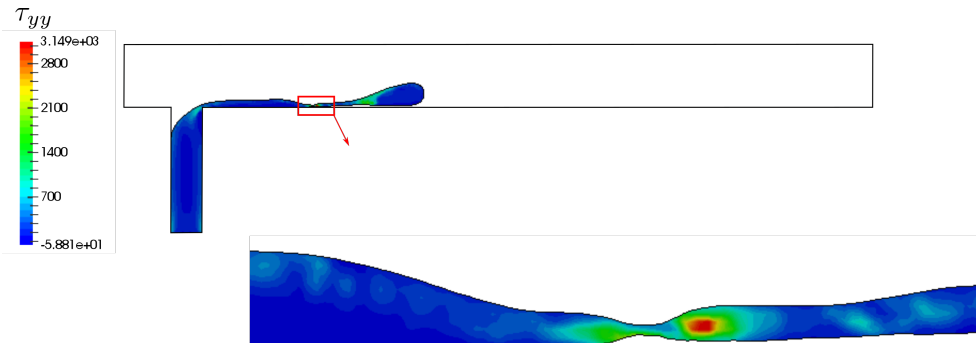


Figure 5.31: Polymeric tensile stress τ_{yy} just before drop detachment ($t = 0.18s$).

5.8 Conclusions

In this work, the influence of imposed channel flow, channel height, interfacial tension and elasticity on drop detachment and size was studied in a microfluidic T-junction. The rheology of the viscoelastic fluid was modeled using the Giesekus model.

A direct relationship was found between the capillary number and drop size in both P-flow and C-flow. In particular, for a given wall shear rate, the size of droplets generated in P-flow was found to be larger than that in C-flow. The interfacial tension was found to have a strong effect on the droplet formation mechanisms. Within the range of parameters used in this study, both jetting and dripping was seen. Reducing interfacial tension resulted in a decrease in drop size which is expected since reducing interfacial tension results in a higher Ca. However, when it is reduced beyond a

critical value satellite droplets are formed at high shear rate. The existence of satellite droplets results in increasing the polydispersity of droplets and in cases where they merge with primary droplets reduce the mixing precision in applications. The monodispersity of droplets was found to be strongly influenced by interfacial tension. Also, its effect on the two types of flows considered in this study - P-flow and C-flow - was different. For a given constant interfacial tension, C-flow was found to produce more uniformly-sized droplets than P-flow with coefficients of variation less than 4.7%. The effect of elasticity was insignificant until above a critical continuous phase flow rate where a minimal reduction in drop size was seen as elasticity increased for $H = 500\mu$m. With a reduced channel height to $H = 100\mu$m, we observed that increasing elasticity resulted in an increase in drop size. This effect became more pronounced as the average velocity of the continuous phase increased - for a fixed De_c, a higher v_c produces larger droplets.

All simulation cases were scheduled to end only after five or more drops had detached from the pore channel, after which the average is taken and reported. Hence the run time for each case was not the same. For example, the run time for the case with $v_c = 0.25$m/s is smaller than the run time for the case $v_c = 0.05$m/s. This holds for both solvers. In particular, using the solver `viscoelasticInterFoam` on 8 processors, it took about 3 and a half days for the case $v_c = 0.25$m/s and 5 days for the case $v_c = 0.05$m/s. On the other hand, `clsVeInterDymFoam` on 5 processors spent about 3 days on the case $v_c = 0.25$m/s to complete and about 5 days on the case $v_c = 0.05$m/s. A

comparison of the convergence plots for pressure and stress fields between the solvers, `viscoelasticInterFoam` and `clsVeInterDymFoam` was also made and is shown in Appendix G. The initial residuals of `viscoelasticInterFoam` for the pressure and stress fields was found to be lower than that of `clsVeInterDymFoam`.

Chapter 6

Summary and Future Work

6.1 Summary

The aims of this thesis were to: (1) develop an improved two-phase flow solver for viscoelastic-Newtonian fluid systems; (2) test the improved solver on a series of test problems; and (3) use the improved solver to investigate the formation and detachment of viscoelastic drops in a T-shaped microchannel.

This work began by presenting the conservation laws together with the stress constitutive equation. A description of the finite volume discretization of these partial differential equations was outlined. For a more accurate calculation of the curvature at the material interface, the Volume of Fluid method was coupled with the level

set method. In addition, fourth-order least squares method was employed for the discretization of the gradient term in the reinitialization equation of the level set function. A description of the adaptive mesh refinement procedure was also given.

A new solver, `clsVeInterDymFoam`, has been developed based on the code, `viscoelasticInterFoam` of Favero [26] and the 2D-planar and axisymmetric AMR libraries of Baniabedalruhman [27]. The robustness and accuracy of `clsVeInterDymFoam` was investigated using the following benchmark cases: Two dimensional drop under static conditions; Drop deformation in simple shear flow for the cases where the drop is Newtonian and the continuous phase is viscoelastic, and vice versa and Rise of a Newtonian drop in a Giesekus fluid. The method was further validated by comparing simulation predictions of drop formation in a T-junction to results from experimental studies. The study revealed that `clsVeInterDymFoam` is superior to `viscoelasticInterFoam` in the static drop problem analysis. On enabling the dynamic mesh functionality of `clsVeInterDymFoam` for the drop deformation in shear flow and rising drop problem, a significant saving in grid size and computation time was observed for the cases with dynamic mesh in comparison to similar test cases ran with a static mesh. In addition, a good qualitative agreement was found for the drop formation in a T-junction case between numerical results and their experimental counterpart.

Finally, the formation and detachment of viscoelastic drops in a Newtonian matrix in

a T-junction was investigated. In particular, the role of imposed flow type, channel height, wall shear rate, interfacial tension and elasticity on formation process of viscoelastic droplets in a Newtonian fluid was critically examined. For both P-flow and C-flow, an increase in the capillary number resulted in a decrease in drop size. For a given wall shear rate, we found that the size of droplets generated in P-flow was found to be larger than that in C-flow. The effect of elasticity was insignificant until above a critical continuous phase flow rate where a minimal reduction in drop size was seen as elasticity increased. On reducing the channel height, an increase in elasticity was accompanied by an increase in drop size. This effect became more pronounced as the average velocity of the continuous phase increased - for a fixed De_c, a higher $\boldsymbol{v}_c$ produces larger droplets.

6.2 Future work

For a more accurate reproduction of droplet formation process in a T-shaped microchannel, it would be useful to conduct the numerical simulations in three-dimensions even though this would demand more CPU time.

The results obtained with `clsVeInterDymFoam` and `viscoelasticInterDymFoam` were different: the detachment behavior and the size of droplet obtained in all simulation cases were not all the same for both solvers e.g. jetting behavior for

`viscoelasticInterFoam` still remained jetting with `clsVeInterDymFoam` but dripping behavior seen in `viscoelasticInterFoam` for large $\boldsymbol{v}_c$ changed when ran with `clsVeInterDymFoam`. The reason for these differences is not known and further research is needed to investigate the rationale behind these differences.

References

[1] Li, X.-B.; Li, F.-C.; Kinoshita, H.; Oishi, M.; Oshima, M. *Earth & Space* **2010**.

[2] Gopala, V. R.; van Wachem, B. G. *Chemical Engineering Journal* **2008**, *141*(1), 204–221.

[3] Jahanbakhsh, E.; Panahi, R.; Seif, M. *International Journal of Numerical Methods for Heat & Fluid Flow* **2007**, *17*(4), 384–404.

[4] Panahi, R.; Jahanbakhsh, E.; Seif, M. S. *Applied Ocean Research* **2006**, *28*(3), 171–181.

[5] Unverdi, S. O.; Tryggvason, G. *Journal of computational physics* **1992**, *100*(1), 25–37.

[6] Sussman, M.; Puckett, E. G. *Journal of Computational Physics* **2000**, *162*(2), 301–337.

[7] Peng, D.; Merriman, B.; Osher, S.; Zhao, H.; Kang, M. *Journal of computational physics* **1999**, *155*(2), 410–438.

[8] Harlow, F. H.; Welch, J. E.; others. *Physics of fluids* **1965**, *8*(12), 2182.

[9] Tome, M. F.; McKee, S. *Journal of Computational Physics* **1994**, *110*(1), 171–186.

[10] Boris, J. P.; Book, D. L. *Journal of computational physics* **1973**, *11*(1), 38–69.

[11] Hirt, C. W.; Nichols, B. D. *Journal of computational physics* **1981**, *39*(1), 201–225.

[12] Jasak, H.; Weller, H. *Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine* **1995**.

[13] Youngs, D. L. *Numerical methods for fluid dynamics* **1982**, *24*(2), 273–285.

[14] Scardovelli, R.; Zaleski, S. *Annual review of fluid mechanics* **1999**, *31*(1), 567–603.

[15] Afkhami, S.; Bussmann, M. *International Journal for Numerical Methods in Fluids* **2008**, *57*(4), 453–472.

[16] Cummins, S. J.; Francois, M. M.; Kothe, D. B. *Computers & structures* **2005**, *83*(6), 425–434.

[17] Afkhami, S.; Leshansky, A.; Renardy, Y. *Physics of Fluids (1994-present)* **2011**, *23*(2), 022002.

[18] Fluent, A. *Ansys Inc* **2009**.

[19] 4.10 and star-ccm+ documentation, 2009. Version, S.-C.

[20] Published by, ansys cfx, release 11.0, december, 2006, ansys. CFX-Manual, A. **2006**.

[21] ESI-US, R. *Huntsville, AL* **2010**.

[22] Palacios, F.; Colonno, M. R.; Aranake, A. C.; Campos, A.; Copeland, S. R.; Economon, T. D.; Lonkar, A. K.; Lukaczyk, T. W.; Taylor, T. W.; Alonso, J. J. *AIAA Paper* **2013**, *287*, 2013.

[23] Popinet, S. *Journal of Computational Physics* **2003**, *190*(2), 572–600.

[24] Richards, K.; Senecal, P.; Pomraning, E. *Convergent Science, Inc., Middleton, WI* **2012**.

[25] Weller, H. G.; Tabor, G.; Jasak, H.; Fureby, C. *Computers in physics* **1998**, *12*(6), 620–631.

[26] Favero, J.; Secchi, A. R.; Cardozo, N.; Jasak, H. *Computers & chemical engineering* **2010**, *34*(12), 1984–1993.

[27] Baniabedalruhman, A. *Dynamic meshing around fluid-fluid interfaces with applications to droplet tracking in contraction geometries* PhD thesis, Michigan Technological University, **2015**.

[28] Shariff, K.; Moser, R. D. *Journal of Computational Physics* **1998**, *145*(2), 471–488.

[29] Kravchenko, A.; Moin, P.; Moser, R. *Journal of Computational Physics* **1996**, *127*(2), 412–423.

[30] Berger, M. J.; Oliger, J. *Journal of computational Physics* **1984**, *53*(3), 484–512.

[31] Berger, M. J. *SIAM Journal on Scientific and Statistical Computing* **1986**, *7*(3), 904–916.

[32] Berger, M. J.; Colella, P. *Journal of computational Physics* **1989**, *82*(1), 64–84.

[33] van der Holst, B.; Keppens, R. *Journal of computational physics* **2007**, *226*(1), 925–946.

[34] DeZeeuw, D.; Powell, K. G. *Journal of Computational Physics* **1993**, *104*(1), 56–68.

[35] Coirier, W. J. *An adaptively-refined, Cartesian, cell-based scheme for the Euler and Navier-Stokes equations* PhD thesis, Citeseer, **1994**.

[36] Coirier, W. J.; Powell, K. G. **1993**.

[37] Coirier, W. J.; Powell, K. G. *AIAA journal* **1996**, *34*(5), 938–945.

[38] Berger, M. J.; Leveque, R. J. *AIAA paper* **1989**, *1930*, 1989.

[39] Bank, R. E. *Elliptic problem solvers.(A 83-14076 03-59) New York, Academic Press, 1981,* **1981**, pages 1–16.

[40] Babuvška, I.; Rheinboldt, W. C. *SIAM Journal on Numerical Analysis* **1978**, *15*(4), 736–754.

[41] Sherman, A.; Saeger, M. *Advances in Computer Methods for Partial Differential Equations (3rd ed.), VI, Intl. Assoc. for Math. and Comp. in Simulation, New Brunswick, New Jersey* **1981**, page 88.

[42] Davis, S. F.; Flaherty, J. E. *SIAM Journal on Scientific and Statistical Computing* **1982**, *3*(1), 6–27.

[43] Dwyer, H.; Kee, R.; Sanders, B. *AIAA Journal* **1980**, *18*(10), 1205–1212.

[44] Winkler, K. Numerical procedure for the calculation of nonsteady spherical shock fronts with radiation Technical report, Max-Planck-Institut fuer Physik und Astrophysik, **1976**.

