The present work was submitted to the Chair for Computational Analysis of Technical Systems.

**Simulation of rising bubbles in viscoelastic fluids**

Master Thesis

Presented by
von Danwitz, Maximilian Hans Wilhelm

Evaluated by
Prof. Marek Behr, Ph.D.
Dipl.-Phys. Philipp Knechtges

Communicated by
Prof. Marek Behr, Ph.D.

Aachen, September 30, 2015
Affidavit:

I hereby declare that I wrote this thesis on my own and without the use of any other than the cited sources and tools and all explanations that I copied directly or in their sense are marked as such, as well as that the thesis has not been available to any audit authority yet.

Aachen, September 30, 2015

Maximilian Hans Wilhelm von Danwitz
# Contents

**List of Figures**

<table>
<thead>
<tr>
<th>List of Figures</th>
<th>I</th>
</tr>
</thead>
</table>

**List of Tables**

<table>
<thead>
<tr>
<th>List of Tables</th>
<th>III</th>
</tr>
</thead>
</table>

1. **Introduction**

2. **Background on two phase flow simulation**
   2.1. Governing equations .................................. 2
   2.2. Initial, boundary, and interface conditions ........ 3
   2.3. Weak form .............................................. 4
   2.4. Deforming spatial domain space-time finite element method 4
   2.5. Interface tracking .................................. 6
   2.6. Elastic mesh update .................................. 7

3. **Adapting the mesh update strategy to rising bubbles**
   3.1. Need for action ...................................... 8
   3.2. Rigid body motion of the computational domain ...... 10
   3.3. Mesh slip along fluid-fluid-interface ................ 11
   3.4. An error adaptive mesh update ....................... 12
      3.4.1. Derivation of weakly incorporated error-induced body forces 13
      3.4.2. First results .................................... 14

4. **Benchmarks with Newtonian fluid**
   4.1. Two-dimensional test case "TC1-Hysing" ............. 19
   4.2. Three-dimensional test case "TC2-Adelsberger" ..... 24

5. **Background on viscoelastic fluid flow simulations**
   5.1. Modeling viscoelastic fluids .......................... 27
   5.2. Log-conformation formulation ......................... 30
   5.3. Finite element implementation ....................... 31

6. **Viscoelastic simulations**
   6.1. Two-dimensional test cases .......................... 32
       6.1.1. Test case "TC3-Zainali" .......................... 32
       6.1.2. Test case "TC4-Vahabi" .......................... 35
       6.1.3. Test case "TC5-Damanik" ........................ 38
       6.1.4. Test case "TC6" .................................. 39
   6.2. Three-dimensional test cases ........................ 44
       6.2.1. Test case "TC7-Pillapakkam" .................... 44
       6.2.2. Test case "TC8" .................................. 46
7. Conclusion

8. Acknowledgments

References

A. Poiseuille flow

B. Two-dimensional test case "TC1-Hysing"

C. Three-dimensional test case "TC2-Adelsberger"

D. Test case "TC3-Zainali"

E. Test case "TC4-Vahabi"

F. Test case "TC5-Damanik"

G. Test case "TC6"

H. Test case "TC7-Pillapakkam"

I. Test case "TC8"
# List of Figures

1. Problem Setup ......................................................... 2
2. Generic geometry for all two-dimensional computations ............ 8
3. Mesh deformation as bubble rises .................................... 9
4. Mesh quality after 3000 time steps .................................. 11
5. Node distribution along the bubble interface ....................... 12
6. Error adaptive mesh update for Poiseuille flow .................... 14
7. Error adaptive mesh update for L-channel flow ..................... 15
8. Error adaptive mesh update for L-channel flow, close-up ......... 16
9. Error adaptive mesh update for L-channel flow, after 20 time steps 17
10. Error adaptive mesh update for L-channel flow, after 100 time steps 18
11. Test case "TC1-Hysing" Hysing et al. (2009): Initial geometry and parameters .................................................. 19
12. Test case "TC1-Hysing", discretization with P1 elements: Rise velocity and circularity ...................................................... 21
13. Test case "TC1-Hysing", discretization with P1 elements: Quasi-steady state bubble shapes ($t = 3$) and volume conservation ......... 22
14. Test case "TC1-Hysing", discretization with P2 elements: Rise velocity and circularity ...................................................... 23
15. Test case "TC1-Hysing", discretization with P2 elements: Quasi-steady state bubble shapes ($t = 3$) and volume conservation ......... 23
16. Test case "TC2-Adelsberger" Adelsberger et al. (2014): Initial geometry and parameters .................................................. 24
17. Test case "TC2-Adelsberger", discretization with P1 and P2 elements: Rise velocity and sphericity .................................................. 26
18. Test case "TC2-Adelsberger": Stable ellipsoidal bubble shape ($t = 3$, $P_2$-simulation) and volume conservation ..................... 27
19. Illustration of viscoelastic fluid models with spring and damper elements ................................................................. 28
20. Test case "TC3-Zainali" Zainali et al. (2013): Initial geometry and parameters in cgs units .................................................. 32
21. Test case "TC3-Zainali": Negative wake behind the bubble observed on the fine mesh at $t = 0.13 \, s$ ................................................. 33
22. Test case "TC3-Zainali": Bubble shapes observed on the different meshes for Oldroyd-B (a,c,e) and Giesekus model, with $\alpha = 0.3$ (b,d,e) ........................ 34
23. Test case "TC3-Zainali": Rise velocity and circularity ..................... 34
24. Test case "TC4-Vahabi" Vahabi and Sadeghy (2015): Initial geometry and parameters in cgs units .................................................. 35
25. Test case "TC4-Vahabi": Rise velocity and circularity ..................... 36
26. Test case "TC4-Vahabi": Comparison with published data ..................... 36
27. Test case "TC4-Vahabi": Influence of the no-slip side wall proximity ....... 37
28. Test case "TC4-Vahabi": Influence of the side wall boundary conditions, $L_x = 8\,cm$ .......................... 38
29. Test case "TC5-Damanik" Damanik et al. (2013): Initial geometry and parameters ........................................... 38
30. Test case "TC5-Damanik": Evolution of bubble shape and $\Psi_{xx}$ ............... 39
31. Test case "TC6": Initial geometry and parameters in cgs units .................. 39
32. Test case "TC6": Rise velocity and circularity .......................... 40
33. Test case "TC6": Evolution of bubble shape and $\Psi_{xx}$ for $\mu_2 = 10.25$ .... 40
34. Test case "TC6": Influence of the bubble viscosity $\mu_2$ .......................... 42
35. Test case "TC6": Bubble shapes and $\Psi_{xx}$ at $t = 0.5\,s$ in the upper row and $t = 1\,s$ in the lower row .......................... 42
36. Test case "TC6": Influence of the Giesekus mobility parameter $\alpha$ .............. 43
37. Test case "TC6": Bubble shapes and $\Psi_{xx}$ at $t = 1\,s$ in the upper row and $t = 2\,s$ in the lower row .......................... 43
38. Test case "TC7-Pillapakkam" Pillapakkam et al. (2007): Initial geometry and parameters in cgs units .......................... 44
39. Test case "TC7-Pillapakkam": Rise velocity and sphericity ....................... 45
40. Test case "TC7-Pillapakkam": Bubble shapes and velocity vectors, $\alpha = 0.0$ 46
41. Test case "TC8": Initial geometry and parameters in cgs units .................. 47
42. Test case "TC8": Rise velocity, sphericity, and normalized vertical bubble length .......................... 48
43. Test case "TC8": Bubble shapes at $t = 2\,s$, labeled with the initial bubble radius ............................................ 48
44. Test case "TC8": Vertical bubble length and shape changes .................... 49
45. Test case "TC8": Rise velocity at $t = 2\,s$ (as approximation of the terminal bubble velocity) as function of the bubble radius and volume ............... 50
46. Test case "TC8": Bubble shapes at $t = 2\,s$ (right) compared to experimentally observed shapes Ohta et al. (2015a) (left) .......................... 51
47. Test case "TC8": Bubble shape at $t = 3\,s$ (right) compared to experimentally observed shape Ohta et al. (2015b) (left) .......................... 52
List of Tables

1. Test case "TC1-Hysing": Mesh characteristics and performance . . . . . . 20
2. Parameters in viscoelastic simulations . . . . . . . . . . . . . . . . . . . . . . . . . . . 32
1. Introduction

A variety of materials can be modeled as a viscoelastic fluid. Frequently, these fluids play an important role in industrial processes as, e.g., in the case of plastics profile extrusion or in medical applications as, e.g., in the case of blood pumps. To learn more about the physics beneath these applications, it is instructive to study specific viscoelastic flow phenomena by means of simple systems. An example of such a system is a bubble rising in a viscoelastic fluid.

A large number of researchers have been investigating the rise of a Newtonian bubble in viscoelastic fluid by means of physical experiments, cf. Amirnia et al. (2013); Liu et al. (1995); Ohta et al. (2015a, b) to mention only a few. Compared to the number of experiments, there have been only a few numerical considerations of the problem so far, e.g., Damanik et al. (2013); Pillapakkam et al. (2007); Vahabi and Sadeghy (2015); Zainali et al. (2013). The numerical investigation of a rising bubble in viscoelastic fluid is also the main subject of this work.

Remarkable features of the flow have been observed experimentally as well as numerically: Firstly, the trailing end of the bubble develops to a cusp-like shape. Secondly, a recirculation zone is observed in the fluid behind the bubble. This is called negative wake. Thirdly, when varying the size of the examined bubbles, there is a critical volume at which the terminal velocity of bubbles jumps to a higher value.

Especially the last point is still subject of scientific discourse. Amirnia et al. summarized in Amirnia et al. (2013) the results of 19 experimental studies and three theoretical/numerical investigations. Twelve publications observed a velocity volume discontinuity, ten did not. In their own experiments, Amirnia et al. did not observe a velocity volume discontinuity. In most of the experiments, the viscoelastic fluid has been a polymer dissolved in water. Frequently used models that approximate the constitutive behavior of such fluids include the Oldroyd-B and Giesekus model Giesekus (1982), which are also the two models to be considered in this thesis.

Based on an adequate constitutive model, simulations can be very useful to gain insight into physical phenomena that are otherwise hard or impossible to detect. In the case of a bubble in viscoelastic fluid for example, the detailed velocity field in the wake of the bubble is of great interest, but experimentally very hard to determine.

Considering the numerical implementation, a code developed at the Chair for Computational Analysis of Technical Systems was used for all simulations conducted in this work. The code features a method for dealing with moving computational domains – as will be introduced by the deforming bubble – that nevertheless allows for boundary conforming meshes; i.e., we pursue an interface tracking approach. The aim of this study is to adapt the available mesh movement techniques to better cope with the large deformations caused by the moving and deforming bubble. Furthermore, the code
intended for viscoelastic simulations is for the first time employed to calculate the rise of a bubble.

The thesis is organized as follows: Section 2 describes the background of simulating two phase flow with the finite element method. The adaptations of the mesh movement technique are outlined in Section 3. The validation of the adapted methods with accepted Newtonian bubble dynamics benchmarks is shown in Section 4. In the following, Section 5 gives a short introduction to the modeling of viscoelastic fluids and to the respective implementation in the code. Section 6 presents the results of the two- and three-dimensional viscoelastic simulations and compares them to published numerical and experimental findings. Finally, Section 7 concludes this thesis.

2. Background on two phase flow simulation

2.1. Governing equations

Figure 1: Phase 1 in contained in domain $\Omega_1$, phase 2 in $\Omega_2$. $\Gamma_2$ separates the two phases.

The presented work considers instationary, incompressible, isothermal fluid flow of two immiscible phases. An illustration of the problem setup can be found in Figure 1. Fluid 1 is contained in domain $\Omega_1 \subset \mathbb{R}^{n_{sd}}$, with $n_{sd} = 2, 3$ being the number of space dimensions. Fluid 1 surrounds fluid 2 in $\Omega_2 \subset \mathbb{R}^{n_{sd}}$. The fluid-fluid-interface between $\Omega_1$ and $\Omega_2$ is denoted by $\Gamma_2$, while $\Gamma_1$ stands for the outer boundary of fluid 1. Note that the domains as well as boundaries are varying with time. In this and the following section both phases are modeled as Newtonian fluids, in Section 5 and 6 the matrix fluid ($\Omega_1$) is considered to be viscoelastic. In the mathematical description, we follow Gross and Reusken (2011) and state that the time-dependent flow obeys the incompressible Navier-Stokes equations in each fluid phase ($i = 1, 2$):

\[
\begin{align*}
\nabla \cdot \mathbf{u} &= 0 & \text{in } \Omega_i, & \forall \ t \in [0, T], \\
\rho_i (\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} - \mathbf{f}) - \nabla \cdot \mathbf{\sigma}_i &= 0 & \text{in } \Omega_i, & \forall \ t \in [0, T].
\end{align*}
\]

(1)
Therein, $\mathbf{u}(x, t)$ is the velocity vector, $\mathbf{\sigma}_C(x, t)$ is the Cauchy stress tensor, and $\mathbf{f}(x, t)$ is the vector of external body forces, such as gravity. The stress tensor $\mathbf{\sigma}_C_i$ and the density $\rho_i$ vary with the respective fluid. The constitutive equation for the Newtonian fluids is given by

$$\mathbf{\sigma}_C_i = -p \mathbf{1} + 2\mu_i \varepsilon(\mathbf{u}),$$

(2)

$$\varepsilon(\mathbf{u}) = \frac{1}{2} \left( \nabla \mathbf{u} + (\nabla \mathbf{u})^T \right).$$

(3)

The viscoelastic constitutive equation will be introduced in Section 5.

### 2.2. Initial, boundary, and interface conditions

For the transient simulation of the unsteady fluid flow, initial conditions are needed. We assume that both fluids are initially at rest:

$$\mathbf{u}(x, 0) = 0 \quad \text{in} \quad \Omega_i \text{ at } t = 0.$$  

(4)

In the case of a viscoelastic matrix fluid, the matrix fluid is further assumed to be stress free as the simulation starts.

On $\Gamma_1$, different types of boundary conditions are prescribed in the various test cases. They can be classified as Dirichlet boundary conditions, where a velocity value $\mathbf{u}$ is prescribed:

$$\mathbf{u} = \hat{\mathbf{u}} \quad \text{on} \quad \Gamma_u,$$

(5)

or Neumann boundary conditions, where a stress value $\mathbf{h}$ is prescribed:

$$\mathbf{n} \cdot \mathbf{\sigma}_1 = \hat{\mathbf{h}} \quad \text{on} \quad \Gamma_h.$$  

(6)

Slip boundary conditions combine Dirichlet and Neumann type boundary conditions for the individual degrees of freedom. For example, the two-dimensional Newtonian benchmark test case according to Hysing Hysing et al. (2009) uses slip boundary conditions on the vertical walls and no-slip boundary conditions on the horizontal walls. The boundary conditions will be further discussed for the individual test cases.

On the phase boundary $\Gamma_2$, we use the coupling conditions derived in Gross and Reusken (2011). Firstly, the velocity has to be continuous across the interface:

$$\mathbf{u}_1 = \mathbf{u}_2 \quad \text{on} \quad \Gamma_2.$$  

(7)

Secondly, stresses between the two domains are coupled by:

$$(\mathbf{\sigma}_{C_2} - \mathbf{\sigma}_{C_1}) \mathbf{n} = -\gamma \kappa \mathbf{n} \quad \text{on} \quad \Gamma_2.$$  

(8)

Here, $\mathbf{n}$ is the unit normal vector on $\Gamma_2$ pointing into $\Omega_1$, $\gamma$ is the surface tension coefficient, and $\kappa$ the curvature of $\Gamma_2$. 
2.3. Weak form

The first step towards discretization and implementation is the derivation of the weak form of the governing equations. To this end, test functions for velocity $w$ and pressure $q$ have to be chosen from admissible function spaces. The continuity equation (first line of Equation (1)) is multiplied by $q$, the momentum equation (second line of Equation (1)) by $w$, before both are integrated over the computational domains $\Omega_1$ and $\Omega_2$. Momentum and continuity equation can be combined by addition. Subsequently, selected terms are integrated by parts, which reduces the differentiability requirements on the trial functions. Integrating by parts also includes Neumann type boundary conditions and likewise the second interface condition (Equation (8)). The resulting weak form of the two-phase flow problem reads:

Find $u$ and $p$ from the trial function spaces such that $\forall w, \forall q$:

$$
\sum_{i=1}^{2} \left[ \int_{\Omega_i} w \cdot \rho_i \left( \partial_t u + u \cdot \nabla u - f \right) dx + \int_{\Omega_i} \varepsilon(w) \cdot \sigma_{C_i}(p, u) dx + \int_{\Omega_i} q \nabla \cdot u dx \right] = \int_{\Gamma_h} w \cdot \hat{h} dx - \gamma \int_{\Gamma_2} \kappa w \cdot n dx.
$$

(9)

Of particular importance is the numerical treatment of the surface tension (included by the last term of Equation (9)). Frequently, one of the two following ways is chosen. The Continuum Surface Force approach adds a volume force $f_{st} = \gamma \kappa \delta n$ to the strong form of the governing equations (Equation (1)). The Dirac delta function $\delta$ is used to concentrate the surface tension forces on the interface. This approach is used by Pillapakkam et al. (2007), Zainali et al. (2013), and Vahabi and Sadeghy (2015) for their viscoelastic two-phase flow computations, which we compare to in Section 6.

We use instead the Laplace-Beltrami technique as described by Elgeti et al. (2015). Here, $\kappa n$ in Equation (8) is replaced by $\Delta^g X$, the Laplace Beltrami operator $\Delta^g$ applied to the immersion $X$ that parametrizes the boundary of the domain. So, the following term is obtained on the right hand side of the weak form (Equation (9)):

$$
- \gamma \int_{\Gamma_2} w \cdot \Delta^g X dx.
$$

(10)

2.4. Deforming spatial domain space-time finite element method

The discretization of the weak form is based on the deforming-spatial-domain/stabilized space-time (DSD/SST) procedure as introduced in Tezduyar et al. (1992). The procedure uses finite elements for the spatial and temporal discretization. Hence, the variational form is written over the space-time domain of the problem. Deformations of the spatial computational domain over time – for example the circular $\Omega_2$ becomes elliptical – are automatically accounted for. They appear as deformation of the space-time elements. Consequently, this approach is very well suited for computations on moving meshes.
As the usage of equal order interpolation functions for the pressure and velocity approximation is intended, the Ladyzhenskaya-Babuška-Brezzi (LBB) compatibility condition is not fulfilled. To still obtain a stable approximate solution, a stabilization technique has to be applied. The stabilized space-time formulation for deforming domains can be found in Behr and Tezduyar (1994). Instead of the space-time formulation here only the spatial discretization is shown for brevity. With \( w^h \) and \( q^h \) being the test functions and \( u^h \) and \( p^h \) being the trial functions from a similar admissible finite function space, the spatially discretized weak form can be written as:

Find \( u^h \) and \( p^h \) such that \( \forall w^h, \forall q^h : \)

\[
\sum_{i=1}^{2} \left[ \int_{\Omega_i} w^h \cdot \rho_i (\partial_t u^h + u^h \cdot \nabla u^h - f) \, dx + \int_{\Omega_i} \varepsilon(w^h) : \sigma_i(p^h, u^h) \, dx \right. \\
+ \sum_{el=1}^{n_{el}} \int_{\Omega_i} \tau_{MOM} \frac{1}{\rho_i} \left[ \rho_i (\partial_t u^h + u^h \cdot \nabla u^h) + \nabla q^h + \mu_i \Delta u^h \right] \\
\left. \cdot \left[ \rho_i (\partial_t u^h + u^h \cdot \nabla u^h - f + \nabla p^h - \mu_i \Delta u^h) \, dx \right] \right]
\]

\[
+ \int_{\Omega_i} q^h \nabla \cdot u^h \, dx + \sum_{el=1}^{n_{el}} \int_{\Omega_i} \tau_{CONT} \rho_i \nabla \cdot \nabla u^h \, dx \\
= \int_{\Gamma} w^h \cdot \hat{h} \, dx - \gamma \int_{\Gamma_2} w^h \cdot \nabla \varphi X \, dx.
\]

This kind of stabilization is called adjoint Galerkin/Least Stabilization (GLS) Franca and Frey (1992). The therein used stabilization parameter for the momentum equation \( \tau_{MOM} \) is – in our simulations with first order elements – set to:

\[
\tau_{MOM} = \min \left( \frac{\rho_i h^2}{72 \mu_i}, \frac{h}{2|u|}, \frac{\Delta t}{2} \right) \quad \text{for } n_{sd} = 2,
\]

\[
\tau_{MOM} = \min \left( \frac{\rho_i h^2}{160 \mu_i}, \frac{h}{2|u|}, \frac{\Delta t}{2} \right) \quad \text{for } n_{sd} = 3.
\]

In simulations with second order elements, we use Knechtges (2015); Knechtges et al. (2014):

\[
\tau_{MOM} = \min \left( \frac{\rho_i h^2}{314 \mu_i}, \frac{h}{2|u|}, \frac{\Delta t}{2} \right) \quad \text{for } n_{sd} = 2,
\]

\[
\tau_{MOM} = \min \left( \frac{\rho_i h^2}{600 \mu_i}, \frac{h}{2|u|}, \frac{\Delta t}{2} \right) \quad \text{for } n_{sd} = 3.
\]

In the formula above \( \rho_i \) is the density and \( \mu_i \) the solvent viscosity of the different materials. In the calculation of the element length \( h \), the aspect ratio is taken into account. \( h \) is computed as smallest singular value of the mapping from the reference element to the physical element. \( \Delta t \) is the time step size. \( |u| \) is the velocity magnitude at the element center. On moving meshes, the relative velocity has to be used. Additionally, in the case of viscoelastic simulations, the time derivatives of the weighting functions
need to be included explicitly in the constitutive equation. The user can achieve both by writing the keywords *dtngls* on in addition to *relativeu* on in the input file (*xns.in*). The stabilization parameter for the continuity equation \( \tau_{\text{CONT}} \) is set to zero as the stabilization of the continuity equation is not necessary at low Reynolds numbers Knechtges et al. (2014), which is expected for all test cases.