[45] Harten, A.; Hyman, J. M.; Lax, P. D.; Keyfitz, B. *Communications on pure and applied mathematics* **1976**, *29*(3), 297–322.

[46] Popinet, S. *Journal of Computational Physics* **2009**, *228*(16), 5838–5866.

[47] Vincent, S.; Caltagirone, J.-P. *Journal of Computational Physics* **2000**, *163*(1), 172–215.

[48] Vladisavljević, G. T.; Kobayashi, I.; Nakajima, M. *Microfluidics and nanofluidics* **2012**, *13*(1), 151–178.

[49] der Graaf, S. V.; Steegmans, M. L. J.; der Sman, R. G. M. V.; Schroën, C. G. P. H.; Boom, R. M. *Colloids and Surfaces A: Physicochemical and Engineering Aspects* **2005**, *266*, 106–116.

[50] Herstellen von Öl-in wasser-emulsionen mit mikroporösen membranen. Schröder, V.

[51] Manga, M. S.; Cayre, O. J.; Williams, R. A.; Biggs, S.; York, D. W. *Soft Matter* **2012**, *8*(5), 1532–1538.

[52] Zhu, J.; Barrow, D. *Journal of membrane science* **2005**, *261*(1), 136–144.

[53] Link, D. R.; Anna, S. L.; Weitz, D. A.; Stone, H. A. *Physical review letters* **2004**, *92*(054503), 337–341.

[54] Vladisavljević, G. T.; Shimizu, M.; Nakashima, T. *Journal of membrane science* **2006**, *284*(1), 373–383.

[55] Schadler, V.; Windhab, E. J. *Desalination* **2006**, *189*, 130–135.

[56] Li, X.; Li, Q.; Gong, F. L.; Lei, J. D.; Zhao, X.; Ma, G. H.; Su, Z. G. *Journal of Non-Newtonian Fluid Mechanics* **2015**, *476*, 30–39.

[57] Fuchigami, T.; Toki, M.; Nakanishi, K. *Journal of Sol-Gel Science and Technology* **2000**, *19*, 337–341.

[58] Kobayashi, I.; Yasuno, M.; Iwamoto, S.; Shono, A.; Satoh, K.; Nakajima, M. *Colloids and Surfaces A: Physicochemical and Engineering Aspects* **2002**, *207*, 185–196.

[59] der Graaf, S. V.; Schroën, C. G. P. H.; der Sman, R. G. M. V.; Boom, R. M. *Journal of Non-Newtonian Fluid Mechanics* **2004**, *277*, 456–463.

[60] Williams, R. A.; Peng, S. J.; Wheeler, D. A.; Morley, N. C.; Taylor, D.; Whalley, M.; Houldsworth, D. W. *Chemical Engineering Research and Design* **1998**, *76*, 902–910.

[61] Hong, J. S.; Cooper-White, J. *Korea-Australia Rheology Journal* **2009**, *21*(4), 269–280.

[62] Arratia, P. E.; Cramer, L.; Gollub, J. P.; Durian, D. J. *New Journal of Physics* **2009**, *11*(11), 115006.

[63] Steinhaus, B.; Shen, A. Q.; Sureshkumar, R. *Physics of Fluids (1994-present)* **2007**, *19*(7), 073103.

[64] Husny, J.; Cooper-White, J. J. *Journal of Non-Newtonian Fluid Mechanics* **2006**, *137*, 121–136.

[65] Nisisako, T.; Torii, T.; Higuchi, T. *Chemical Engineering Journal* **2004**, *101*, 23–29.

[66] Davidson, M. R.; Cooper-White, J. J. *Applied mathematical modelling* **2006**, *30*(11), 1392–1405.

[67] Davidson, M. R.; Harvie, D. J.; Cooper-White, J. J. *Korea-Australia Rheology Journal* **2006**, *18*(2), 41–49.

[68] Zhou, C.; Yue, P.; James, F. J. *Physics of Fluids (1994-present)* **2006**, *18*(9), 092105.

[69] Dupret, F.; Marchal, J.; Crochet, M. *Journal of non-newtonian fluid mechanics* **1985**, *18*(2), 173–186.

[70] Joseph, D.; Saut, J. *Journal of Non-Newtonian Fluid Mechanics* **1986**, *20*, 117–141.

[71] Owens, R. G.; Phillips, T. N. *Computational rheology*, Vol. 14; World Scientific, 2002.

[72] Tomé, M.; Mangiavacchi, N.; Cuminato, J.; Castelo, A.; McKee, S. *Journal of Non-Newtonian Fluid Mechanics* **2002**, *106*(2), 61–106.

[73] Baaijens, F. P. *Journal of Non-Newtonian Fluid Mechanics* **1998**, *79*(2), 361–385.

[74] Guénette, R.; Fortin, M. *Journal of non-newtonian fluid mechanics* **1995**, *60*(1), 27–52.

[75] King, R. C.; Apelian, M. R.; Armstrong, R. C.; Brown, R. A. *Journal of Non-Newtonian Fluid Mechanics* **1988**, *29*, 147–216.

[76] Xue, S.-C.; Phan-Thien, N.; Tanner, R. *Journal of Non-Newtonian Fluid Mechanics* **1995**, *59*(2), 191–213.

[77] Phillips, T.; Williams, A. *Journal of Non-Newtonian Fluid Mechanics* **1999**, *87*(2), 215–246.

[78] Shonibare, O.; Feigl, K.; Tanner, F. *ILASS Americas 27th Annual Conference on Liquid Atomization and Spray Systems* **2015**.

[79] Perera, M.; Walters, K. *Journal of Non-Newtonian Fluid Mechanics* **1977**, *2*(1), 49–81.

[80] Rajagopalan, D.; Armstrong, R. C.; Brown, R. A. *Journal of Non-Newtonian Fluid Mechanics* **1990**, *36*, 159–192.

[81] Oliveira, P.; Pinho, F. T. d.; Pinto, G. *Journal of Non-Newtonian Fluid Mechanics* **1998**, *79*(1), 1–43.

[82] Patankar, S. V.; Spalding, D. B. *International journal of heat and mass transfer* **1972**, *15*(10), 1787–1806.

[83] Rhie, C.; Chow, W. *AIAA journal* **1983**, *21*(11), 1525–1532.

[84] Jasak, H.; Weller, H. G. Interface tracking capabilities of the inter-gamma differencing scheme Technical report, Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine, **1995**.

[85] Sussman, M.; Smereka, P.; Osher, S. *Journal of Computational physics* **1994**, *114*(1), 146–159.

[86] Albadawi, A.; Donoghue, D.; Robinson, A.; Murray, D.; Delaure, Y. *International Journal of Multiphase Flow* **2013**, *53*, 11–28.

[87] Guénette, R.; Fortin, M. *Journal of non-newtonian fluid mechanics* **1995**, *60*(1), 27–52.

[88] Favero, J. L.; Secchi, A. R.; Cardozo, N. S. M.; Jasak, H. *Computers and chemical engineering* **2010**, *116*, 1984–1993.

[89] Issa, R. I. *Journal of computational physics* **1986**, *62*(1), 40–65.

[90] Jasak, H. *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows, 1996* PhD thesis, Ph. D. Thesis, University of London Imperial College, **1996**.

[91] Weller, H. *OpenCFD Ltd., Report TR/HGW/04* **2008**.

[92] Popinet, S.; Zaleski, S. *International Journal for Numerical Methods in Fluids* **1999**, *30*(6), 775–793.

[93] Meier, M.; Yadigaroglu, G.; Smith, B. L. *European Journal of Mechanics-B/Fluids* **2002**, *21*(1), 61–73.

[94] Rudman, M. *International journal for numerical methods in fluids* **1997**, *24*(7), 671–691.

[95] Shirani, E.; Ashgriz, N.; Mostaghimi, J. *Journal of Computational Physics* **2005**, *203*(1), 154–175.

[96] Torres, D.; Brackbill, J. *Journal of Computational Physics* **2000**, *165*(2), 620–644.

[97] Li, X.-B.; Li, F.-C.; Kinoshita, H.; Oishi, M.; Oshima, M. *Microfluidics and Nanofluidics* **2015**, *18*(5-6), 1007–1021.

[98] Kwak, J. C. *Polymer-surfactant systems*, Vol. 77; CRC Press, 1998.

[99] Craster, R.; Matar, O.; Papageorgiou, D. *Physics of Fluids (1994-present)* **2002**, *14*(4), 1364–1376.

[100] Kwak, S.; Pozrikidis, C. *International journal of multiphase flow* **2001**, *27*(1), 1–37.

[101] Timmermans, M.-L. E.; Lister, J. R. *Journal of Fluid Mechanics* **2002**, *459*, 289–306.

[102] Van der Graaf, S.; Nisisako, T.; Schroen, C.; Van Der Sman, R.; Boom, R. *Langmuir* **2006**, *22*(9), 4144–4152.

[103] Feigl, K.; Tanner, F. X.; Holzapfel, S.; Windhab, E. J. *Chemical Engineering Science* **2014**, *116*, 327–382.

[104] Bird, R. B.; Armstrong, R. C.; Hassager, O. *Dynamics of polymeric liquids, Vol. 1: Fluid mechanics.;* 1987.

[105] Stern, F.; Wilson, R. V.; Coleman, H. W.; Paterson, E. G. *Journal of fluids engineering* **2001**, *123*, 793–802.

[106] Xue, S.-C.; Tanner, R.; Phan-Thien, N. *Computer methods in applied mechanics and engineering* **1999**, *180*(3), 305–331.

[107] Eggers, J. *ZAMM - Journal of Applied Mathematics and Mechanics/Zeitschrift fr Angewandte Mathematik und Mechanik* **2005**, *85*, 400–410.

[108] Li, J.; Fontelos, M. A. *Physics of Fluids (1994-present)* **2003**, *15*(4), 922–937.

[109] Hafez, M.; Oshima, K.; others. *Computational fluid dynamics review 1995;* World Scientific, 1995.

[110] Haworth, D.; Jansen, K. *Computers & fluids* **2000**, *29*(5), 493–524.

[111] Habla, F.; Obermeier, A.; Hinrichsen, O. *Journal of Non-Newtonian Fluid Mechanics* **2013**, *199*, 70–79.

[112] Frigo, M.; Johnson, S. G. In *Proceedings of the IEEE*, Vol. 93, page 216, 2005.

[113] Spagnolie, S. *Complex Fluids in Biological Systems: Experiment, Theory, and Computation;* Springer, 2014.

[114] Doi, Y.; Gray, F. M.; Buchholz, F. L.; Graham, A. T.; Striegel, A.; Yau, W. W.; Kirkland, J. J.; Bly, D. D.; Gordon Jr, M. J.; Archer, R. D.

[115] Barnes, H. A.; Hutton, J. F.; Walters, K. *An introduction to rheology*, Vol. 3; Elsevier, 1989.

[116] Boger, D. *Journal of Non-Newtonian Fluid Mechanics* **1977**, *3*(1), 87–91.

[117] Boger, D. V.; Nguyen, H. *Polymer Engineering & Science* **1978**, *18*(13), 1037–1043.

[118] White, J.; Metzner, A. *Journal of Applied Polymer Science* **1963**, *7*(5), 1867–1889.

[119] Giesekus, H. *Rheologica Acta* **1966**, *5*(3), 239–252.

[120] Larson, R. *Journal of Rheology (1978-present)* **1984**, *28*(5), 545–571.

[121] Phan-Thien, N. *Journal of Rheology (1978-present)* **1978**, *22*(3), 259–283.

[122] Bird, R. B.; Armstrong, R. C.; Hassager, O.; Curtiss, C. F. *Dynamics of polymeric liquids*, Vol. 1; Wiley New York, 1977.

[123] Brinson, H. F.; Brinson, L. C. *Polymer engineering science and viscoelasticity;* Springer, 2008.

[124] Coleman, B. D.; Noll, W. *Archive for Rational Mechanics and Analysis* **1960**, *6*(1), 355–370.

[125] Layec-Raphalen, M.-N.; Wolff, C. *Journal of Non-Newtonian Fluid Mechanics* **1976**, *1*(2), 159–173.

[126] Green, R. G.; Griskey, R. G. *Transactions of The Society of Rheology (1957-1977)* **1968**, *12*(1), 13–25.

[127] Metzner, A.; Whitlock, M. *Transactions of The Society of Rheology (1957-1977)* **1958**, *2*(1), 239–254.

[128] Hudson, N.; Ferguson, J. *Transactions of The Society of Rheology (1957-1977)* **1976**, *20*(2), 265–286.

[129] Piau, J.; El Kissi, N.; Tremblay, B. *Journal of non-newtonian fluid mechanics* **1988**, *30*(2-3), 197–232.

[130] Groisman, A.; Steinberg, V. *Nature* **2000**, *405*(6782), 53–55.

[131] Berti, S.; Bistagnino, A.; Boffetta, G.; Celani, A.; Musacchio, S. *Physical Review E* **2008**, *77*(5), 055306.