Concerning the discretization of the degrees of freedom at the interface using continuous Lagrangian finite elements, the analysis of the interface conditions in Elgeti and Sauerland (2015) shows: The velocity is continuous, but has a kink across the interface due to the different viscosities. The pressure has a kink and a jump across the interface. In order to allow for the jump, the pressure degrees of freedom on the interface are doubled: One holds the pressure value of the outer phase, one of the inner phase. In viscoelastic simulations, also all other degrees of freedom besides the velocity are doubled at the interface. This allows the polymeric stresses – non-zero in the viscoelastic fluid – to jump to zero in the Newtonian bubble. In the implementation, all degrees of freedom on both sides of the bubble interface are connected using periodic boundary conditions.

### 2.5. Interface tracking

Since the computational domains, \( \Omega_1 \) and \( \Omega_2 \), deform in the course of the simulation it is necessary to monitor the interface evolution. For this task, two approaches have been widely used. The interface capturing approach, which is based on a Eulerian view, represents the interface implicitly by a scalar field. This scalar field is advected with the fluid velocity. A well-known representative of this approach is the level-set method, e.g., used by TP2D and FreeLIFE in the two-dimensional Newtonian benchmark computations Hysing et al. (2009) to which we compare in Section 4.1 and by Pillapakkam et al. Pillapakkam et al. (2007) for the three-dimensional viscoelastic computations to which we compare in Section 6.2.

In contrast, our method to resolve the bubble interface over time follows the interface tracking approach, which is based on a Lagrangian view. The domain boundary is directly represented by the element edges of the discretization. We expect this method to properly conserve the fluid masses, which is essential for obtaining correct results. The interface tracking method is not able to capture topological changes, therefore the simulations are only applicable to physical situations where no breakup or coalescence occurs.

The straightforward way to implement interface tracking is to directly use the fluid velocity at the nodes to move the nodes on the interface. This means the mesh velocity \( v(x) \) is equal to the fluid velocity:

\[
v(x) = u(x).
\]  

(14)
However, as Behr pointed out in Behr (1992) only the normal component of the mesh velocity needs to be restricted by the kinematic boundary condition:
\[ \mathbf{v}(x) \cdot \mathbf{n}(x) = \mathbf{u}(x) \cdot \mathbf{n}(x), \] (15)
to ensure that no fluid crosses the interface. The tangential component of the mesh velocity can be chosen freely – as long as aligned perfectly with the interface. Two choices for the tangential component will be discussed in Section 3.3.

### 2.6. Elastic mesh update

The idea of the elastic mesh update – proposed in Johnson and Tezduyar (1994) – is to have the mesh respond elastically to deformations of the boundaries or interfaces. Accordingly, the computational mesh is treated as linear elastic body in the domain \( \Omega_\# \subset \mathbb{R}^{nd} \). The equations of linear elasticity are solved for the mesh displacement \( \mathbf{\nu} \), which relates to the mesh velocity \( \mathbf{v} \) as \( \mathbf{\nu} = \mathbf{v} \cdot \Delta t \):

\[
\begin{align*}
\nabla \cdot \mathbf{\sigma}_\# &= 0 \quad \text{in } \Omega_\#, \\
\mathbf{\sigma}_\#(\mathbf{\nu}) &= \lambda_\#(\text{tr}(\mathbf{\varepsilon}_\#(\mathbf{\nu}))) \mathbf{1} + 2 \mu_\# \mathbf{\varepsilon}_\#(\mathbf{\nu}), \\
\mathbf{\varepsilon}_\#(\mathbf{\nu}) &= \frac{1}{2} (\nabla \mathbf{\nu} + (\nabla \mathbf{\nu})^T).
\end{align*}
\] (16)

\( \lambda_\# \) and \( \mu_\# \) are the Lamé parameters. In the implementation, their ratio \( \frac{\lambda_\#}{\mu_\#} \) appears as \texttt{mesh\_ratio}. The domain boundary \( \Gamma \) consists of the parts \( \Gamma_D \) and \( \Gamma_S \). On \( \Gamma_D \) prescribed displacements of the boundary or interface nodes \( \hat{\mathbf{\nu}} \) are incorporated as Dirichlet boundary conditions:

\[ \mathbf{\nu} = \hat{\mathbf{\nu}} \quad \text{on } \Gamma_D. \] (17)

Alternatively, we consider slip boundary conditions on \( \Gamma_S \). In this case, the coordinate system is aligned with the boundary/interface. A Dirichlet boundary condition is applied in normal direction and a Neumann boundary condition in tangential direction. The displacement in normal direction is given as \( \hat{\mathbf{\nu}}_n \). In tangential direction, the boundary or interface node slips along the boundary/interface:

\[
\begin{align*}
\mathbf{\nu} \cdot \mathbf{n} &= \hat{\mathbf{\nu}}_n \quad \text{on } \Gamma_S, \\
\mathbf{t} \cdot \mathbf{\sigma}_\# \cdot \mathbf{n} &= 0 \quad \text{on } \Gamma_S, \\
\mathbf{b} \cdot \mathbf{\sigma}_\# \cdot \mathbf{n} &= 0 \quad \text{on } \Gamma_S.
\end{align*}
\] (18)

\( \mathbf{n}, \mathbf{t} \) and \( \mathbf{b} \) are the normal, tangent and bi-tangent vectors on the boundary. With the computed mesh displacement, the nodal coordinates are updated.

For the previously considered problem setup (see Figure 1) the mesh and the elastic body extend over \( \Omega_1 \) and \( \Omega_2 \), so \( \Omega_\# = \Omega_1 \cup \Omega_2 \). The displacements of the interface nodes on \( \Gamma_2 \) are determined with the interface tracking method. Details on boundary conditions for the elastic mesh update on \( \Gamma_1 \) will be discussed in Section 3.2.
3. Adapting the mesh update strategy to rising bubbles

As mentioned previously, we are using an interface tracking approach to resolve the phase boundary over time. Challenging for this mesh movement technique is to deal with large deformations of the computational domain. Difficulties and solution attempts are studied for the two-dimensional benchmark test case published by Hysing and colleagues Hysing et al. (2009). The test case setup is depicted in Figure 2a. An initially circular bubble is placed into a heavier matrix fluid. For the fluid, no-slip boundaries are prescribed at the bottom and top of the computational domain. On the vertical side walls, the fluid motion is restricted in x-direction, but free in y-direction. The parameters are chosen according to test case 1 in the above cited publication. As gravity acts, the low density bubble rises in the high density matrix fluid. Concurrently, the bubble changes its shape. The quality of the fluid flow approximation is evaluated in Section 4.1. The current section focuses solely on the mesh quality.

3.1. Need for action

Figure 3a shows the initial mesh, which is refined in the vicinity of the bubble interface in order to increase the resolution of the phase boundary. The two-dimensional first order simplex mesh is generated with the Frontal algorithm within GMSH, a finite element mesh generator Geuzaine and Remacle (2009). To have a quantitative measure of the mesh quality, the minimum angle of the triangular elements is computed. In
Figure 3c, a histogram classifies the elements of the mesh according to their minimum angle. In the initial mesh, there is no element with a minimum angle of 35° or less. As the simulation progresses for 3000 time steps, the rising bubble changes its shape and position significantly. The corresponding mesh is displayed in Figure 3b. In general, the elements below the bubble are stretched while the elements above are compressed. In this configuration, 72 elements have a minimum angle of 5° and below. The absolute minimum angle between two edges of the mesh is 1.76°, the absolute maximum angle is 175.7°. These are indicators for a poor mesh quality. Shewchuk shows in Shewchuk (2001) that extreme angles close to 0° and 180° are harmful for the stiffness matrix conditioning, while large maximum angles cause large errors in the finite element approximation of the solution gradients. Also, kinks in the finite element approximation become visible along the edges of long and stretched elements. Moreover, the mesh update itself runs into problems with ill-shaped triangles. The node associated with a very large maximum angle in a triangle, might be moved across the opposite side of the triangle by the mesh update. This inverts the orientation of the element and changes the sign of the Jacobian. In this case, we speak of a mesh that tangles. Elements with an inverted orientation can cause the simulation to fail.

(a) Initial mesh  
(b) Mesh after 3000 time steps  
(c) Mesh quality

Figure 3: As the bubble rises, the mesh deformation causes ill-shaped elements.
3.2. Rigid body motion of the computational domain

The deforming-spatial-domain/stabilized space-time (DSD/SST) procedure allows computations on moving meshes Tezduyar et al. (1992). Similar to an Arbitrary-Lagrangian-Eulerian (ALE) approach, one can combine Eulerian and Lagrangian reference frames in this context. We will apply them in the following way: The complete mesh follows the bubble in a Lagrangian fashion. Inside this moving domain, the fluid is passing through the elements not influencing the elements’ position – hence a Eulerian description. An exception are the nodes on the bubble interface, which are moved in normal direction to the interface with the fluid velocity in normal direction at these points – which is again a Lagrangian technique.

The first approach to preserve a good mesh quality throughout the course of the simulation is to move the entire mesh with the rise velocity of the bubble. The rise velocity is computed as:

$$U_r = \frac{\int_{\Omega_2} u_v \, dx}{\int_{\Omega_1} 1 \, dx}, \quad (19)$$

with $u_v$ being the velocity component in vertical direction. $U_r$ is multiplied with the time step size to obtain a length. This length is than is prescribed as a Dirichlet boundary condition in the elastic mesh update.

In terms of the implementation, the rise velocity is computed and treated as a component of the mean velocity of the material which occupies the bubble. The materials’ mean velocity is initialized with zero and then updated in every time step. Further, it is made accessible to the input lexer. The user can access it in the input file ($xns.in$) with the keyword meanvel. For example, the line

```plaintext
rngxexp 5 2 mat2.meanvel.v * dt
```

prescribes on reference node group (or boundary) 5 a displacement in y-direction. This displacement is the second component ($v$) of the mean velocity (meanvel) of material 2 ($mat2$) multiplied with the time step size ($dt$).

For computations in which the entire computational domain is moving with the bubble rise velocity, the initial setup is modified. In order to minimize the influence of the horizontal noslip boundaries at the bottom and top of the domain, the initial bubble is not placed at the bottom of the domain as in Figure 2a, but in the middle of the domain (Figure 2b). Further, placing the bubble at the center of the mesh is helpful in the generation of a high quality mesh and avoids unnecessary computational domain above the bubble. This modification influences the solution only negligibly as will be shown in Section 4.

In a simulation with a fixed computational domain the bubble eventually reaches the top of the computational domain. So, to simulate a long period of time, a very tall computational domain is necessary. In case of a moving domain only the interesting
Figure 4: The mesh quality after 3000 time steps is slightly improved when using a moving computational domain. Employing the mesh slip along the fluid-fluid-interface in addition, decreases the number of ill-shaped element enormously.

Computational domain around the bubble is needed and moved with the bubble. This adaptation allows us to simulate much longer time periods. Additionally, for viscoelastic simulations it might not be clear at the beginning of the simulation how long we have to simulate to see the viscoelastic phenomena or how fast the bubble will rise. This uncertainty makes is very hard to determine a suitable size of a fixed computational domain before the simulation. Therefore, it is helpful to use a moving computational domain.

Moving the computational domain likely effects the boundary conditions on parts of the domain boundary. In our example, the movement of the domain in y-direction has the effect that the upper boundary develops an inflow characteristic. This has to be considered, especially in the viscoelastic case where the bubble travels through an initially stress-free matrix fluid.

### 3.3. Mesh slip along fluid-fluid-interface

For our specific example, however, the rigid body motion alone increased the quality of the mesh after 3000 time steps only slightly, see Figure 4. Reason is the mesh boundary condition on the fluid-fluid-interface. The normal displacement of the nodes on the interface is fixed by the interface tracking procedure. The tangential displacement, in contrast, is free to choose. Initially, the nodes’ displacement in tangential direction was set to zero.

Figure 5a shows an excerpt of the mesh after 3000 time steps with the just mentioned boundary condition (noslipx). Due to the above mentioned constraint, interface nodes with a vertical tangent cannot move upwards together with the bubble and consequently
accumulate on the lower part of the bubble. This leads to elements stretched along the interface in the upper part of the bubble and elements compressed along the interface on the lower part.

After adapting the alignment procedure (align.F) for periodic boundaries, the slip boundary condition in tangential direction (slipx) can now be applied to these boundaries. The elastic mesh update is now able to freely move the interface nodes along the phase boundary. This results in a preferable, almost equidistant spacing of the nodes along the interface throughout the simulation. Figure 5b shows the corresponding mesh after 3000 time steps. Also, in the histogram of Figure 4, it is visible how the combination of the slipx boundary condition with the rigid body motion of the computational domain improved the mesh quality considerably.

3.4. An error adaptive mesh update

Recently, Zwicke implemented a residual-based error estimator Zwicke (2014). It indicates elements and mesh regions where the mesh quality is not sufficient to properly resolve the flow phenomena. One possible use of such an error estimator is to modify the mesh based on the error with the aim of reducing the latter through a more adequate distribution of the mesh nodes. By "more adequate" we mean a clustering of nodes in regions with large error. The mesh adaptation will work with the existing mesh; no new nodes will be introduced. The presented approach avoids the cost and inaccuracy introduced by remeshing. To perform the mesh adaptations, the error estimate is
included in the elastic mesh update (Section 2.6). Body forces are derived from the error estimate and applied to the fictitious elastic solid in the mesh update.

### 3.4.1. Derivation of weakly incorporated error-induced body forces

The behavior of the fictitious elastic solid is governed by the equations of linear elasticity given in Equation (16). A variational formulation of the linear elastic solid, including prescribed body forces can be found in Johnson and Tezduyar (1994). Following this exposition, we introduce the finite element function spaces for the test functions \( w^h \) and trial functions \( \nu^h \):

\[
W^h := \{ w^h \in [H^1(\Omega_\#)]^{n_{sd}}, w^h = 0 \text{ on } \Gamma_D \text{ and } w^h \cdot n = 0 \text{ on } \Gamma_S \},
\]

\[
V^h := \{ \nu^h \in [H^1(\Omega_\#)]^{n_{sd}}, \nu^h = \hat{\nu} \text{ on } \Gamma_D \text{ and } \nu^h \cdot n = \hat{\nu}_n \text{ on } \Gamma_S \}.
\]

\( H^1(\Omega) \) is the Sobolev space consisting of the square integrable functions with one square integrable derivative. \( H^1(\Omega_\#) \) is the finite dimensional function space on the mesh domain \( \Omega_\# \). We use either linear or quadratic interpolation functions. \([H^1(\Omega_\#)]^{n_{sd}}\) is the space of vector functions with \( n_{sd} \) components for which each component is in \( H^1(\Omega_\#) \). The trial functions fulfill the boundary conditions on \( \Gamma_D \) and \( \Gamma_S \) with the precision given by the mesh.

Assuming Dirichlet boundary conditions on the boundary \( \Gamma_D \) and slip boundary conditions on \( \Gamma_S \) as given in Equation (17) and (18) the discretized weak form reduces to:

\[
\int_{\Omega_\#} \varepsilon_\#(w^h) : \sigma_\#(\nu^h) \, d\Omega - \int_{\Omega_\#} w^h \cdot f \, d\Omega = 0.
\]

The element-wise error estimate \( e_{est} \) normalized with the maximum estimated error \( e_{max} \) can be interpreted as a potential field \( e = \frac{e_{est}}{e_{max}} \). Based on this field one can calculate forces pointing towards larger errors estimates as: \( f = -\nabla e \). These forces are inserted in Equation (21):

\[
\int_{\Omega_\#} \varepsilon_\#(w^h) : \sigma_\#(\nu^h) \, d\Omega - \int_{\Omega_\#} w^h \cdot (-\nabla e) \, d\Omega = 0.
\]

Using the product rule, the body forces term can be rewritten as:

\[
\int_{\Omega_\#} w^h \cdot (\nabla e) \, d\Omega = \int_{\Omega_\#} \nabla \cdot (e w^h) \, d\Omega - \int_{\Omega_\#} e (\nabla \cdot w^h) \, d\Omega.
\]

With the divergence theorem follows:

\[
\int_{\Omega_\#} \nabla \cdot (e w^h) \, d\Omega = \int_{\Gamma} (e w^h) \cdot n \, d\Gamma = 0
\]

The last equality holds for the Dirichlet boundary \( \Gamma_D \), and the normal displacement degree of freedom of the slip boundary \( \Gamma_S \), because the test functions \( w^h \) vanish by
After 1 time step

After 2 time steps

After 4 time steps

After 8 time steps

Figure 6: The error adaptive mesh update stretches the elements in the upper part of the mesh and compresses the elements in the lower part to balances the estimated error.

definition (Equation (20)). In tangential and bi-tangential direction the projection \((e w^h) \cdot n\) vanishes on \(\Gamma_S\) (see Section 2.6).

Inserting Equation (24) and (23) in Equation (22), we obtain:

\[
\int_{\Omega^h} \varepsilon^h \varepsilon^h \cdot \sigma^h \nu^h \, d\Omega - \int_{\Omega^h} e (\nabla \cdot w^h) \, d\Omega = 0.
\] (25)

The advantage of this formulation is that we can use the normalized element-wise error estimate \(e\) directly and do not have to calculate its gradient.

3.4.2. First results

The first test case for the error adaptive mesh update is a simple Poiseuille flow through a rectangular channel with dimensions \(L_x = 4\) and \(L_y = 1\). The Stokes equations, which can be obtained from Equation (1) by neglecting the inertia term \((u \cdot \nabla u)\), are solved with a parabolic flow profile prescribed at the inlet on the left side of the channel. The parabolic velocity profile – symmetric with respect to the line \(y = 0.5\) – is expected to propagate through the channel. For the details of the simulation see Appendix A.

The initial mesh (cf. Figure 6) is generated along seven horizontal lines dividing the mesh into eight horizontal domains, each discretized with first order triangular (P1) elements. The vertical spacing of the lines is such that four horizontal domains discretize the upper quarter of the rectangle and four the lower three quarters. This leads to a
(a) After 1 time step  
(b) After 8 time steps

Figure 7: Larger error estimates and mesh changes are observed around the re-entrant corner.

high resolution in the upper quarter of the rectangle and a less accurate resolution in the lower three quarters.

As expected, the error estimated after one time step is larger in the lower three quarters of the channel. In the following time steps, the elements in the lower part of the domain are compressed and the ones in the upper part stretched by the error-induced body forces. After eight time steps, the mesh size in the upper and lower part of the channel is approximately equal and the estimated error is distributed nearly evenly across the domain. In the inlet and outlet region, the error distribution is disturbed. Still, for this simple test case and over the few time steps considered, the algorithm seems to work in that sense that elements with large error estimates are shrunk.

The simulation is repeated on a more complex domain and with half the time step size. The channel is extended in vertical direction resulting in an L-shaped computational domain with re-entrant corner (see Figure 7). On the initially uniformly spaced mesh, larger errors (and error estimates) are expected only in vicinity of the re-entrant corner.

In Figure 7, the element-wise error estimates and meshes after one and eight time steps are compared. Significant error and mesh changes are restricted to the close neighborhood of the re-entrant corner. Therefore, a close-up of the computational domain around the corner is shown in Figure 8. We find that over the depicted eight time steps, the size of elements is decreased, according to the element-wise error estimate.

However, already after 8 time steps we find one element close to the re-entrant corner ill-shaped (indicated for example by a small minimum angle). Cases, where the error of an element remains larger than the error in the surrounding elements despite the small size of this element are problematic. As the algorithm has no means to control the mesh quality, the mesh quality deteriorates following a vicious circle: Once an element has
attained a small minimum angle, the error estimate increases. This leads to a further compression of the element, further decreasing the minimum angle and so on. The fatal consequences can be seen in Figure 9a, showing the mesh after 20 time steps. Some elements have such a bad shape that they are not distinguishable from a line at the used resolution.

Besides a small minimum angle or a large maximum angle, ill-shaped elements can be identified by a large condition number of the Jacobian of the mapping from the reference element to the global element. Knupp proposes the so constructed condition number as objective function for mesh optimization Knupp (1999). We will denote the Jacobian of
the mapping from reference to physical coordinates by $J_K$. The condition number of the mapping is given by the ratio of the singular values, which can be computed as follows:

$$
\kappa_2(J_K) = \frac{\sigma_{\text{max}}}{\sigma_{\text{min}}} = \sqrt{\frac{\lambda_{\text{max}}(J_K^T J_K)}{\lambda_{\text{min}}(J_K^T J_K)}}
$$  \hspace{1cm} (26)

where $\lambda_{\text{min/max}}$ is the minimum/maximum eigenvalue of the square matrix $J_K^T J_K$. With $\lambda_{\text{min}}(J_K^T J_K) = \frac{1}{\lambda_{\text{max}}(J_K^{-1})}$, and $r_K = 2 \cdot \frac{\text{det}(J_K^{-1})}{\text{tr}(J_K^{-1})}$, the condition number can – in the two-dimensional case – be rewritten as:

$$
\kappa_2(J_K) = 1 + \frac{1}{r_K} \sqrt{1 - r_K^2}
$$  \hspace{1cm} (27)

To add some mesh quality information to the update procedure, the following idea was pursued: In `blksldlin.F` all the terms contributing to left and right hand side except for the error induced body forces are multiplied with a factor $k$. In the first simulation, $k$ is chosen as the condition number of the mapping from the reference to physical coordinates, $k = \kappa_2(J_K)$. For elements with small condition number – meaning they are well-shaped, since similar to the reference element – the body forces still have a significant contribution. Elements with a large condition number, in the contrary, are less influenced by the error induced body force. This way, the further compression of an already ill-shaped element should be avoided. In the following, this procedure is called $k$-weighting.

Figure 9 compares the result with $k$-weighting to the previous results. For after 20 time steps, the triangle shapes around the re-entrant corner are substantially improved by the $k$-weighting. However, it seems that problems are only postponed. After 60 time steps, the mesh quality also deteriorates with $k = \kappa_2(J_K)$. Still, as this appears to be a step in the right direction, further steps were investigated. In Figure 10, the meshes obtained with different $k$-parameters are compared after 100 time steps. Clearly visible,
(a) $k = \kappa_2(J_K)$
(b) $k = (\kappa_2(J_K))^2$
(c) $k = \exp(\kappa_2(J_K))$

Figure 10: The meshes after 100 time steps show that a stronger $k$-weighting yields better shaped elements around the re-entrant corner.