[132] Groisman, A.; Steinberg, V. *New Journal of Physics* **2004**, *6*(1), 29.

[133] Bonn, D.; Ingremeau, F.; Amarouchene, Y.; Kellay, H. *Physical Review E* **2011**, *84*(4), 045301.

[134] Larson, R. G. *Nature* **2000**, *405*(6782), 27–28.

[135] Koshizuka, S.; Oka, Y. *Nuclear science and engineering* **1996**, *123*(3), 421–434.

[136] Brackbill, J. U.; Kothe, D. B.; Zemach, C. *Journal of computational physics* **1992**, *100*, 335–354.

[137] Giesekus, H. *Journal of Non-Newtonian Fluid Mechanics* **1982**, *11*, 69–109.

[138] Marchal, J.; Crochet, M. *Journal of Non-Newtonian Fluid Mechanics* **1986**, *20*, 187–207.

[139] Marchal, J.; Crochet, M. *Journal of Non-Newtonian Fluid Mechanics* **1987**, *26*(1), 77–114.

[140] http://wikipedia.org/. *Wikipedia;* Wikipedia: The Internet, 2012.

[141] Car, R.; Parrinello, M. *Physical Review Letters* **1985**, *55*, 2471.

[142] CPMD (v3.15.1): An *ab initio* Electronic Structure and Molecular Dynamcis Program. The CPMD Consortium, **2011**.

[143] Dovesi, R.; Orlando, R.; Civalleri, B.; Roetti, C.; Saunders, V. R.; Zicovich-Wilson, C. M. *Zieitschrift für Kristallographie* **2005**, *220*, 571.

[144] Crystal 09 user's manual. Dovesi, R.; Saunders, V. R.; Roetti, C.; Orlando, R.; Zicovich-Wilson, C. M.; Pascale, F.; Civalleri, B.; Doll, K.; Harrison, N. M.; Bush, I. J.; D'Arco, P.; Llunell, M.; University of Torino, Italy, **2009**.

[145] Delley, B. *Journal of Chemical Physics* **1990**, *92*, 508.

[146] Delley, B. *Journal of Physical Chemistry* **1996**, *100*, 6107.

[147] Dongarra, J. J.; Bunch, J.; Moler, C.; Stewart, G. W. *LINPACK User's Guide;* Society for Industrial and Applied Mathematics, Philadelphia, PA, 1979.

[148] Dongarra, J. J. *Argonne National Laboratory Report, ANL-80-24* **1980**.

[149] Dongarra, J. J.; Croz, J. D.; Hammarling, S.; Hanson, R. *Association for Computing Machinery Transactions on Mathematical Software* **1988**, *14*, 1.

[150] Dongarra, J. J.; Croz, J. D.; Hammarling, S.; Duff, I. S. *Association for Computing Machinery Transactions on Mathematical Software* **1990**, *16*, 1.

[151] Hess, B.; Kutzner, C.; van der Spoel, D.; Lindahl, E. *Journal of Chemical Theory and Computation* **2008**, *4*, 235.

[152] Gale, J. D. *Journal of Chemical Society, Faraday Transactions* **1997**, *93*, 629.

[153] Gale, J. D. *Philosophical Magazine B* **1996**, *73*, 3.

[154] Falgout, R. D.; Yang, U. M. In *Proceedings of the International Conference on Computational Science - Part III*, ICCS '02, page 632, London, UK, 2002. Springer-Verlag.

[155] Plimpton, S. J. *Journal of Computational Physics* **1995**, *117*, 1.

[156] R: A Language and Environment for Statistical Computing. R Development Core Team.; R Foundation for Statistical Computing, Vienna, Austria, **2011**.

[157] Soler, J. M.; Artacho, E.; Gale, J. D.; García, A.; Junquera, J.; Ordejón, P.; Sánchez-Portal, D. *Journal of Physics: Condensed Matter* **2002**, *14*, 2745.

[158] Ordejó, P.; Drabold, D. A.; Grumbach, M. P.; Martin, R. M. *Physical Review B* **1993**, *48*, 14646.

[159] Rungger, I.; Sanvito, S. *Physical Review B* **2008**, *78*, 035407.

[160] Rocha, A. R. *Theoretical and Computational Aspects of Electronic Transport at the Nanoscale* PhD thesis, University of Dublin, Trinity College, **2007**.

[161] Kresse, G.; Hafner, J. *Physical Review B* **1993**, *47*, 558.

[162] Kresse, G.; Hafner, J. *Physical Review B* **1994**, *49*, 14251.

[163] Schaftenaar, G.; Noordik, J. H. *Journal of Computer-Aided Molecular Design* **2000**, *14*, 123.

[164] Humphrey, W.; Dalke, A.; Schulten, K. *Journal of Molecular Graphics* **1996**, *14*, 33.

[165] Kolkaji, A. *Molecular Graphics Modelling* **1999**, *17*, 176.

[166] Anderson, E.; Bai, Z.; Bischof, C.; Blackford, S.; Demmel, J.; Dongarra, J. J.; Croz, J. D.; Greenbaum, A.; Hammarling, S.; McKenney, A.; Sorensen, D. *LAPACK Users' Guide;* Society for Industrial and Applied Mathematics: Philadelphia, PA, 3 ed., 1999.

Appendix A

Letter for Fig. 2.2

NATURE PUBLISHING GROUP LICENSE TERMS AND CONDITIONS

Mar 27, 2017

This Agreement between Olabanji Shonibare ("You") and Nature Publishing Group ("Nature Publishing Group") consists of your license details and the terms and conditions provided by Nature Publishing Group and Copyright Clearance Center.

License Number	4077221173500
License date	Mar 27, 2017
Licensed Content Publisher	Nature Publishing Group
Licensed Content Publication	Nature
Licensed Content Title	Fluid dynamics: Turbulence without inertia
Licensed Content Author	Ronald G. Larson
Licensed Content Date	May 4, 2000
Licensed Content Volume	405
Licensed Content Issue	6782
Type of Use	reuse in a dissertation / thesis
Requestor type	academic/educational
Format	electronic
Portion	figures/tables/illustrations
Number of figures/tables/illustrations	1
Figures	Figure 1
Author of this NPG article	no
Your reference number	
Title of your thesis / dissertation	Numerical Simulation of Viscoelastic Multiphase Flows Using an Improved Two-phase Flow Solver
Expected completion date	Apr 2017
Estimated size (number of pages)	254
Requestor Location	Olabanji Shonibare 2003 Woodmar Dr. Apt. F HOUGHTON, MI 49931 United States Attn: Olabanji Shonibare
Billing Type	Invoice
Billing Address	Olabanji Shonibare 2003 Woodmar Dr. Apt. F HOUGHTON, MI 49931 United States Attn: Olabanji Shonibare
Total	0.00 USD

Terms and Conditions

Terms and Conditions for Permissions

Nature Publishing Group hereby grants you a non-exclusive license to reproduce this material for this purpose, and for no other use,subject to the conditions below:

1. NPG warrants that it has, to the best of its knowledge, the rights to license reuse of this material. However, you should ensure that the material you are requesting is original to Nature Publishing Group and does not carry the copyright of another entity (as credited in the published version). If the credit line on any part of the material you have requested indicates that it was reprinted or adapted by NPG with permission from another source, then you should also seek permission from that source to reuse the material.

2. Permission granted free of charge for material in print is also usually granted for any electronic version of that work, provided that the material is incidental to the work as a whole and that the electronic version is essentially equivalent to, or substitutes for, the print version.Where print permission has been granted for a fee, separate permission must be obtained for any additional, electronic re-use (unless, as in the case of a full paper, this has already been accounted for during your initial request in the calculation of a print run).NB: In all cases, web-based use of full-text articles must be authorized separately through the 'Use on a Web Site' option when requesting permission.

3. Permission granted for a first edition does not apply to second and subsequent editions and for editions in other languages (except for signatories to the STM Permissions Guidelines, or where the first edition permission was granted for free).

4. Nature Publishing Group's permission must be acknowledged next to the figure, table or abstract in print. In electronic form, this acknowledgement must be visible at the same time as the figure/table/abstract, and must be hyperlinked to the journal's homepage.

5. The credit line should read:
   Reprinted by permission from Macmillan Publishers Ltd: [JOURNAL NAME] (reference citation), copyright (year of publication)
   For AOP papers, the credit line should read:
   Reprinted by permission from Macmillan Publishers Ltd: [JOURNAL NAME], advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM].XXXXX)

   Note: For republication from the *British Journal of Cancer,* the following credit lines apply.
   Reprinted by permission from Macmillan Publishers Ltd on behalf of Cancer Research UK: [JOURNAL NAME] (reference citation), copyright (year of publication)For AOP papers, the credit line should read:
   Reprinted by permission from Macmillan Publishers Ltd on behalf of Cancer Research UK: [JOURNAL NAME], advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM].XXXXX)

6. Adaptations of single figures do not require NPG approval. However, the adaptation should be credited as follows:

   Adapted by permission from Macmillan Publishers Ltd: [JOURNAL NAME] (reference citation), copyright (year of publication)

   Note: For adaptation from the *British Journal of Cancer,* the following credit line applies.
   Adapted by permission from Macmillan Publishers Ltd on behalf of Cancer Research UK: [JOURNAL NAME] (reference citation), copyright (year of publication)

7. Translations of 401 words up to a whole article require NPG approval. Please visit http://www.macmillanmedicalcommunications.com for more information.Translations of up to a 400 words do not require NPG approval. The translation should be credited as follows:

   Translated by permission from Macmillan Publishers Ltd: [JOURNAL NAME] (reference

citation), copyright (year of publication).

Note: For translation from the *British Journal of Cancer,* the following credit line applies.
Translated by permission from Macmillan Publishers Ltd on behalf of Cancer Research UK: [JOURNAL NAME] (reference citation), copyright (year of publication)

We are certain that all parties will benefit from this agreement and wish you the best in the use of this material. Thank you.
Special Terms:
v1.1

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.

Appendix B

Letter for Figs. 2.3, 2.6

Copyright Clearance Center

Confirmation Number: 11634419
Order Date: 03/27/2017

Customer Information

Customer: Olabanji Shonibare
Account Number: 3001070756
Organization: Olabanji Shonibare
Email: oyshonib@mtu.edu
Phone: +1 (906) 369-3977
Payment Method: Invoice

This is not an invoice

Order Details

An introduction to rheology

Billing Status: **N/A**

Order detail ID:	70370853
ISBN:	978-0-444-87469-6
Publication Type:	Book
Publisher:	ELSEVIER
Author/Editor:	Barnes, H. A. ; Hutton, J. F. ; Walters, Kenneth

Permission Status: **Granted**

Permission type: Republish or display content
Type of use: Thesis/Dissertation
Order License Id: 4077231258457

Requestor type	Academic institution
Format	Electronic
Portion	chart/graph/table/figure
Number of charts/graphs/tables/figures	2
Title or numeric reference of the portion(s)	Chapter 4, Figure 4.7. Chapter 6, Figure 6.3
Title of the article or chapter the portion is from	N/A
Editor of portion(s)	N/A
Author of portion(s)	N/A
Volume of serial or monograph	N/A
Page range of portion	61, 100
Publication date of portion	1989
Rights for	Main product
Duration of use	Life of current edition
Creation of copies for the disabled	no
With minor editing privileges	no
For distribution to	Worldwide
In the following language(s)	Original language of publication
With incidental promotional use	no

Lifetime unit quantity of new product	Up to 499
Made available in the following markets	Education
The requesting person/organization	Olabanji Shonibare
Order reference number	
Author/Editor	Olabanji Shonibare
The standard identifier of New Work	NSVMF254
Title of New Work	Numerical Simulation of Viscoelastic Multiphase Flows Using an Improved Two-phase Flow Solver
Publisher of New Work	Michigan Technological University
Expected publication date	Apr 2017
Estimated size (pages)	254

Note: This item was invoiced separately through our **RightsLink service.** More info **$ 0.00**

Total order items: 1 **Order Total: $0.00**

Appendix C

Letter for Fig. 2.5

AIP PUBLISHING LLC LICENSE TERMS AND CONDITIONS

Mar 28, 2017

This Agreement between Olabanji Shonibare ("You") and AIP Publishing LLC ("AIP Publishing LLC") consists of your license details and the terms and conditions provided by AIP Publishing LLC and Copyright Clearance Center.