The stronger influence of the condition number improves the mesh quality after 100 time steps.

Here, I list some ideas that could further improve the error adaptive mesh update: Firstly, $k$ could be chosen independently of the mesh quality to make the error adaptive mesh update overall more or less aggressive. Secondly, considering the mesh depicted in Figure 10c, it is observed that edge flips could improve the mesh quality enormously (for details on edge flips see for example Shewchuk (1997) page 14). Thirdly, the implementation of the elastic mesh update has two parameters that control the stiffness of the individual elements (mesh_ratio and mesh_fraction). It was seen that in particular mesh_fraction $\in [0, 1]$, which controls how much of $J_K$ is included in the numerical computation of the first term of Equation (21), has a very strong influence on the mesh obtained with the error adaptive mesh update. Hence, fine tuning this parameter will most like improve the mesh. For some preliminary viscoelastic simulations the mesh parameters were varied, but it seemed that the default values (mesh_ratio = 1.0 and mesh_fraction = 0.0) worked the best for the viscoelastic bubbles.

Concluding the section of the error adaptive mesh update, it is important to note that the error adaptive mesh update did not converge to a steady solution of the mesh equation for any of the tests performed. This is the case although geometry and flow situation remain unchanged or nearly unchanged. So far, the algorithm was not tested for a transient simulation with deforming computational domain. In the simulations described in Section 4 and 6, the rigid body motion of the mesh (Section 3.2) and the
slip boundary condition for the fluid-fluid-interface (Section 3.3) are used, but the error adaptive mesh update (Section 3.4) is not employed.

4. Benchmarks with Newtonian fluid

This section attempts to validate our implementation of the described mesh movement techniques. As test cases that challenge the above mentioned techniques cannot be solved analytically, numerical benchmarking – meaning the comparison with the results of other simulations – is necessary. In the following subsections, the already mentioned two-dimensional test case "TC1-Hysing" provided by Hysing et al. (2009) and the three-dimensional test case "TC2-Adelsberger" provided by Adelsberger et al. (2014) are studied. For the discretization, both first and second order elements are used.

4.1. Two-dimensional test case "TC1-Hysing"

\[
\begin{align*}
L_x &= 1 \\
L_y &= 2 \\
c_x &= 0.5 \\
c_y &= 1 \\
2r &= 0.5
\end{align*}
\]

\[
\begin{array}{cccccc}
\rho_1 & \rho_2 & \mu_1 & \mu_2 & \gamma & g & \Delta t \\
1.0 & 0.1 & 0.01 & 0.001 & 0.0245 & 0.98 & 0.001
\end{array}
\]

Figure 11: Two-dimensional test case "TC1-Hysing" Hysing et al. (2009): Initial geometry and parameters

Hysing et al. proposed two benchmark test cases for two-dimensional bubble dynamics in 2009 Hysing et al. (2009). Since then the test cases have frequently been referred to, e.g., by Zainali et al. (2013); Elgeti and Sauerland (2015); Vahabi and Sadeghy (2015). In the first test case the parameters are chosen such that the bubble does not break apart. The adapted initial geometry and parameters that we use in our benchmark computation can be found in Figure 11.

The publication of Hysing et al. Hysing et al. (2009) collects the simulation results of three groups. The first contributing code is TP2D (Transport Phenomena in 2D), developed at the Institute of Applied Mathematics at TU Dortmund Turek (1999). The discretization is done with quadrilateral finite elements and a level-set method is used to
describe the phase boundary. Based on triangular finite elements and and a similar level-set method is the code FreeLIFE (Free-Surface Library of Finite Elements), developed at the Institute of Analysis and Scientific Computing at EPFL Lausanne Parolini (2004). The third reference data contributor is MooNMD (Mathematics and object-oriented Numerics in MagDeburg) John and Matthies (2004), a mixed finite element code. For the bubble interface, an Arbitrary Lagrangian-Eulerian approach involving remeshing is used.

In order to quantitatively compare the simulation, results three quantities of interest are defined in Hysing et al. (2009), namely the center of mass coordinates, the circularity and the rise velocity. As the translation of the bubble is described by the center of mass coordinates as well as by the rise velocity and the measurements of both are inherently coupled, we will describe the motion of the bubble just with the rise velocity, computed as in Equation (19). For the description of the bubble shape, the circularity is used. The circularity relates the perimeter of the currently observed bubble to the perimeter of a perfect circle with the same area as the bubble. We compute the circularity \( \phi \) in terms of the bubble area \( |\Omega_2| = \int_{\Omega_2} 1 \, dx \) and the bubble perimeter \( |\Gamma_2| = \int_{\Gamma_2} 1 \, dx \) as

\[
\phi(t) = \frac{2(\pi|\Omega_2(t)|)^{\frac{1}{2}}}{|\Gamma_2(t)|}.
\]

(28)

The prefactor in the formula ensures that a circular bubble has a circularity of 1.

The simulation is performed on six different discretizations: An unstructured coarse, medium and fine grid consisting of first (P1) or second order (P2) triangular elements, respectively. Table 1 lists the characteristics for the individual meshes. \( \text{ne} \) stands for the number of elements, \( \text{nn} \) for the total number of nodes and \( \text{nni} \) for the number of nodes discretizing the phase interface. The finer meshes are obtained by splitting the coarser grid element edges in the middle. The splitting preserves the geometry and is performed with GMSH.

<table>
<thead>
<tr>
<th></th>
<th>P1 Coarse</th>
<th>P1 Medium</th>
<th>P1 Fine</th>
<th>P2 Coarse</th>
<th>P2 Medium</th>
<th>P2 Fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{ne} )</td>
<td>1742</td>
<td>6968</td>
<td>27872</td>
<td>480</td>
<td>1920</td>
<td>7680</td>
</tr>
<tr>
<td>( \text{nn} )</td>
<td>1918</td>
<td>7318</td>
<td>28570</td>
<td>2106</td>
<td>8050</td>
<td>31458</td>
</tr>
<tr>
<td>( \text{nni} )</td>
<td>52</td>
<td>104</td>
<td>208</td>
<td>56</td>
<td>112</td>
<td>224</td>
</tr>
<tr>
<td>CPUs</td>
<td>4</td>
<td>8</td>
<td>32</td>
<td>4</td>
<td>8</td>
<td>24</td>
</tr>
<tr>
<td>Time</td>
<td>0:22 h</td>
<td>0:57 h</td>
<td>2:19 h</td>
<td>0:36 h</td>
<td>1:35 h</td>
<td>12:49 h</td>
</tr>
</tbody>
</table>

Table 1: The simulation of the two-dimensional test case “TC1-Hysing” is performed on six different meshes. One mesh for first order interpolation functions and one for second order interpolation functions is generated. These two are each refined twice. The computations with second order interpolation functions are computationally more expensive.
The mesh coordinates move in the course of the simulation as described above, but the number of degrees of freedom remains constant. No remeshing is performed. In order to indicate the computational effort needed for a certain degree of accuracy, the number of processors (CPUs) and the wall clock time (Time) is given in the last two rows. As time-step size $\Delta t = 10^{-3}$ is chosen. The input file with all simulation parameters can be found in Appendix B.

Figure 12 displays bubble velocity and circularity during the simulations based on $P1$-elements. For the plots on the left – covering the full simulated time of three time units – only the finest grid solution of MooNMD is displayed as reference since no significant difference between the finest grid computations of the benchmark can be noticed at this scale Hysing et al. (2009). The coarse, medium and fine mesh show the same behavior as the benchmark at this scale. The right-hand side plots zoom in on the point of the maximum bubble velocity and the minimum circularity. The choice of the time axis range follows the benchmark paper. In these plots, the finest grid solutions of TP2D, FreeLIFE and MooNMD are displayed as benchmarks. The velocity benchmark curves are approached from below as the grid is refined. The minimum circularity is
Figure 13: Test case "TC1-Hysing", discretization with P1 elements: Quasi-steady state bubble shapes ($t = 3$) and volume conservation

approached from above. The same is observed in the mesh refinement studies of the three benchmark codes Hysing et al. (2009).

The bubble interfaces at $t = 3$ are shown in Figure 13. All three meshes result in the same ellipsoidal shape for the quasi-steady state. To find small differences one has to zoom in in regions of high curvature. On the right side of Figure 13, the bubble volume approximation of the coarse, medium and fine mesh is compared to the exact value $\pi \cdot 0.25^2 \approx 0.19634954$. Firstly, one should notice that the bubble volume is almost perfectly conserved. The slip of interface nodes tangential to the bubble boundary does not artificially increase or decrease the bubble domain. Secondly, it is observed that the spacing of the volume curves is similar to the spacing of the predicted maximum velocity and minimum circularity. That the volume approximation influences the quantities of interest is not surprising as it enters directly into the computation of these quantities. However, the similar spacing indicates that the dominating error in the simulation is introduced by the volume approximation.

The simulations are repeated on the $P_2$-meshes. Figure 14 displays the respective results for rise velocity and circularity. The global plots on the left indicate very good general agreement with the finest grid MooNMD solution. The local plots on the right show that our solution is grid-converged. The twice repeated splitting of the elements does not lead to a significant difference in the velocity or circularity curve at this scale. The velocity maximum and circularity minimum of all three computations are close to the ones of the fine $P_1$-mesh. The excellent performance of the second order elements can be explained by their capability to very accurately resolve the curved bubble shape, leading to a superior approximation of the bubble volume even with coarse grids (Figure 15).

Non of the benchmark computations (TP2D, FreeLIFE and MooNMD) are converged to an accuracy similar to our $P_2$-simulations (Figure 4b, 7b and 9b in Hysing et al. (2009)). The finest grid solutions of TP2D and MooNMD are very close to each other (right-hand side plots of Figure 14). The difference between our finest grid solution and these two
Figure 14: Test case "TC1-Hysing", discretization with P2 elements: Rise velocity and circularity

Figure 15: Test case "TC1-Hysing", discretization with P2 elements: Quasi-steady state bubble shapes ($t = 3$) and volume conservation

is about the same as the difference between these two and FreeLIFE. This difference might be attributable to the adapted initial geometry that we use. However, for the simulation results on a global scale (left-hand side plots), the influence of our employed mesh movement techniques is negligible.
4.2. Three-dimensional test case "TC2-Adelsberger"

Figure 16: Initial geometry and parameters of the three-dimensional test case "TC2-Adelsberger" Adelsberger et al. (2014). The sphere is initially centered at $x = 0.5$, $y = 0.5$, $z = 1$. The subscript 1 denotes the heavier matrix fluid, 2 the bubble.

<table>
<thead>
<tr>
<th>$\rho_1$</th>
<th>$\rho_2$</th>
<th>$\mu_1$</th>
<th>$\mu_2$</th>
<th>$\gamma$</th>
<th>$g$</th>
<th>$\Delta t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.1</td>
<td>0.01</td>
<td>0.001</td>
<td>0.0245</td>
<td>0.98</td>
<td>0.001</td>
</tr>
</tbody>
</table>

From the previously discussed two-dimensional benchmark, Adelsberger et al. Adelsberger et al. (2014) derived benchmark computations for three-dimensional incompressible two-phase flow. Again, two test cases are proposed. Since our interface tracking procedure cannot account for topological changes, only test case 1 is computed. A noteworthy difference to the two-dimensional benchmark test case is that noslip boundary conditions are prescribed on all parts of the outer boundary $\Gamma_1$. The adapted initial geometry and parameters for this benchmark computation can be found in Figure 16.