License Number	4077680435480
License date	Mar 28, 2017
Licensed Content Publisher	AIP Publishing LLC
Licensed Content Publication	Journal of Rheology
Licensed Content Title	Correlation and Molecular Interpretation of Data Obtained in Elongational Flow
Licensed Content Author	N. E. Hudson,J. Ferguson
Licensed Content Date	Jul 1, 1976
Licensed Content Volume	20
Licensed Content Issue	2
Type of Use	Thesis/Dissertation
Requestor type	Student
Format	Electronic
Portion	Figure/Table
Number of figures/tables	1
Title of your thesis / dissertation	Numerical Simulation of Viscoelastic Multiphase Flows Using an Improved Two-phase Flow Solver
Expected completion date	Apr 2017
Estimated size (number of pages)	254
Requestor Location	Olabanji Shonibare 2003 Woodmar Dr. Apt. F HOUGHTON, MI 49931 United States Attn: Olabanji Shonibare
Billing Type	Invoice
Billing Address	Olabanji Shonibare 2003 Woodmar Dr. Apt. F HOUGHTON, MI 49931 United States Attn: Olabanji Shonibare
Total	0.00 USD
Terms and Conditions	

The Society of Rheology -- Terms and Conditions: Permissions Uses

The Society of Rheology ("SOR") hereby grants to you the non-exclusive right and license to use and/or distribute the Material according to the use specified in your order, on a one-time basis, for the specified term, with a maximum distribution equal to the number that you have ordered. Any links or other content accompanying the Material are not the subject of this license.

1. You agree to include the following copyright and permission notice with the reproduction of the Material:"Reprinted with permission from [FULL CITATION]. Copyright [PUBLICATION YEAR], The Society of Rheology." For an article, the copyright and permission notice must be printed on the first page of the article or book chapter. For photographs, covers, or tables, the copyright and permission notice may appear with the Material, in a footnote, or in the reference list.
2. If you have licensed reuse of a figure, photograph, cover, or table, it is your responsibility to ensure that the material is original to SOR and does not contain the copyright of another entity, and that the copyright notice of the figure, photograph, cover, or table does not indicate that it was reprinted by SOR, with permission, from another source. Under no circumstances does SOR, purport or intend to grant permission to reuse material to which it does not hold copyright.
3. You may not alter or modify the Material in any manner. You may translate the Material into another language only if you have licensed translation rights. You may not use the Material for promotional purposes. SOR reserves all rights not specifically granted herein.
4. The foregoing license shall not take effect unless and until SOR or its agent, Copyright Clearance Center, receives the Payment in accordance with Copyright Clearance Center Billing and Payment Terms and Conditions, which are incorporated herein by reference.
5. SOR or the Copyright Clearance Center may, within two business days of granting this license, revoke the license for any reason whatsoever, with a full refund payable to you. Should you violate the terms of this license at any time, SOR, The Society of Rheology, or Copyright Clearance Center may revoke the license with no refund to you. Notice of such revocation will be made using the contact information provided by you. Failure to receive such notice will not nullify the revocation.
6. SOR makes no representations or warranties with respect to the Material. You agree to indemnify and hold harmless SOR, The Society of Rheology, and their officers, directors, employees or agents from and against any and all claims arising out of your use of the Material other than as specifically authorized herein.
7. The permission granted herein is personal to you and is not transferable or assignable without the prior written permission of SOR. This license may not be amended except in a writing signed by the party to be charged.
8. If purchase orders, acknowledgments or check endorsements are issued on any forms containing terms and conditions which are inconsistent with these provisions, such inconsistent terms and conditions shall be of no force and effect. This document, including the CCC Billing and Payment Terms and Conditions, shall be the entire agreement between the parties relating to the subject matter hereof.

This Agreement shall be governed by and construed in accordance with the laws of the State of New York. Both parties hereby submit to the jurisdiction of the courts of New York County for purposes of resolving any disputes that may arise hereunder.

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.

Appendix D

Letter for Fig. 2.7

AIP PUBLISHING LLC LICENSE TERMS AND CONDITIONS

Mar 28, 2017

This Agreement between Olabanji Shonibare ("You") and AIP Publishing LLC ("AIP Publishing LLC") consists of your license details and the terms and conditions provided by AIP Publishing LLC and Copyright Clearance Center.

License Number	4077680229490
License date	Mar 28, 2017
Licensed Content Publisher	AIP Publishing LLC
Licensed Content Publication	Journal of Rheology
Licensed Content Title	Flow Behavior of Concentrated (Dilatant) Suspensions
Licensed Content Author	A. B. Metzner,Malcolm Whitlock
Licensed Content Date	Dec 31, 1969
Licensed Content Volume	2
Licensed Content Issue	1
Type of Use	Thesis/Dissertation
Requestor type	Student
Format	Electronic
Portion	Figure/Table
Number of figures/tables	1
Title of your thesis / dissertation	Numerical Simulation of Viscoelastic Multiphase Flows Using an Improved Two-phase Flow Solver
Expected completion date	Apr 2017
Estimated size (number of pages)	254
Requestor Location	Olabanji Shonibare 2003 Woodmar Dr. Apt. F HOUGHTON, MI 49931 United States Attn: Olabanji Shonibare
Billing Type	Invoice
Billing Address	Olabanji Shonibare 2003 Woodmar Dr. Apt. F HOUGHTON, MI 49931 United States Attn: Olabanji Shonibare
Total	0.00 USD

Terms and Conditions

The Society of Rheology -- Terms and Conditions: Permissions Uses

The Society of Rheology ("SOR") hereby grants to you the non-exclusive right and license to use

and/or distribute the Material according to the use specified in your order, on a one-time basis, for the specified term, with a maximum distribution equal to the number that you have ordered. Any links or other content accompanying the Material are not the subject of this license.

1. You agree to include the following copyright and permission notice with the reproduction of the Material:"Reprinted with permission from [FULL CITATION]. Copyright [PUBLICATION YEAR], The Society of Rheology." For an article, the copyright and permission notice must be printed on the first page of the article or book chapter. For photographs, covers, or tables, the copyright and permission notice may appear with the Material, in a footnote, or in the reference list.
2. If you have licensed reuse of a figure, photograph, cover, or table, it is your responsibility to ensure that the material is original to SOR and does not contain the copyright of another entity, and that the copyright notice of the figure, photograph, cover, or table does not indicate that it was reprinted by SOR, with permission, from another source. Under no circumstances does SOR, purport or intend to grant permission to reuse material to which it does not hold copyright.
3. You may not alter or modify the Material in any manner. You may translate the Material into another language only if you have licensed translation rights. You may not use the Material for promotional purposes. SOR reserves all rights not specifically granted herein.
4. The foregoing license shall not take effect unless and until SOR or its agent, Copyright Clearance Center, receives the Payment in accordance with Copyright Clearance Center Billing and Payment Terms and Conditions, which are incorporated herein by reference.
5. SOR or the Copyright Clearance Center may, within two business days of granting this license, revoke the license for any reason whatsoever, with a full refund payable to you. Should you violate the terms of this license at any time, SOR, The Society of Rheology, or Copyright Clearance Center may revoke the license with no refund to you. Notice of such revocation will be made using the contact information provided by you. Failure to receive such notice will not nullify the revocation.
6. SOR makes no representations or warranties with respect to the Material. You agree to indemnify and hold harmless SOR, The Society of Rheology, and their officers, directors, employees or agents from and against any and all claims arising out of your use of the Material other than as specifically authorized herein.
7. The permission granted herein is personal to you and is not transferable or assignable without the prior written permission of SOR. This license may not be amended except in a writing signed by the party to be charged.
8. If purchase orders, acknowledgments or check endorsements are issued on any forms containing terms and conditions which are inconsistent with these provisions, such inconsistent terms and conditions shall be of no force and effect. This document, including the CCC Billing and Payment Terms and Conditions, shall be the entire agreement between the parties relating to the subject matter hereof.

This Agreement shall be governed by and construed in accordance with the laws of the State of New York. Both parties hereby submit to the jurisdiction of the courts of New York County for purposes of resolving any disputes that may arise hereunder.

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.

Appendix E

Letter for Figs. 4.29, 4.30

Michigan Tech

Olabanji Shonibare <oyshonib@mtu.edu>

Seeking permission to use a figure from a paper

PERMISSIONS <permissions@asce.org> Tue, Sep 27, 2016 at 3:08 PM
To: Olabanji Shonibare <oyshonib@mtu.edu>

Dear Olabanji,

Permission is granted for you to reuse on figure from “Formation of Uniform Plugs and Monodispersed Droplets for Viscoelastic Fluid Flow in Microchannels” for your thesis.

A full credit line must be added to the material being reprinted. For reuse in non-ASCE publications, add the words "With permission from ASCE" to your source citation. For Intranet posting, add the following additional notice: "This material may be downloaded for personal use only. Any other use requires prior permission of the American Society of Civil Engineers.”

Regards,

Joann

Joann Fogleson

Manager, Product and Subscription Services

American Society of Civil Engineers

1801 Alexander Bell Drive

Reston, VA 20191

PERMISSIONS@asce.org

703-295-6112

E-mail: jfogleson@asce.org

Internet: www.asce.org/pubs | www.ascelibrary.org | http://ascelibrary.org/page/rightsrequests

personal use only. Any other use requires prior permission of the American Society of Civil Engineers."

Each license is unique, covering only the terms and conditions specified in it. Even if you have obtained a license for certain ASCE copyrighted content, you will need to obtain another license if you plan to reuse that content outside the terms of the existing license. For example: If you already have a license to reuse a figure in a journal, you still need a new license to use the same figure in a magazine. You need separate license for each edition.

Authors may post the final draft of their work on open, unrestricted Internet sites or deposit it in an institutional repository when the draft contains a link to the bibliographic record of the published version in the ASCE Library or Civil Engineering Database. "Final draft" means the version submitted to ASCE after peer review and prior to copyediting or other ASCE production activities; it does not include the copyedited version, the page proof, or

a PDF of the published version.

For more information on how an author may reuse their own material, please view: http://ascelibrary.org/page/informationforasceauthorsreusingyourownmaterial

From: Olabanji Shonibare [mailto:oyshonib@mtu.edu]
Sent: Tuesday, September 27, 2016 12:32 PM
To: PERMISSIONS <permissions@asce.org>
Subject: Seeking permission to use a figure from a paper

[Quoted text hidden]

Appendix F

Code Documentation

1. The code below was written to print the front of a filament for each time step.

Listing F.1: filament

```
#!/bin/bash

# prints the evolution of the filament length
# Usage: filament 'startTime' 'endTime' 'timeStep'
#
#
#should be run in caseDir/surfaces


#Usage: full details
#1. edit sampleDict to print xy-plane for alpha
#2. run 'sample'
#3. cd surfaces
#4. run the application, 'filament'
#5. It prints result to a file called 'filamentR'

if [ "$1" == "-h" ]; then
echo "Usage: filament 'startTime' 'endTime' '↩
   timeStep'"
exit 0
```

```
fi


rm -rf filamentR

#Algorithm

#1.accept arguments
startTime=$1
endTime=$2
step=$3


#2. determine the x points to use

#print the first field (x-cordinates) > 0, sort ↩
   and print only unique lines
#xPts: positive x coordinates
cd 0  #any directory is fine for base case
awk ' $1 >= 20e-6  {print $1} ' ↩
   alpha_constantPlane.raw | sort
-g | uniq > xPts

#assign contents of the xPts file into array, xPts
xPts=( `cat "xPts" ` )

cd .. #go back to the surfaceDir


#iterate time directories

#no of time directories
total=$(echo "(($endTime-$startTime)/$step)+1" | ↩
   bc)

t=1
time=$startTime
while [ $t -le $total ]
do
cd $time
```

```
##aim: get x-location of filament front

##search the alpa_constantPlane.raw file,
#if you see a line that satisfies alpha[x] >= 0.5
#    continue to the next xpoint
#else
#    break and report time and the previous x in ↩
   the array


#array index
counter=-1

for x in "${xPts[@]}"
do
counter=$((counter+1))

search=$(awk -v var=$x ' $1==var && $4>=0.5 {print↩
    "true"; exit}'
alpha_constantPlane.raw)

if [[ $search = "true" ]]; then
continue
else
prevCounter=$((counter-1))
echo -e "$time \t ${xPts[$prevCounter]}" >> ../↩
   filamentR
break
##just in case of counter problems, use this
#prevCounter=$((counter-1))
#if [ $prevCounter -lt 0 ]; then
#    echo -e "0.2 \t ${xPts[0]}"
#    exit 0
#else
#    echo -e "0.2 \t ${xPts[$prevCounter]}"
#    exit 0
#fi
fi

done
```

```
cd ..  #move back to surfaceDir

#increment time and remove trailing zeros
time=$(echo "$time $step" | awk '{printf "%f", $1+↩
   $2}' | awk '
sub("\\.*0+$","")')


#update counter
t=$((t+1))
done


echo "done!"
```

2. Modified the stress equation by re-arranging its terms and multiplying by λ to also include the case when the relaxation time, λ equals zero.

Listing F.2: Giesekus.H

```
class Giesekus
:
public viscoelasticLaw
{


//- Construct from components
Giesekus
(
const word& name,
const volScalarField& alpha,
const volVectorField& U,
const surfaceScalarField& phi,
const dictionary& dict
);


// Member Functions
```

```
//- Return the viscoelastic stress tensor
virtual tmp<volSymmTensorField> tau() const
{
return tau_;
}

//- Return the coupling term for the momentum ↩
   equation
virtual tmp<fvVectorMatrix> divTau(volScalarField&↩
    alpha,
volVectorField& U) const;

//- Correct the viscoelastic stress
virtual void correct();
};

tmp<fvVectorMatrix> Giesekus::divTau(↩
   volScalarField& alpha,
volVectorField& U) const
{

volScalarField alpha1f =
min(max(alpha, scalar(0)), scalar(1));

volScalarField etaS =
alpha1f*etaS1_ + (scalar(1) - alpha1f)*etaS2_;

volScalarField etaP =
alpha1f*etaP1_ + (scalar(1) - alpha1f)*etaP2_;

return
(
fvc::div(tau_, "div(tau)")
+ fvm::laplacian( etaS + etaP , U, "laplacian(etaS↩
   +etaP,U)")
+ ( fvc::grad(U) & fvc::grad(etaS) )
- fvc::laplacian(etaP, U, "laplacian(etaP,U)")
);

}
```

```
void Giesekus::correct()
{
// Velocity gradient tensor
volTensorField L = fvc::grad(U());

// Convected derivate term
volTensorField C = tau_ & L;

// Twice the rate of deformation tensor
volSymmTensorField twoD = twoSymm(L);

// Two phase transport properties treatment
volScalarField alpha1f =
min(max(alpha(), scalar(0)), scalar(1));

volScalarField lambda =
alpha1f*lambda1_ + (scalar(1) - alpha1f)*lambda2_;

volScalarField etaP =
alpha1f*etaP1_ + (scalar(1) - alpha1f)*etaP2_;

volScalarField alpha =
alpha1f*alpha1_ + (scalar(1) - alpha1f)*alpha2_;

// Stress transport equation
tmp<fvSymmTensorMatrix> tauEqn
(
/*
Multiplied through by lambda to
avoid divison by zero
*/
lambda *  //OS
(
fvm::ddt(tau_)
+ fvm::div(phi(), tau_)
)
==
etaP * twoD
- fvm::Sp( scalar(1), tau_ )
+ lambda *
(
twoSymm( C )
```

```
- (alpha / etaP) * ( tau_ & tau_)
)
);

tauEqn().relax();
solve(tauEqn);

}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *↩
    * * * * * * * * * * * * * * * * * * * * * //

} // End namespace Foam
```

3. Coupling level-set method with solver:

   † Created a new folder for easy adaptation to the libraries of solver.

   † Source file modified to include the computation of surface tension force using the level set function and also for compatibility with OF-2.3.x.

Listing F.3: UEqn.H

```
/*
surfaceScalarField muEff
(
"muEff",
twoPhaseProperties.muf()
+ fvc::interpolate(rho*turbulence->nut())
);
*/

fvVectorMatrix UEqn
(
fvm::ddt(rho, U)
+ fvm::div(rhoPhi, U)
//- fvm::laplacian(muEff, U)
//- (fvc::grad(U) & fvc::grad(muEff))
```

```
//- fvc::div(muEff*(fvc::interpolate(dev(fvc↩
   ::grad(U))) & mesh.Sf()))
- visco.divTau(alpha, U) //OS - stress ↩
   contribution
);

UEqn.relax();

if (momentumPredictor)
{
solve
(
UEqn
==
fvc::reconstruct
(
(
//  fvc::interpolate(interface.sigmaK())*fvc↩
   ::snGrad(alpha1)
//OS1
sigma*fvc::snGrad(psi)*fvc::interpolate(C)*
fvc::interpolate(delta)
//OS2
- ghf*fvc::snGrad(rho)
- fvc::snGrad(pd)
) * mesh.magSf()
)
);
}
```

Listing F.4: clsVeInterDymFoam.C

```
//OS1
#   include "mappingPsi.H"
#   include "solveLSFunction.H"
#   include "calcNewCurvature.H"
//OS2
/* * * * * * * * * * * * * * * * * * * * * * * * ↩
   * * * * * * * *
* * * * * * */
```

```
Info<< "\nStarting time loop\n" << endl;

while (runTime.run())
{
#       include "readPISOControls.H"
#       include "readTimeControls.H"
#       include "CourantNo.H"
#       include "setDeltaT.H"

runTime++;

Info<< "Time = " << runTime.timeName() << nl ↩
   << endl;

#       include "alphaEqnSubCycle.H"

//OS1
#       include "mappingPsi.H"
#       include "solveLSFunction.H"
#       include "calcNewCurvature.H"
#       include "updateFlux.H"
//OS2


#       include "UEqn.H"

// --- SIMPLE loop
for (int corr=0; corr<nCorr; corr++)
{
#           include "pEqn.H"
}



// Correct stress
visco.correct();

#include "continuityErrs.H"

p = pd + rho*gh;

if (pd.needReference())
```

```
{
p += dimensionedScalar
(
"p",
p.dimensions(),
pRefValue - getRefCellValue(p, pdRefCell)
);
}

strainRate =
alpha*Foam::sqrt(2.0)*mag
(
symm(fvc::grad(U))
)
+
(scalar(1) - alpha)*Foam::sqrt(2.0)*mag
(
symm(fvc::grad(U))
);

runTime.write();

Info<< "ExecutionTime = " << runTime.↩
   elapsedCpuTime() << " s"
<< "  ClockTime = " << runTime.↩
   elapsedClockTime() << " s"
<< nl << endl;
}

Info<< "End\n" << endl;

return(0);
}
```

4. AMR

† The velocity equation:

Listing F.5: UEqn.H

```
fvVectorMatrix UEqn
(
fvm::ddt(rho, U)
+ fvm::div(rhoPhi, U)
//+ turbulence->divDevRhoReff(rho, U)
//the stress contribution
- visco.divTau(alpha1, U) //OS
==
fvOptions(rho, U)
);


UEqn.relax();

fvOptions.constrain(UEqn);

if (pimple.momentumPredictor())
{
solve
(
UEqn
==
fvc::reconstruct
(
(

//mixture.surfaceTensionForce()
fvc::interpolate(interface.sigmaK())*fvc::↩
   snGrad(alpha1) //OS
- ghf*fvc::snGrad(rho)
- fvc::snGrad(p_rgh)
) * mesh.magSf()
)
);

fvOptions.correct(U);
}
```

† The volume fraction equation. Replaced most of the original code in default alphaEqn.H in OF-1.6.ext with that of 2.3.x. for compatibility.

Listing F.6: alphaEqn.H

```
{
word alphaScheme("div(phi,alpha)");
word alpharScheme("div(phirb,alpha)");


//OS1
tmp<fv::ddtScheme<scalar> > ddtAlpha
(
fv::ddtScheme<scalar>::New
(
mesh,
mesh.ddtScheme("ddt(alpha)")
)
);

// Set the off-centering coefficient ↩
   according to ddt scheme
scalar ocCoeff = 0;
if
(
isType<fv::EulerDdtScheme<scalar> >(ddtAlpha↩
   ())
|| isType<fv::localEulerDdtScheme<scalar> >(↩
   ddtAlpha())
)
{
ocCoeff = 0;
}
else if (isType<fv::CrankNicolsonDdtScheme<↩
   scalar> >(ddtAlpha()))
{
if (nAlphaSubCycles > 1)
{
FatalErrorIn(args.executable())
<< "Sub-cycling is not supported "
"with the CrankNicolson ddt scheme"
<< exit(FatalError);
}

ocCoeff =
```

```
refCast<fv::CrankNicolsonDdtScheme<scalar> >(↩
    ddtAlpha()).ocCoeff();
}
else
{
FatalErrorIn(args.executable())
<< "Only Euler and CrankNicolson ddt schemes ↩
    are supported"
<< exit(FatalError);
}

scalar cnCoeff = 1.0/(1.0 + ocCoeff);

// Standard face-flux compression coefficient
surfaceScalarField phic(interface.cAlpha()*↩
    mag(phi/mesh.magSf()));

// Add the optional isotropic compression ↩
    contribution
if (icAlpha > 0)
{
phic *= (1.0 - icAlpha);
phic += (interface.cAlpha()*icAlpha)*fvc::↩
    interpolate(mag(U));
}

// Do not compress interface at non-coupled ↩
    boundary faces
// (inlets, outlets etc.)
forAll(phic.boundaryField(), patchi)
{
fvsPatchScalarField& phicp = phic.↩
    boundaryField()[patchi];

if (!phicp.coupled())
{
phicp == 0;
}
}

tmp<surfaceScalarField> phiCN(phi);
```

```
// Calculate the Crank-Nicolson off-centred ↩
   volumetric flux
if (ocCoeff > 0)
{
phiCN = cnCoeff*phi + (1.0 - cnCoeff)*phi.↩
   oldTime();
}

if (MULESCorr)
{
fvScalarMatrix alpha1Eqn
(
#ifdef LTSSOLVE
fv::localEulerDdtScheme<scalar>(mesh, rDeltaT↩
   .name()).fvmDdt(alpha1)
#else
fv::EulerDdtScheme<scalar>(mesh).fvmDdt(↩
   alpha1)
#endif
+ fv::gaussConvectionScheme<scalar>
(
mesh,
phiCN,
upwind<scalar>(mesh, phiCN)
).fvmDiv(phiCN, alpha1)
);

alpha1Eqn.solve();

Info<< "Phase-1 volume fraction = "
<< alpha1.weightedAverage(mesh.Vsc()).value()
<< "  Min(" << alpha1.name() << ") = " << min↩
   (alpha1).value()
<< "  Max(" << alpha1.name() << ") = " << max↩
   (alpha1).value()
<< endl;

tmp<surfaceScalarField> tphiAlphaUD(alpha1Eqn↩
   .flux());
phiAlpha = tphiAlphaUD();
```

```
if (alphaApplyPrevCorr && tphiAlphaCorr0.↩
   valid())
{
Info<< "Applying the previous iteration ↩
   compression flux" << endl;
#ifdef LTSSOLVE
MULES::LTScorrect(alpha1, phiAlpha, ↩
   tphiAlphaCorr0(), 1, 0);
#else
MULES::correct(alpha1, phiAlpha, ↩
   tphiAlphaCorr0(), 1, 0);
#endif

phiAlpha += tphiAlphaCorr0();
}

// Cache the upwind-flux
tphiAlphaCorr0 = tphiAlphaUD;

alpha2 = 1.0 - alpha1;

interface.correct();
}


for (int aCorr=0; aCorr<nAlphaCorr; aCorr++)
{

surfaceScalarField phir(phic*interface.nHatf↩
   ());

tmp<surfaceScalarField> tphiAlphaUn
(
fvc::flux
(
phi,
alpha1,
alphaScheme
)
+ fvc::flux
(
-fvc::flux(-phir, alpha2, alpharScheme),
```

```
alpha1,
alpharScheme
)
);


if (ocCoeff > 0)
{
tphiAlphaUn =
cnCoeff*tphiAlphaUn + (1.0 - cnCoeff)*↩
   phiAlpha.oldTime();
}

if (MULESCorr)
{
tmp<surfaceScalarField> tphiAlphaCorr(↩
   tphiAlphaUn() - phiAlpha);
volScalarField alpha10("alpha10", alpha1);

#ifdef LTSSOLVE
MULES::LTScorrect(alpha1, tphiAlphaUn(), ↩
   tphiAlphaCorr(), 1, 0);
#else
MULES::correct(alpha1, tphiAlphaUn(), ↩
   tphiAlphaCorr(), 1, 0);
#endif


if (aCorr == 0)
{
phiAlpha += tphiAlphaCorr();
}
else
{
alpha1 = 0.5*alpha1 + 0.5*alpha10;
phiAlpha += 0.5*tphiAlphaCorr();
}
}
else
{
phiAlpha = tphiAlphaUn;
```

```
#ifdef LTSSOLVE
MULES::explicitLTSSolve(alpha1, phi, phiAlpha↩
    , 1, 0);
#else
MULES::explicitSolve(alpha1, phiCN, phiAlpha,↩
     1, 0);
#endif
}

alpha2 = 1.0 - alpha1;


interface.correct();
}

if (alphaApplyPrevCorr && MULESCorr)
{
tphiAlphaCorr0 = phiAlpha - tphiAlphaCorr0;
}

if
(
word(mesh.ddtScheme("ddt(rho,U)"))
== fv::EulerDdtScheme<vector>::typeName
)
{
rhoPhi = phiAlpha*(rho1 - rho2) + phiCN*rho2;
}
else
{
if (ocCoeff > 0)
{
// Computation flux of volume fraction field ↩
   time step
phiAlpha = (phiAlpha - (1.0 - cnCoeff)*↩
   phiAlpha.oldTime())/cnCoeff;
}

//the end of the volume flux time step is ↩
   computed here.
rhoPhi = phiAlpha*(rho1 - rho2) + phi*rho2;
}
```

```
Info<< "Phase-1 volume fraction = "
<< alpha1.weightedAverage(mesh.Vsc()).value()
<< "  Min(" << alpha1.name() << ") = " << min↩
   (alpha1).value()
<< "  Max(" << alpha1.name() << ") = " << max↩
   (alpha1).value()
<< endl;
}
//OS2
```

† Correction of the pimple corrector

Listing F.7: correctPhi.H

```
while (pimple.correctNonOrthogonal())
{
fvScalarMatrix pcorrEqn
(
fvm::laplacian(rAUf, pcorr) == fvc::div(phi) ↩
   divUCorr //OS1
);
//The code above was added for compatibility

pcorrEqn.setReference(pRefCell, pRefValue);
pcorrEqn.solve();

if (pimple.finalNonOrthogonalIter())
{
phi -= pcorrEqn.flux();
}
}
```

5. To account for the contact angle boundary condition, the variable below was created for easy adaptation. This was added in 1.6-ext but wasn't needed in the 2.3.x version.