The reference data given in Adelsberger et al. (2014) was established with three different flow solvers each adopting a different numerical technique. DROPS Gross and Reusken (2011), developed at the Chair for Numerical Mathematics at RWTH Aachen, uses extended finite elements (XFEM) for the spatial discretization and a level-set method to capture the bubble interface. At the Institute for Numerical Simulations at University of Bonn the NaSt3DGPF Croce et al. (2010) code is developed. Here, the spatial discretization is performed with finite differences and the bubble interface is again captured with a level-set method. The third flow solver contributing to the reference data is OpenFoam Ope (2013), which is based on finite volumes and uses the volume-of-fluid method to capture the bubble interface.

In order to compare the solutions quantitatively, four quantities of interest were defined in Adelsberger et al. (2014), namely the barycenter of the bubble, the rise velocity, the diameter and the sphericity. We focus on the rise velocity describing the bubble motion and the sphericity as quantitative description of the bubble shape. The rise velocity is obtained by integrating the velocity in vertical direction over the bubble and normalizing with the bubble volume. The sphericity $\Psi(t)$ indicates the similarity of
the bubble to a sphere. It is computed by first calculating the surface area of a sphere, which has the same volume as the bubble ($|\Omega_2(t)|$), and then dividing it by the current bubble surface area ($|\Gamma_2(t)|$):

$$\Psi(t) = \frac{\pi^\frac{1}{3}(6|\Omega_2(t)|)^\frac{2}{3}}{|\Gamma_2(t)|}.$$ \hspace{1cm} (29)

Two tetrahedral meshes are constructed: one with linear and one with quadratic interpolation functions. Both are refined towards the bubble interface. The mesh denoted as $P1$ consists of 38686 linear tetrahedrons with a total of 18186 nodes of which two times 1605 are distributed on the bubble interface. The characteristic length (side length of a of tetrahedron) on the interface is $l_c = \frac{2\pi r}{64} \approx 0.025$. It has a similar size as the one used in the DROPS computation ($l_c = \frac{1}{32} = 0.03125$). The second mesh, denoted with $P2$, is formed of 11939 quadratic tetrahedrons with together 39298 nodes. Here, the interface has two times 1734 nodes. A characteristic length of $l_c = \frac{2\pi r}{44} \approx 0.044$ on the interface was used during the mesh generation. The time step size for both simulations is set to $\Delta t = 10^{-3}$. Further details about the parameters of the simulations can be found in the input file in appendix C.

The simulation with mesh $P1$ took 8 hours on 16 processors, the simulation with mesh $P2$ took 36 hours on 32 processors. For a performance comparison with the published data, the numerical effort is used as a metric. In this work we define numerical effort as wall clock time times number of CPUs. In our $P1$-simulations the numerical effort was significantly smaller than the numerical effort reported for the three reference computations. The $P2$-simulation required more numerical effort than the DROPS computation, but much less than NaSt3d and OpenFOAM (DROPS: two weeks on single processor, NaSt3d: one week on 32 processors, OpenFOAM: 2.5 days on 32 processors) Adelsberger et al. \(2014\).

In accordance with the observations reported in the benchmark paper, the bubble initially accelerates for roughly the first third of the simulation. This is followed by a period of moderate deceleration before reaching a quasi-steady state, in which the bubble rises with almost constant velocity. A graph of the velocity evolution can be found in Figure 17. As in the benchmark paper, it is zoomed in on the time from $t = 0.7$ until $t = 1.5$. The $P1$-simulation, which has the less accurate approximation of the bubble volume compared to the $P2$-simulation, has its velocity maximum at the same height as the OpenFOAM computation. From there, the curve progresses parallel to the DROPS and NaSt3D computations, resulting in a terminal velocity closer to DROPS and NaST3D than OpenFOAM. The $P2$-simulation, which offers the more accurate approximation of the bubble volume, lies in-between the DROPS and NaST3D computation, showing very good agreement with the data of these two. The OpenFOAM computation appears to be an outlier concerning the bubble velocity.

The bubble shape evolves as follows: After a short period of time, in which the bubble rises in its initial spherical shape, the rear end of the bubble moves faster than the
equatorial regions, since the latter are stronger influenced by the retarding force of the matrix fluid. This deformation, however, is limited by the surface tension forces, leading to the stable ellipsoidal shape in the quasi-steady state. The two lower graphs of Figure 17 show the sphericity evolution. The authors of Adelsberger et al. (2014) consider "the sphericity results" to be "in high agreement as the differences at steady state are below 0.5% at most". The three benchmark codes and our P1- and P2-computation come to a very similar result. A closer look shows, that the result of the P2-computation is in very good agreement with DROPS and NaSt3d. From their mean value for the terminal sphericity – which is most likely close to the exact value – the P1-computation is offset to the top (probably due to the less accurate volume approximation). The distance from the OpenFOAM-curve to the NaSt3D-curve (both part of the benchmark) is larger then the distance from the P1-curve to the NaSt3D-curve.

Figure 18a displays the bubble in the P2-simulation at $t = 3$. The second order tetrahedrons allow a very accurate representation of the bubble surface even in regions of high curvature. The bubble volume approximation of the P1- and P2-mesh is plotted against the exact value of $\frac{4}{3} \cdot \pi \cdot 0.25^3 \approx 0.06544985$ in Figure 18b. At $t = 3$, the bubble
volume in the $P_2$-simulation deviates with a value of $|\Omega_2(t = 3)| = 0.06544628$ only by $5.45 \cdot 10^{-3}\%$ from the exact value.

The two previous sections are summarized as follows: The described methods allow us to execute simulations of a rising bubble for any desired time period. Given that the bubble undergoes moderate deformation, a good mesh quality is preserved. Based on the agreement with the reference data, the implementation is considered validated and we can go on to viscoelastic simulations.

5. Background on viscoelastic fluid flow simulations

5.1. Modeling viscoelastic fluids

In Section 2, the constitutive equation to determine the Cauchy stresses $\sigma_C$ for a Newtonian fluid is stated in Equation (2) and (3). For a viscoelastic fluid, choosing a suitable constitutive equation is the most intricate part of modeling. The presented outline on the choice of the constitutive equation follows the introduction in "Computational Rheology" by Owens and Phillips Owens and Phillips (2002).

What a viscoelastic fluid is, can be well understood when approaching it from the two extremes between which it lies. The one extreme is the Newtonian fluid model which adequately predicts the behavior of some well-known fluids like water and air. The other extreme is a linear elastic solid which can be used to approximate the mechanical behavior of many solids at least for a certain range of deformation.

Let us consider a two-dimensional continuum initially occupying a rectangular domain. The continuum is forced, e.g., by a prescribed displacement at the top, to leave equilibrium and attain a constant strain state. Assuming the continuum is a linear elastic solid,
the shear stresses depend linearly on the strain and therefore attain a finite constant value as soon as the strain state is reached. The relationship between shear stress and strain is known as Hooke’s law:

$$\sigma_{Cxy} = G\gamma_{xy}, \quad (30)$$

where $G$ is a constant named Young’s modulus and $\sigma_{Cxy}$ and $\gamma_{xy}$ denote the shear components of the Cauchy stress and strain tensor.

When the continuum is modeled as Newtonian fluid, the shear stresses depend on the rate of strain, meaning the temporal derivative of the strain. Consequently, shear stresses will be nonzero only as the strain state changes. Once a constant state is reached the shear stresses vanish. The corresponding equation is known as Newton’s hypothesis:

$$\sigma_{Cxy} = \mu_0 \dot{\gamma}_{xy}, \quad (31)$$

where $\mu_0$ is the constant fluid viscosity.

The time required for the shear stresses in a continuum to relax after a constant strain state is reached is denoted as relaxation time $\lambda$. For the linear elastic solid one observes an infinitely long relaxation time as shear stresses do not relax at all. For the Newtonian fluid, on the contrary, one observes a relaxation time of zero, as shear stress disappear immediately. The relaxation time of a real material, however, will have a value somewhere between zero and and infinity. Such a finite non-zero relaxation time is "the defining characteristic of so-called viscoelastic materials" Owens and Phillips (2002).

A finite non-zero relaxation time can be achieved by combining viscous effects of the Newtonian fluid and elastic effects of the linear elastic solid. A simple combination, consisting of a spring and a damper element in series, results in the Maxwell model (see Figure 19a). When a continuum described by the Maxwell model is subjected to constant strain, shear stresses first attain a finite value and then decrease exponentially:

$$\sigma_{Cxy} = G\gamma_{xy} \exp \left( -\frac{Gt}{\mu_0} \right). \quad (32)$$

Owens and Phillips Owens and Phillips (2002) define the relaxation time as "the time taken for the shear stresses in a viscoelastic material obeying the Maxwell model to reduce to $e^{-1}$ of it initial value"¹, hence $\lambda = \frac{\mu_0}{G}$.

---

¹$e^{-1} \approx 0.368$
Leaving the low-dimensional example problem, constitutive equations describing the material behavior become tensorial equations in three-dimensions. For the following constitutive equations of viscoelastic fluids it is common to rewrite the Cauchy stresses $\sigma_C$ as the sum of a pressure contribution and a contribution from the extra stress tensor $T$:

$$\sigma_C = -p 1 + T.$$  \hspace{1cm} (33)

In the case of a Newtonian fluid, $T = 2 \mu_0 \varepsilon(u)$ holds and Equation (2) is recovered from Equation (33).

The extension of the above mentioned Maxwell model to higher dimensions is the upper-convected Maxwell model. In this model, the extra stress tensor has to fulfill the following differential equation:

$$T + \lambda \left[ \partial_t T + u \cdot \nabla T - (\nabla u)T - T(\nabla u)^T \right] = 2 \mu \varepsilon(u).$$  \hspace{1cm} (34)

The term multiplied by $\lambda$ is called upper-convected derivative of $T$, which is denoted by $\overset{\triangledown}{T}$. Implemented numerically, the upper-convected Maxwell model is difficult to handle. However, the numerical difficulties can be mitigated by adding the extra stress tensor of the Newtonian fluid ($2 \mu_0 \varepsilon(u)$) to the system Behr et al. (2004).