Listing F.8: viscoelasticInterFoam.C

```
//More details about this variable can be found
//at https://openfoamwiki.net/index.php/↩
   OpenFOAM_guide
// /The_PISO_algorithm_in_OpenFOAM

//OS1
volScalarField rUA
(
IOobject
(
"(1|A(U))",
runTime.timeName(),
mesh
),
mesh,
dimensionedScalar("rUA", dimTime, 1),
zeroGradientFvPatchScalarField::typeName
);
//OS2
//contact angle corrected using default code for ↩
   easy parallelization
```

6. Oldroyd-B Model:

For the Oldroyb-B Model, the changes made were similar to the Giesekus Model

- the stress equation was multiplied by lambda to avoid division by zero.

Listing F.9: Oldroyd-B.C

```
void Oldroyd_B::correct()
{
// Velocity gradient tensor
volTensorField L = fvc::grad(U());

// Convected derivate term
volTensorField C = tau_ & L;

// Twice the rate of deformation tensor
volSymmTensorField twoD = twoSymm(L);
```

```
// Two phase transport properties treatment
volScalarField alpha1f =
min(max(alpha(), scalar(0)), scalar(1));

volScalarField lambda =
alpha1f*lambda1_ + (scalar(1) - alpha1f)*lambda2_;

volScalarField etaP =
alpha1f*etaP1_ + (scalar(1) - alpha1f)*etaP2_;

// Stress transport equation
tmp<fvSymmTensorMatrix> tauEqn
(
lambda* //OS
(
fvm::ddt(tau_)
+ fvm::div(phi(),tau_)
)
==
etaP * twoD
+ lambda * twoSymm( C )
- fvm::Sp(1, tau_ )
);

tauEqn().relax();
solve(tauEqn);

}
```

Appendix G

Some convergence plots

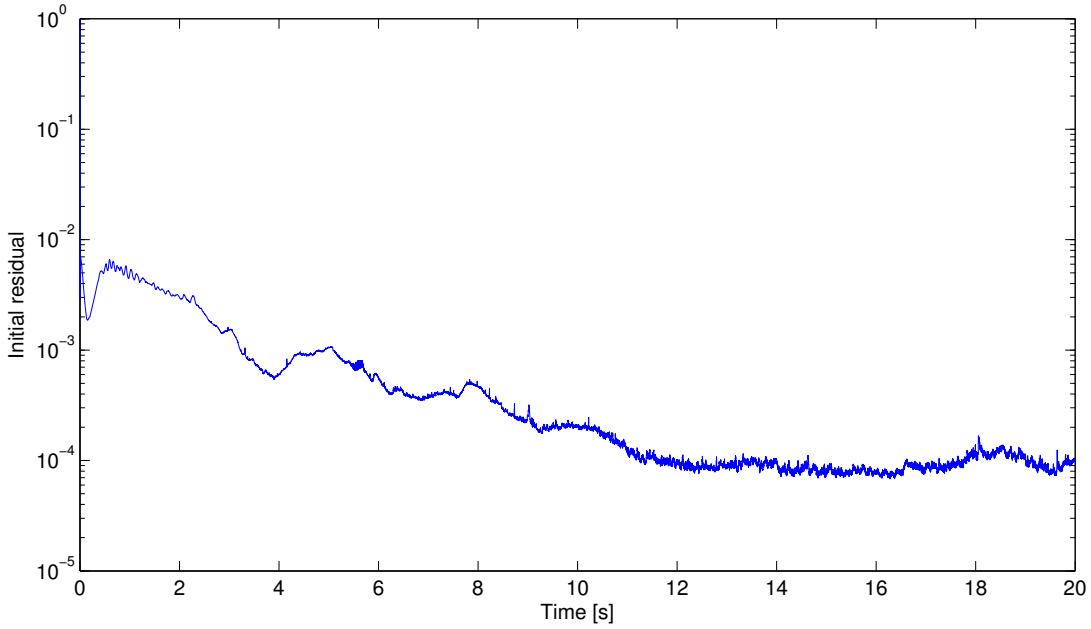


Figure G.1: Initial residual of p_rgh for the drop deformation in shear flow case (mesh1) using `clsVeInterDymFoam`; the velocity at the upper wall is $u = 5$mm/s and at the lower wall is $u = -5$mm/s; the continuous phase is fluid N3 and the dispersed phase is fluid VE1.

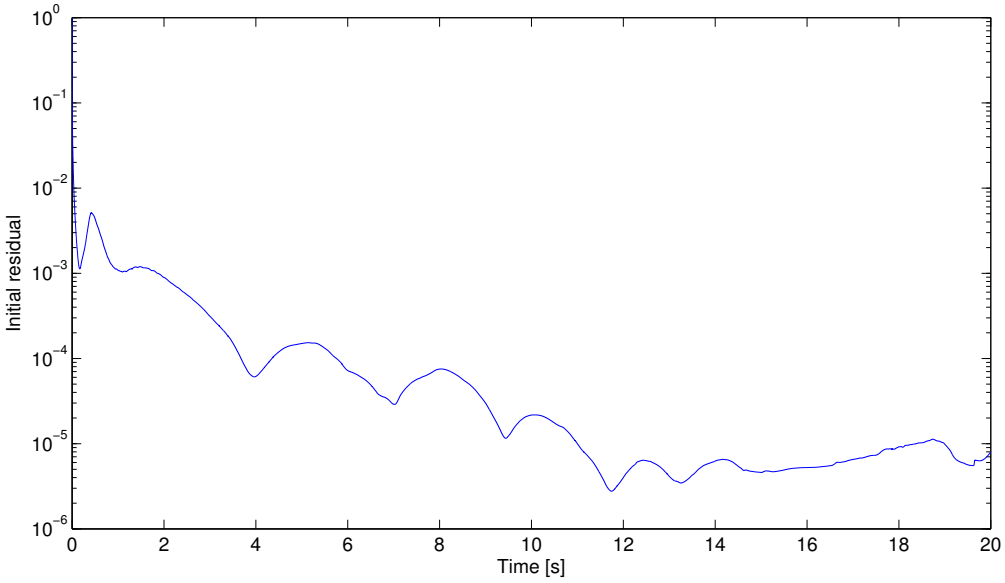


Figure G.2: Initial residual of τ_{xx} for the drop deformation in shear flow case (mesh1) using `clsVeInterDymFoam`; the velocity at the upper wall is $u = 5\text{mm/s}$ and at the lower wall is $u = -5\text{mm/s}$; the continuous phase is fluid N3 and the dispersed phase is fluid VE1.

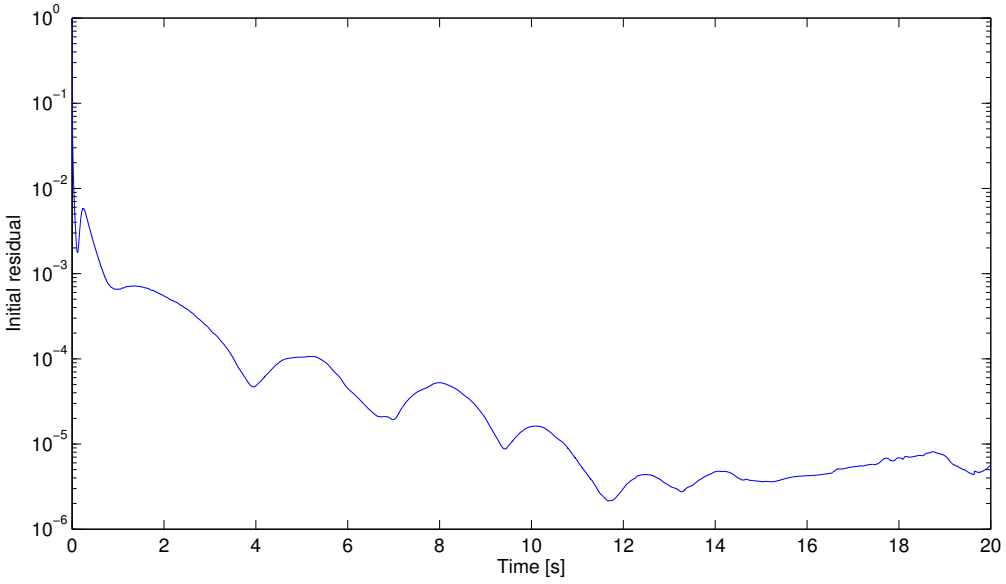


Figure G.3: Initial residual of τ_{xy} for the drop deformation in shear flow case (mesh1) using `clsVeInterDymFoam`; the velocity at the upper wall is $u = 5\text{mm/s}$ and at the lower wall is $u = -5\text{mm/s}$; the continuous phase is fluid N3 and the dispersed phase is fluid VE1.

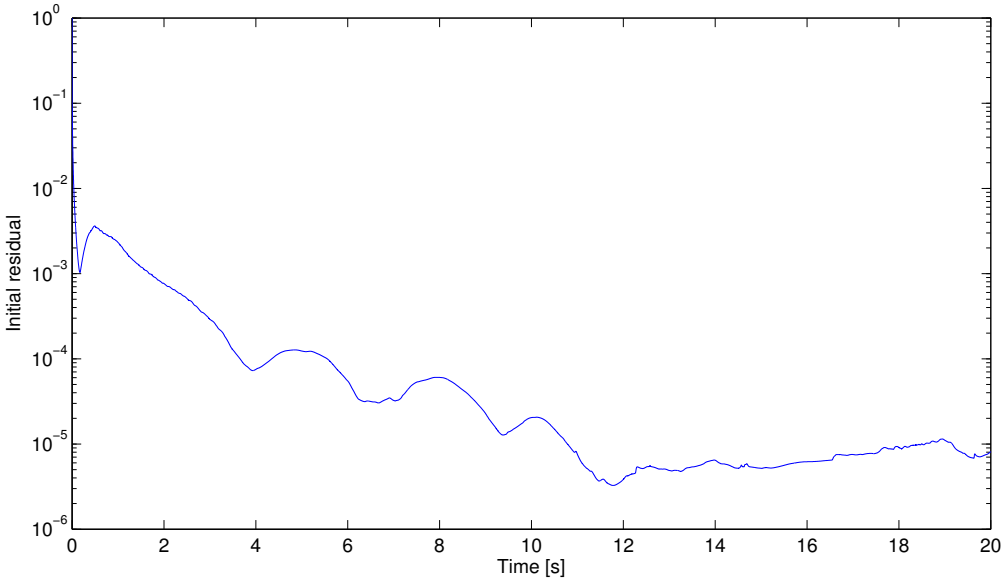


Figure G.4: Initial residual of τ_{yy} for the drop deformation in shear flow case (mesh1) using `clsVeInterDymFoam`; the velocity at the upper wall is $u = 5\text{mm/s}$ and at the lower wall is $u = -5\text{mm/s}$; the continuous phase is fluid N3 and the dispersed phase is fluid VE1.

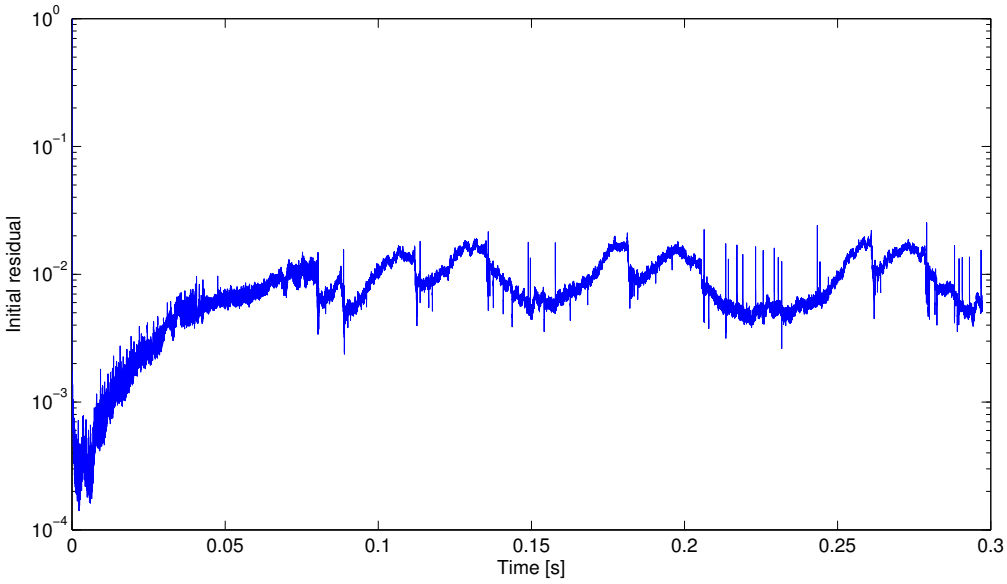


Figure G.5: Initial residual of p_rgh for the P-flow case with dynamic mesh and $H = 500\mu$m using `clsVeInterDymFoam`; the average imposed velocity of the continuous phase and dispersed phase is $v_c = 0.2$m/s and $v_d = 0.0011$m/s respectively; the density of the continuous phase, $\rho_c = 960$kg/m^3, the density of the dispersed phase, $\rho_d = 803.387$kg/m^3, the viscosity of the continuous phase is $\eta_c = 0.106$Pa.s and the solvent and polymeric viscosity of the dispersed phase is $\eta_s = 0.002$Pa.s and $\eta_p = 1.2$Pa.s respectively, the relaxation time of the dispersed phase is $\lambda = 0.005$s and mobility factor of the dispersed phase is $\alpha = 0.05$.

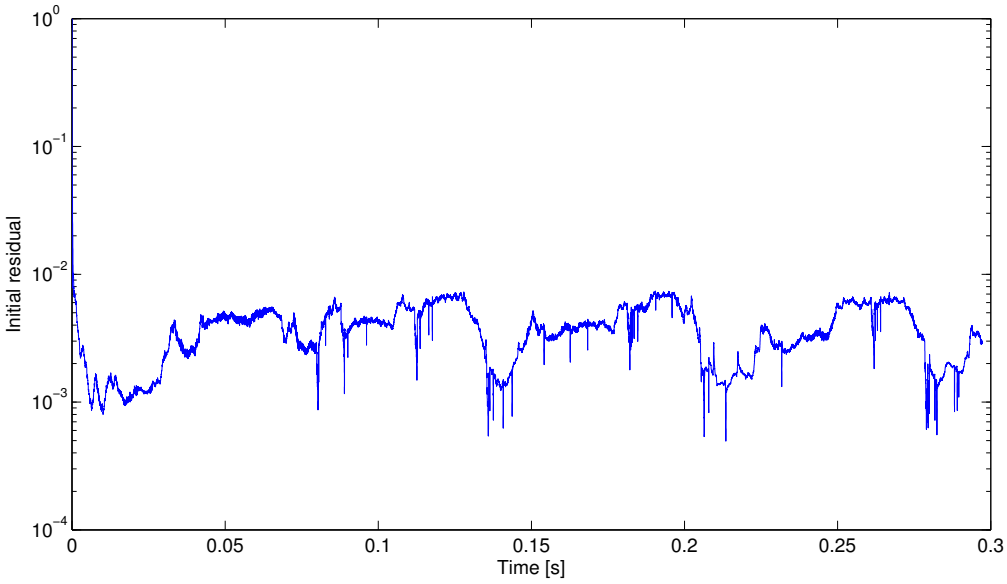


Figure G.6: Initial residual of τ_{xx} for the P-flow case with dynamic mesh and $H = 500\mu$m using `clsVeInterDymFoam`; the average imposed velocity of the continuous phase and dispersed phase is $v_c = 0.2$m/s and $v_d = 0.0011$m/s respectively; the density of the continuous phase, $\rho_c = 960\text{kg}/m^3$, the density of the dispersed phase, $\rho_d = 803.387\text{kg}/m^3$, the viscosity of the continuous phase is $\eta_c = 0.106$Pa.s and the solvent and polymeric viscosity of the dispersed phase is $\eta_s = 0.002$Pa.s and $\eta_p = 1.2$Pa.s respectively, the relaxation time of the dispersed phase is $\lambda = 0.005$s and mobility factor of the dispersed phase is $\alpha = 0.05$.

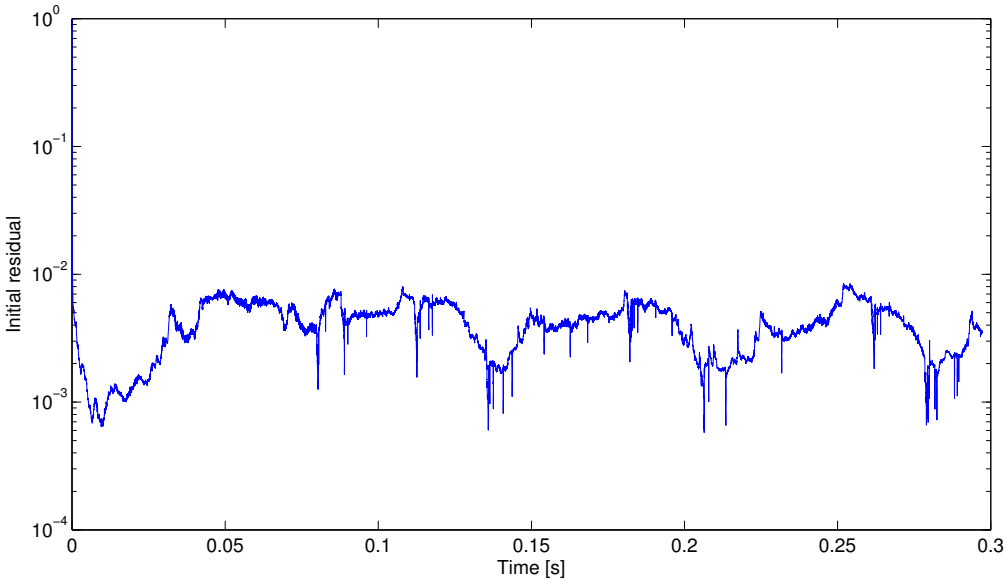


Figure G.7: Initial residual of τ_{xy} for the P-flow case with dynamic mesh and $H = 500\mu$m `clsVeInterDymFoam`; the average imposed velocity of the continuous phase and dispersed phase is $v_c = 0.2$m/s and $v_d = 0.0011$m/s respectively; the density of the continuous phase, $\rho_c = 960$kg/m^3, the density of the dispersed phase, $\rho_d = 803.387$kg/m^3, the viscosity of the continuous phase is $\eta_c = 0.106$Pa.s and the solvent and polymeric viscosity of the dispersed phase is $\eta_s = 0.002$Pa.s and $\eta_p = 1.2$Pa.s respectively, the relaxation time of the dispersed phase is $\lambda = 0.005$s and mobility factor of the dispersed phase is $\alpha = 0.05$.

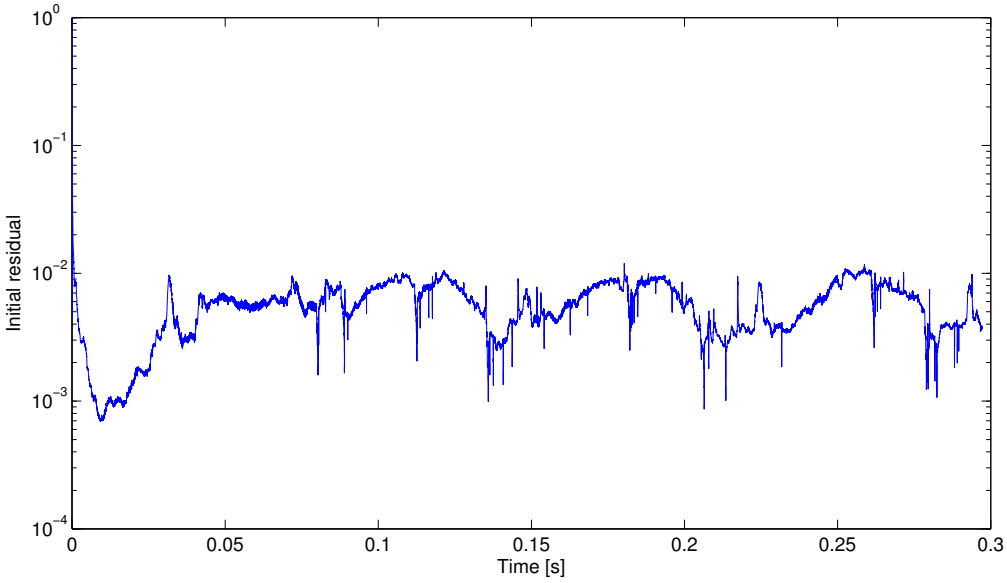
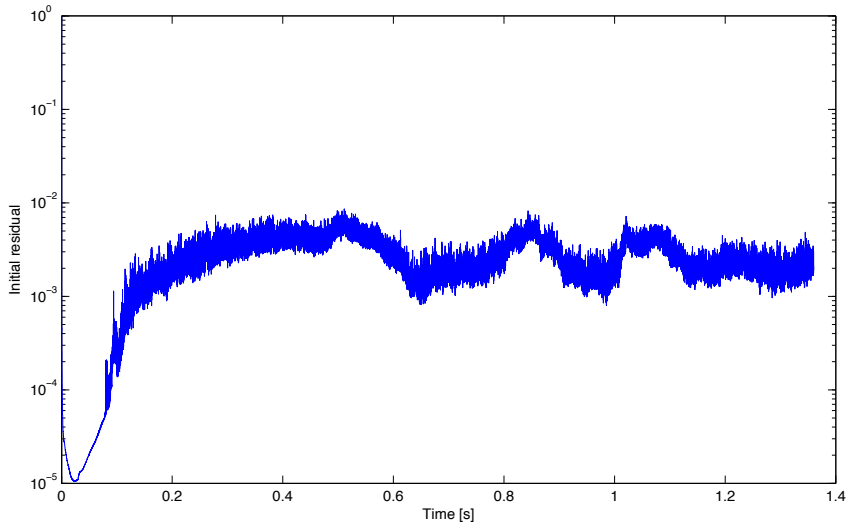
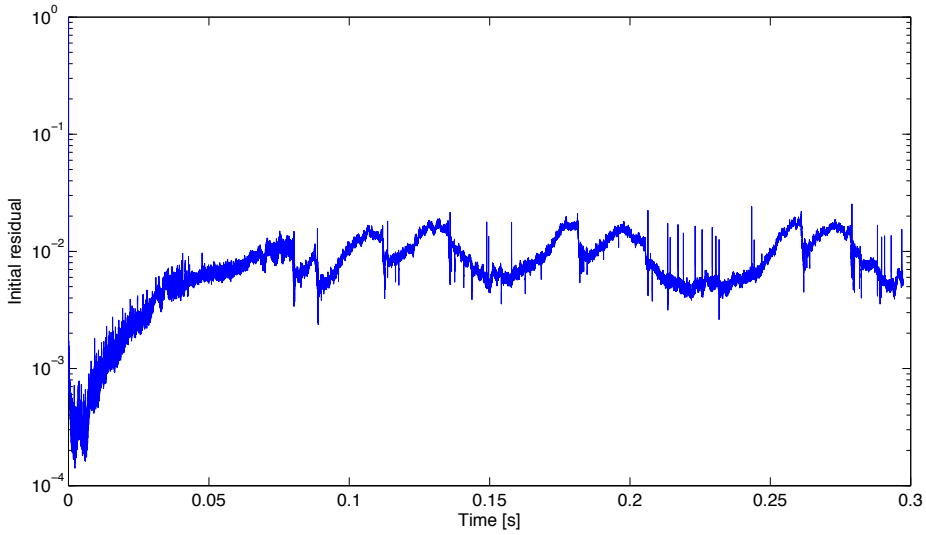


Figure G.8: Initial residual of τ_{yy} for the P-flow case with dynamic mesh and $H = 500\mu$m `clsVeInterDymFoam`; the average imposed velocity of the continuous phase and dispersed phase is $v_c = 0.2$m/s and $v_d = 0.0011$m/s respectively; the density of the continuous phase, $\rho_c = 960$kg/m^3, the density of the dispersed phase, $\rho_d = 803.387$kg/m^3, the viscosity of the continuous phase is $\eta_c = 0.106$Pa.s and the solvent and polymeric viscosity of the dispersed phase is $\eta_s = 0.002$Pa.s and $\eta_p = 1.2$Pa.s respectively, the relaxation time of the dispersed phase is $\lambda = 0.005$s and mobility factor of the dispersed phase is $\alpha = 0.05$.