The so obtained model is called Oldroyd-B model. In a physical experiment, this modification corresponds to adding a small quantity of Newtonian fluid to the considered liquid. As an example, one can think of a bucket with plastic melt to which some water is added. The plastic melt, one might model with the upper convected Maxwell model. The water can be model as Newtonian fluid. To approximate the behavior of the mixture, one might employ the Oldroyd-B model. In terms of spring and damper elements, the addition corresponds to an extra damper parallel to the Maxwell element (depicted in Figure 19b). The Oldroyd-B model can also be derived from a molecular model. Here, elastic dumbbells consisting of two identical masses connected with a Hookean spring are embedded in a Newtonian fluid. Starting from the equations of motion of the masses, Owens and Philips Owens and Phillips (2002) derive the constitutive equation of the Oldroyd-B fluid. The extra stress tensor in this equation reads:

$$T = 2 \mu_s \varepsilon(u) + \tau.$$  \hspace{1cm} (35)

The therein used elastic stress $\tau$ fulfills the upper-convected Maxwell equation:

$$\tau + \lambda \overset{\triangledown}{\tau} = 2 \mu_p \varepsilon(u).$$  \hspace{1cm} (36)

For a fluid modeled with the Oldroyd-B constitutive equation, the total viscosity is the sum of solvent and polymeric viscosity: $\mu_0 = \mu_s + \mu_p$. Accordingly, a solvent contribution ($2 \mu_s \varepsilon(u)$) and a polymeric contribution ($\tau$) to the extra stress tensor $T$ can be identified.
The Oldroyd-B model has been considered to be an appropriate model for polymeric solutions under restricted flow conditions. A drawback is that the shear viscosity is constant, meaning that the model cannot capture shear-thinning effects. Also, the spring connecting the masses in the elastic dumbbells is infinitely extensible. In extensional flow with a certain finite extensional rate, the springs are extended infinitely, which corresponds to an unbounded extensional viscosity. Consequently, the model fails to predict extensional flow.

A model that overcomes these drawbacks is the Giesekus model. It can be obtained from the Oldroyd-B model by adding a higher order term to the differential equation for the elastic stress:

\[
\tau + \lambda \nabla \cdot \tau + \frac{\alpha \lambda}{\mu_p} \tau^2 = 2\mu_p \varepsilon(u). \quad (37)
\]

The dimensionless Giesekus mobility parameter \( \alpha \in [0, 1] \) controls the level of anisotropy in the mobility. With \( \alpha = 0 \) the Oldroyd-B model is recovered. The Giesekus model has bounded extensional stresses and shows shear-thinning. In the paper that first introduced the model, Giesekus suggests that this constitutive equation is applicable best to model moderately concentrated solutions of flexible polymers Giesekus (1982).

### 5.2. Log-conformation formulation

At this point, the elastic stress tensor \( \tau \) is replaced by the conformation tensor \( \sigma \):

\[
\sigma = \frac{\lambda}{\mu_p} \tau + 1. \quad (38)
\]

Therewith, we can write a generic constitutive equation for the above mentioned fluid models as used similarly in Knechtges (2015); Knechtges et al. (2014):

\[
\partial_t \sigma + u \cdot \nabla \sigma - (\nabla u) \sigma - \sigma (\nabla u)^T = -\frac{1}{\lambda} P(\sigma). \quad (39)
\]

The right-hand side term \( P(\sigma) \) is an analytic function of the the conformation tensor. For the Oldroyd-B model, the function becomes \( P(\sigma) = \sigma - 1 \). The Giesekus model results in \( P(\sigma) = \sigma - 1 + \alpha(\sigma - 1)^2 \).

For the stability of the simulation, it is of crucial importance that the conformation tensor remains positive-definite throughout the simulation. This can be ensured using the log-conformation formulation proposed in Fattal and Kupferman (2004). The main idea is to replace the conformation tensor \( \sigma \) by its matrix logarithm \( \Psi \). The quantities are related by:

\[
\sigma = \exp(\Psi). \quad (40)
\]
5.3. Finite element implementation

In order to obtain a complete system of equations for the viscoelastic fluid, the constitutive equation is combined with the incompressible Navier-Stokes equations given in Equation (1):

\[ \nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega, \forall \, t \in [0, T], \]

\[ \rho(\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} - \mathbf{f}) - \nabla p - 2\mu_s \nabla \cdot \varepsilon(\mathbf{u}) - \frac{\mu_p}{\lambda} \nabla \cdot (\exp(\Psi) - 1) = 0 \quad \text{in } \Omega, \forall \, t \in [0, T]. \]

The governing equations for a Newtonian fluid are recovered by setting \( \mu_p = 0 \). The detailed derivation and implementation of the discretized weak form was done by Knechtges et al.. The two-dimensional version is published in Knechtges et al. (2014), the three-dimensional version in Knechtges (2015). The general technique is again the DSD/SST procedure, which relies on stabilization.

For the momentum equation the adjoint GLS stabilization is used (as in Section 2.4). The formulas for \( \tau_{MOM} \) remain unaltered compared to Equation (12) and (13). However, numerical experiments with the stick slip problem by Knechtges indicated that it is advantageous to only use the solvent viscosity \( \mu_s \) for the stabilization. We followed this indication for the simulations in this work and used \( \mu_{si} \) instead of \( \mu_i = \mu_{si} + \mu_{pi} \) in Equation (12) and Equation (13). Also, the constitutive equation needs to be stabilized. Here a plain Streamline Upwind/Petrov-Galerkin (SUPG) stabilization is used. The corresponding parameter reads:

\[ \tau_{CONS} = \min \left( \frac{\Delta t}{2}, \frac{2 |u|}{h} + \lambda^{-1} \right). \]

Therein, \( \lambda \) is the relaxation time of the viscoelastic fluid. \( h \) is the element length computed again as minimum singular value of the mapping from the reference element to the physical element. \( \Delta t \) is the time step size and \( |u| \) remains the velocity magnitude at the element center.

6. Viscoelastic simulations

The Oldroyd-B constitutive model has recently been used in simulations of rising bubbles in viscoelastic fluids. This gives us the opportunity to compare our results to the ones found in literature. The parameters used to specify viscoelastic simulations are given in Table 2. The density and solvent viscosity differ for the Newtonian bubble and the viscoelastic matrix fluid. The polymeric viscosity, relaxation time, and Giesekus mobility parameter only affect the viscoelastic fluid. For the different test cases, parameters will be given in cgs units or without specified units, depending on the paper from which the test case is taken.
### Table 2: Parameters in viscoelastic simulations

<table>
<thead>
<tr>
<th>Variable</th>
<th>Meaning</th>
<th>cgs units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho$</td>
<td>Density</td>
<td>g/cm$^3$</td>
</tr>
<tr>
<td>$\mu_s$</td>
<td>Solvent viscosity</td>
<td>g/(cm · s)</td>
</tr>
<tr>
<td>$\mu_p$</td>
<td>Polymeric viscosity</td>
<td>g/(cm · s)</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Relaxation time</td>
<td>s</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Giesekus mobility parameter</td>
<td></td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Surface tension coefficient</td>
<td>g/s$^2$</td>
</tr>
<tr>
<td>$g$</td>
<td>Gravitational acceleration</td>
<td>cm/s$^2$</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>Time step size</td>
<td>s</td>
</tr>
<tr>
<td>$nni$</td>
<td>Number of nodes on bubble interface</td>
<td></td>
</tr>
</tbody>
</table>

Figure 20: Two-dimensional test case "TC3-Zainali" Zainali et al. (2013): Initial geometry and parameters in cgs units

### 6.1. Two-dimensional test cases

Three two-dimensional test cases are computed to compare our results with the results recently published in literature. Of those, Pillapakkam et al. Pillapakkam and Singh (2001) as well as Damanik et al. Damanik et al. (2013) use finite elements for the spatial discretization and a level-set method to capture the phase interface. The computation of Zainali et al. Zainali et al. (2013) was done with incompressible smoothed particle hydrodynamics, while Vahabi et al. Vahabi and Sadeghy (2015) used the weakly-compressible smoothed particle hydrodynamics method.

#### 6.1.1. Test case "TC3-Zainali"

Zainali et al. Zainali et al. (2013) and Vahabi et al. Vahabi and Sadeghy (2015) have performed computations of this first viscoelastic test case with the Oldroyd-B constitutive model. As in the Newtonian benchmark computations in Section 4, the initial geometry...
of the test case is modified by shifting the initial position of the bubble upwards in the computational domain. A sketch of the problem geometry can be found in Figure 20. To characterize solvent and polymeric contributions to the Oldroyd-B fluid Zainali et al. use the ratio of retardation and relaxation time \( \beta = \frac{\lambda_2}{\lambda_1} \). The following equation relates \( \beta \) to the polymeric concentration \( c = \frac{\mu_p}{\mu_s} \), which is used by others for this purpose:

\[
\beta = \frac{\lambda_2}{\lambda_1} = \frac{\mu_s}{\mu_p + \mu_s} = \frac{1}{c + 1}.
\] (43)

With this equation, the parameters shown in the table in Figure 20 are obtained from Zainali et al. (2013).

As we execute the simulation, we observe two features characteristic for rising bubbles in viscoelastic fluids: The flow field behind the bubble shows a negative wake (Figure 21) and a cusp-like shape develops at the trailing end of the bubble (Figure 22). We observe the negative wake as a phenomenon appearing and disappearing with time. For the current simulation, it starts to develop at \( t = 0.85 \) s. Soon after, the trailing end of the bubble starts being pulled out, which is also visible in the first kink in the circularity plot in Figure 23. The vortex centers lie initially close to the trailing end of the bubble. With time, they move downwards and the velocity magnitude in the negative wake decreases. At \( t = 1.8 \) s the flow direction close to the trailing end of the bubble has changed back to "upwards", coinciding with the second kink in the circularity plot.

To build confidence in our simulation results, a grid convergence study is performed for this test case. The first order triangular mesh is successively refined by splitting the element edges in halves. This leads to 50, 100, and 200 nodes discretizing the bubble boundary. A further mesh refinement step led to node-to-node oscillations in the components of \( \Psi \), that cause a failure of the simulation after few time steps. At...
Figure 22: Test case "TC3-Zainali": Bubble shapes observed on the different meshes for Oldroyd-B (a,c,e) and Giesekus model, with $\alpha = 0.3$ (b,d,e)

Figure 23: Test case "TC3-Zainali": Rise velocity and circularity

t = 0.13 s the bubble shapes of the coarse, medium, and fine mesh (Figure 22a) as well as the rise velocity and circularity (Figure 23) seem to be grid-converged. Only after the second kink in the circularity plot at $t = 1.8 \, s$, the circularity curves begin to diverge.

Figure 22e shows a close up of the very trailing end of the bubble at $t = 0.3 \, s$. It is debatable, whether the bubble shape in this region is approximated sufficiently accurate by the coarser meshes. Interestingly, the bubble discretized with the finer mesh is extended further. The close-up also shows that the trailing end of the bubble is actually not a cusp but a rounded tip.

Comparing the results for this test case with the published data happens on a vague basis. Neither in Zainali et al. (2013), nor in Vahabi and Sadeghy (2015) rigid metrics are defined. There is also no report about measurements of the rise velocity or circularity or about any sort of refinement study. Hence, the comparison is based on the deprecated
"picture norm" (mere visual comparison Hysing et al. (2009)) of the bubble shape and negative wake at $t = 0.13 \text{s}$. Zainali et al. and Vahabi et al. disagree about the bubble shape. The bubble shape that we observe for all three meshes at $t = 0.13 \text{s}$ is somewhere in-between the very sharp shape of Vahabi et al. and the very smooth shape of Zainali et al.. The negative wake, shown in Figure 21, is an important qualitative feature, which we observe in agreement with the two previously mentioned publications.