G.1 Comparison of the convergence plot between `viscoelasticInterFoam` and `clsVeInterDymFoam`

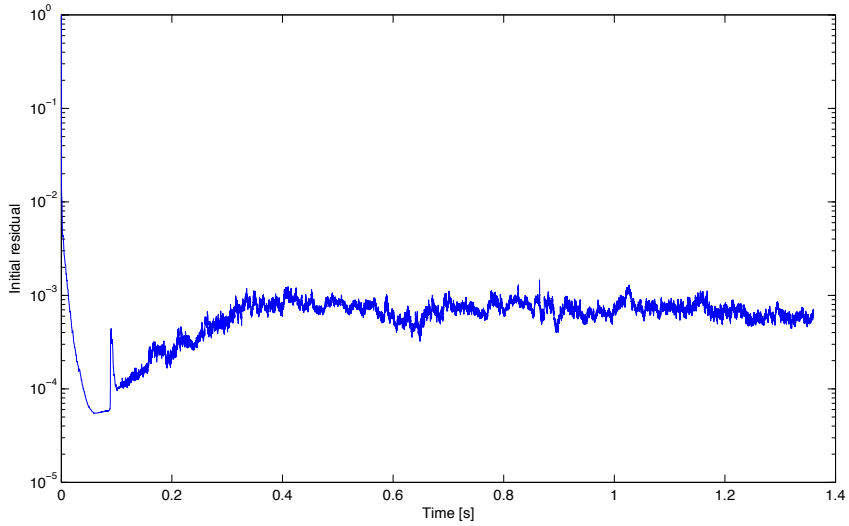


(a)

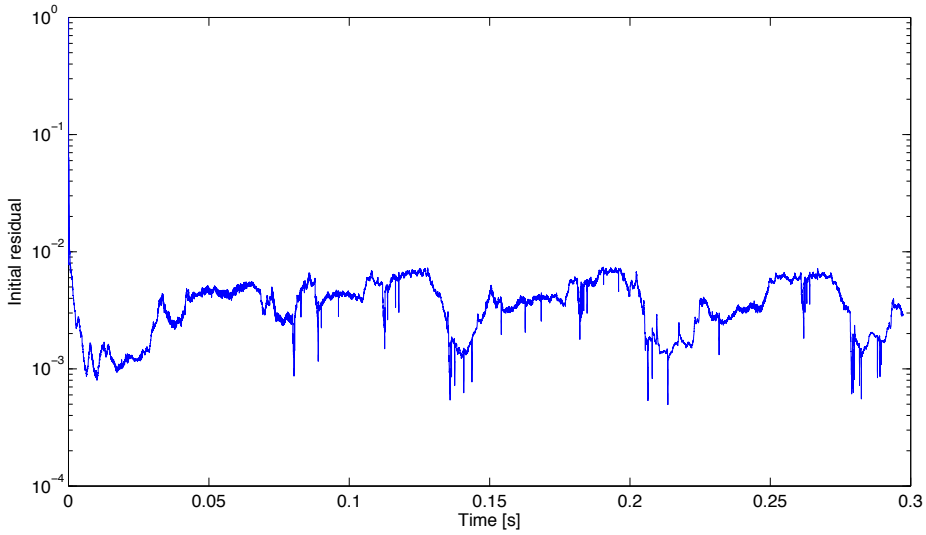


(b)

Figure G.9: Initial residual of p_rgh for the P-flow case with $H = 500\mu$m on (a) static mesh using `viscoelasticInterFoam` and (b) dynamic mesh using `clsVeInterDymFoam`; the average imposed velocity of the continuous phase and dispersed phase is $v_c = 0.2$m/s and $v_d = 0.0011$m/s respectively; the density of the continuous phase, $\rho_c = 960\text{kg}/m^3$, the density of the dispersed phase, $\rho_d = 803.387\text{kg}/m^3$, the viscosity of the continuous phase is $\eta_c = 0.106$Pa.s and the solvent and polymeric viscosity of the dispersed phase is $\eta_s = 0.002$Pa.s and $\eta_p = 1.2$Pa.s respectively, the relaxation time of the dispersed phase is $\lambda = 0.005$s and mobility factor of the dispersed phase is $\alpha = 0.05$.

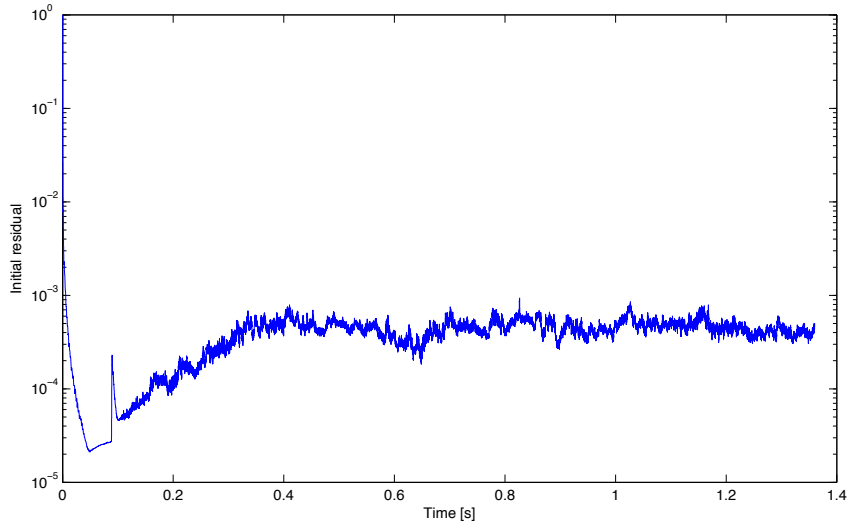


(a)

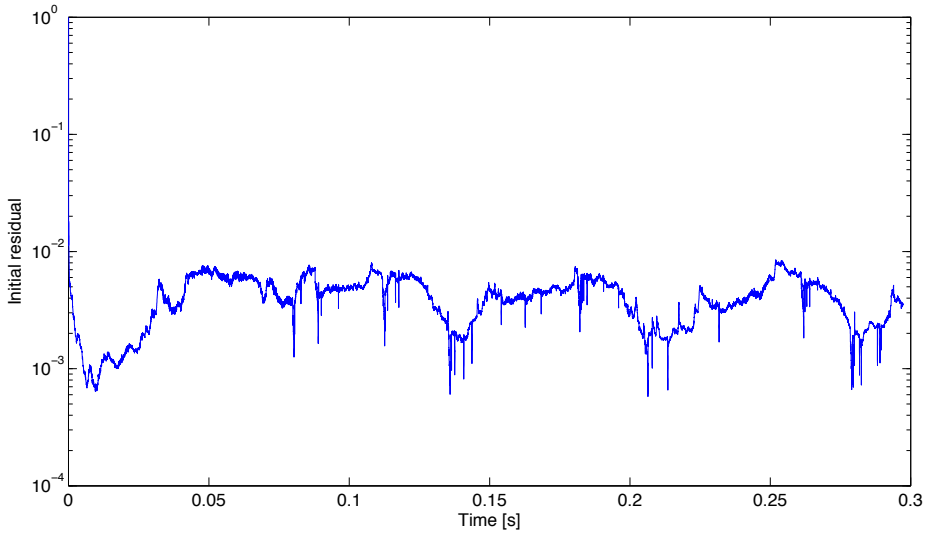


(b)

Figure G.10: Initial residual of τ_{xy} for the P-flow case with $H = 500\mu$m on (a) static mesh using `viscoelasticInterFoam` and (b) dynamic mesh using `clsVeInterDymFoam`; the average imposed velocity of the continuous phase and dispersed phase is $v_c = 0.2$m/s and $v_d = 0.0011$m/s respectively; the density of the continuous phase, $\rho_c = 960\text{kg}/m^3$, the density of the dispersed phase, $\rho_d = 803.387\text{kg}/m^3$, the viscosity of the continuous phase is $\eta_c = 0.106$Pa.s and the solvent and polymeric viscosity of the dispersed phase is $\eta_s = 0.002$Pa.s and $\eta_p = 1.2$Pa.s respectively, the relaxation time of the dispersed phase is $\lambda = 0.005$s and mobility factor of the dispersed phase is $\alpha = 0.05$.

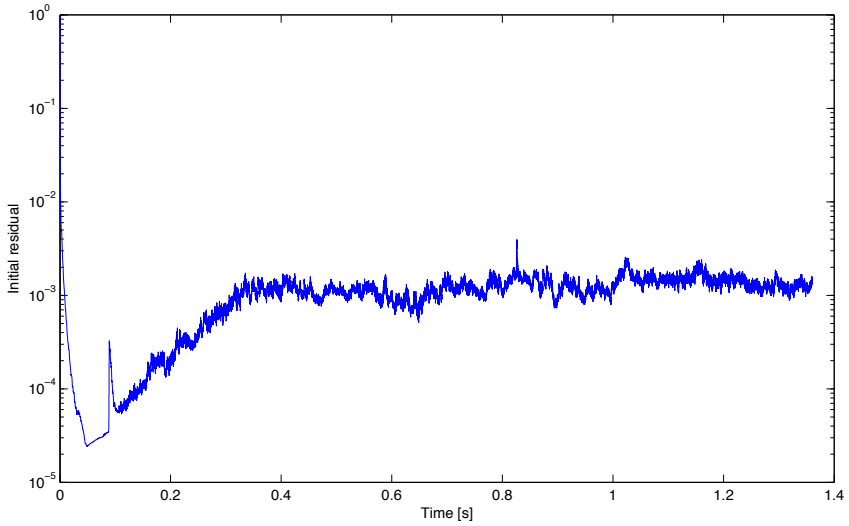


(a)

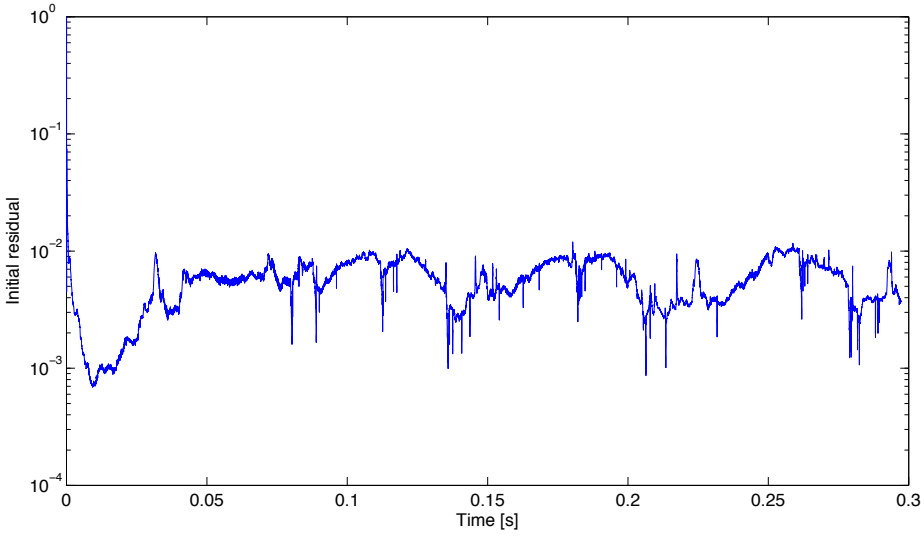


(b)

Figure G.11: Initial residual of τ_{xy} for the P-flow case with $H = 500\mu$m on (a) static mesh using `viscoelasticInterFoam` and (b) dynamic mesh using `clsVeInterDymFoam`; the average imposed velocity of the continuous phase and dispersed phase is $v_c = 0.2$m/s and $v_d = 0.0011$m/s respectively; the density of the continuous phase, $\rho_c = 960$kg/m^3, the density of the dispersed phase, $\rho_d = 803.387$kg/m^3, the viscosity of the continuous phase is $\eta_c = 0.106$Pa.s and the solvent and polymeric viscosity of the dispersed phase is $\eta_s = 0.002$Pa.s and $\eta_p = 1.2$Pa.s respectively, the relaxation time of the dispersed phase is $\lambda = 0.005$s and mobility factor of the dispersed phase is $\alpha = 0.05$.



(a)



(b)

Figure G.12: Initial residual of τ_{yy} for the P-flow case with $H = 500\mu$m on (a) static mesh using `viscoelasticInterFoam` and (b) dynamic mesh using `clsVeInterDymFoam`; the average imposed velocity of the continuous phase and dispersed phase is $v_c = 0.2$m/s and $v_d = 0.0011$m/s respectively; the density of the continuous phase, $\rho_c = 960\text{kg}/m^3$, the density of the dispersed phase, $\rho_d = 803.387\text{kg}/m^3$, the viscosity of the continuous phase is $\eta_c = 0.106$Pa.s and the solvent and polymeric viscosity of the dispersed phase is $\eta_s = 0.002$Pa.s and $\eta_p = 1.2$Pa.s respectively, the relaxation time of the dispersed phase is $\lambda = 0.005$s and mobility factor of the dispersed phase is $\alpha = 0.05$.