To investigate the influence of the constitutive model, the simulations are repeated with the Giesekus model ($\alpha = 0.3$). For the first 0.05 seconds the two constitutive models come to the same results, indicating that the acceleration through gravity and viscous forces exhibited by the Newtonian solvent dominate. At later times, the bubble in the Giesekus fluid moves faster than the one in Oldroyd-B fluid. This can be attributed to the shear-thinning of a Giesekus material. Also, the bubble in Giesekus material has an even longer stretched trailing end.

6.1.2. Test case "TC4-Vahabi"

This second viscoelastic test case has previously been calculated by Pillapakkam and colleagues in 2001 Pillapakkam and Singh (2001) and in 2015 by Vahabi and colleagues Vahabi and Sadeghy (2015). The parameters listed in the table in Figure 24 are extracted form the publication by Vahabi. No-slip boundary conditions are applied on all walls. Additionally to the pictures of the bubble shape at $t = 3.5 \text{s}$, the comparison is also based on the evolution of the vertical bubble length. Our simulation is carried out on a first order triangular mesh with 92 nodes along the interface, 4716 nodes in total, and 4472 elements. These quantities are comparable to the ones of the medium mesh in the previous test case.

To test the influence of our mesh movement techniques in viscoelastic simulations, the simulation is first carried out on a rigid computational domain with $L_y = 10$ and then
repeated on a moving computational domain with $L_y = 8$ (compare Figure 24). In Figure 25 the rise velocity and circularity curves of these two simulations are labeled with "Rigid domain" and "Moving domain", respectively. The difference between the two curves is visible but not too large. To preserve a good quality of the mesh, the following simulations are carried out on a moving computational domain.

In the first step of the comparison, the bubble shapes at $t = 3.5 \text{ s}$ are considered. Figure 26a displays the shape we observe. The cusp-like trailing end of our bubble is less sharp and pronounced than the one of Vahabi et al., which is again slightly less sharp and pronounced than the one observed by Pillapakkam et al. (compare Figure 4 in Vahabi and Sadeghy (2015)).

In the second step of comparison, Figure 26b classifies our results with respect to published data for the time evolution of the vertical bubble length. The plot shows three curves. The first two curves, "FEM/LS (Vahabi)" and "WC-SPH (Vahabi)", are...
(a) Shape at $t = 4 \text{s}$, (b) Shape at $t = 4 \text{s}$, (c) Shape at $t = 4 \text{s}$,  
(d) Vertical bubble length

$L_x = 8 \text{ cm} \quad L_x = 12 \text{ cm} \quad L_x = 16 \text{ cm}$

Figure 27: Test case "TC4-Vahabi": Influence of the no-slip side wall proximity

exported from the tecplot file, which Vahabi provided. Vahabi’s publication (Vahabi and Sadeghy (2015)) concludes that the results of the two published simulations are in agreement. Except for the first data point we agree with this conclusion. The curve denoted with "FEM/IT" is our result for this test case (geometry and parameters as in Figure 24). The curve has a generally similar progression, but the vertical bubble length grows significantly slower. For $t = 1.5 \text{s}$ until $t = 3 \text{s}$ the growth is delayed by approximately $0.5 \text{s}$ to the one simulated by Vahabi.

Searching an explanation for the difference, the influence of the side wall proximity on the bubble shape and length is studied. All parameters are identical to the previous simulation, only the side walls are moved away from the bubble. This means that $L_x$ is increased from originally 8 cm to 12 cm and 16 cm. As the side walls are moved away, the bubble velocity increases and the bubble attains earlier a sharper cusp at the trailing edge. The vertical bubble length increases as $L_x$ is increased (see Figure 27).

To also investigate the influence of the boundary conditions on the side walls, the simulation with $L_x = 8 \text{ cm}$ is repeated with slip boundary conditions on the side walls. It is observed that applying slip boundary conditions has similar consequences as moving the side walls away (see Figure 28). The modified boundary condition on the side walls (slip instead of no-slip) lead to a bubble shape very similar to the shape shown in Pillapakkam and Singh (2001) and Vahabi and Sadeghy (2015) for this test case. It is depicted in Figure 28b. This agreement, however, is additionally to the modified boundary condition based on the above mentioned time shift. We show the bubble at $t = 4 \text{s}$ instead of $t = 3.5 \text{s}$.

Summarizing this test case, our results agree qualitatively but not quantitatively. A reason for the quantitative disagreement was not found.
6.1.3. Test case "TC5-Damanik"

The following test case uses geometry and parameters as defined in Damanik et al. (2013). According to private conversation with the author, he used $\beta = 0.9$, which means $c = \frac{1}{9}$ and leads to the parameter set presented in Figure 29.

Figure 30 shows four snap shots of the bubble. Our results disagree with the findings of Damanik et al.. They observe: "The cusp starts to appear very late at numerical time $t = 5$." For $t = 4$ we clearly see two inwards traveling tips at the trailing edge of the bubble. On this rather fine second order mesh spurious oscillations in $\Psi_{xx}$ appear at later times ($t > 4$). This effect may be attributable to the large time step size of $\Delta t = 0.01$. Further investigations need to be conducted to be conclusive in that regard. Repeating
the simulation with the Giesekus model with $\alpha = 0.1$ resulted in very similar bubble shapes.

In summary of the last three test cases, the following is observed: Only in test case "TC4-Vahabi" data for a quantitative comparison is available. The results disagree. In the test cases "TC4-Vahabi" and "TC3-Zainali" the bubble shows a qualitatively similar behavior. Concerning test case "TC5-Damanik" the observed bubble shapes differ significantly.

6.1.4. Test case "TC6"

The publications from which the three previously discussed two-dimensional test cases are taken do not address the question, whether a steady or at least quasi-steady state is reached, meaning the bubble moves with constant velocity and shape. As they do not have a moving computational domain, the bubble would most likely reach the upper wall of the container before reaching a steady state. To examine a bubble in steady state, we propose the following test case "TC6". The parameters are chosen such that even for long times the bubble deformation is moderate, namely the relaxation
time is set to $\lambda = 0.1$ s and the viscosity ratio between the fluids is one. The Giesekus constitutive model with $\alpha = 0.3$ is used for the viscoelastic fluid. Figure 31 gives an overview of the initial geometry and simulation parameters. A mesh with 7444 first order triangular elements is used. It has 7778 nodes in total and 126 nodes along the interface. To simulate 4 seconds, 8 CPUs were used for 5.5 hours.

As seen in previous simulations, at first, gravity accelerates the bubble, then viscous and viscoelastic forces slow the bubble down. This interplay causes an initial overshoot in the bubble velocity to about twice the final value (Figure 32). After this first overshoot, the rise velocity drops shortly under the final value before it reaches a plateau after 0.2 seconds. Compared to the rise velocity, the bubble shape deviates only slowly from the initial shape. From a circle in the beginning, the appearance changes via an egg-like shape at $t = 0.5$ s (Figure 33a) to a prolate shape with rounded tip at the trailing end at $t = 1$ s (Figure 33b). Over the next second, this shape changes only very little. Comparing Figures 33c and 33d, not only the bubble shape has reached a steady appearance, but also the distribution of $\Psi_{xx}$ around the bubble is stable. These
observations, together with the curves in Figure 32, indicate that the simulated bubble has reached a quasi-steady state at \( t = 2 \text{ s} \).

Peculiar is that the circularity curve takes approximately ten times longer to reach the stable value than the velocity curve. Moreover, the above described moderate deformation of the bubble appears without a negative wake being observed (in visual inspection of the velocity field) at any time during the simulation. This concludes the core computations and observations of test case "TC6". In the following, individual parameters of the set in Figure 31 are varied.

One parameter that appeared to be of great importance in preliminary test runs is the viscosity of the Newtonian bubble \( \mu_2 \). In physical experiments, the influence of this parameter has rarely been studied. In the following simulations \( \mu_2 \) is systematically varied, such that the viscosity ratio between the fluids \( \frac{\mu_1}{\mu_2} \) is increased from 1 over 2 and 4 to 8. Looking at the right upper plot of Figure 34, one observes that doubling the viscosity ratio yields a roughly linear increase of the bubble velocity in the first period of the simulation (up to \( t = 0.5 \text{ s} \)). The influence of the viscosity ratio on the circularity seems to be even stronger. For the range of \( t = 0 \text{ s} \) to \( t = 0.5 \text{ s} \), the circularity seems to depend linearly on the the viscosity ratio: As a consequence, doubling the viscosity ratio, approximately doubles the deviation of the circularity value from 1, i.e., ideal circularity (Figure 34 right lower plot).

It is not surprising that the bubble shapes are greatly influenced by the viscosity of the Newtonian bubble. Figure 35 shows the bubble shapes for the different viscosity ratios at \( t = 0.5 \text{ s} \) and \( t = 1 \text{ s} \). For the larger viscosity ratios, the bubble surface close to the trailing end becomes concave and a very long and narrow tail develops. This tail causes the rise velocity of bubbles with low viscosity to decrease after \( t = 0.5 \text{ s} \) (left upper plot of Figure 34). It is questionable whether the results obtained at later times for bubbles with low viscosity (Figure 35g and 35h) are correct. On the one hand, looking at the experimentally observed bubble shape in Figure 47a, they seem physical. On the other hand, the mesh quality deteriorates (despite the above discussed adaptations) and break-up – that might be physical – is not possible in our algorithm.

To the best of my knowledge, this is the first time the matrix fluid around the rising bubble is modeled with the Giesekus constitutive model. In comparison to the the Oldroyd-B model, the model is enriched by the mobility parameter \( \alpha \). To investigate the impact of this parameter, the simulation with geometry and parameters as given in Figure 31 is repeated with \( \alpha = 0.0, 0.15, 0.3, 0.6, 1.0 \). With \( \alpha = 0.0 \) the Oldroyd-B constitutive model is recovered.

The rise velocity and circularity for the different values of \( \alpha \) is displayed in Figure 36. The rise velocity reaches an approximately constant value at \( t = 1 \text{ s} \). From here the velocity curves progress in parallel. That larger values of \( \alpha \) lead to a higher rise velocity is in agreement with the expected increase of shear thinning with \( \alpha \). The circularity curves start to fan out at \( t \approx 0.6 \text{ s} \). The bubbles in a matrix fluid with \( \alpha \geq 0.3 \) attain a
Figure 34: Test case "TC6": Influence of the bubble viscosity $\mu_2$

Figure 35: Test case "TC6": Bubble shapes and $\Psi_{xx}$ at $t = 0.5$ s in the upper row and $t = 1$ s in the lower row
constant shape in the course of the simulation. The bubbles in a matrix fluid with $\alpha < 0.3$ do not. Besides in the circularity curves, this is also found in the visual comparison of bubble shapes at $t = 1 \, s$ and $t = 2 \, s$ in Figure 37. The larger $\alpha$ values cause $\Psi_{xx}$ to attain more extreme values around the trailing end of the bubble. At the same time, the end is less pulled out.
Figure 38: Initial geometry and parameters of the three-dimensional test case "TC7-Pillapakkam" Pillapakkam et al. (2007). The sphere is initially centered at $x = 1$, $y = 1, z = 2$. The subscript 1 denotes the heavier matrix fluid, 2 the bubble.

The paper introducing the Giesekus constitutive model gives $\alpha \in [0, 1]$ as parameter range Giesekus (1982). Based on the analytical solution of specific flow situations, more recent publications Daprà and Scarpi (2015); Schleiniger and Weinacht (1991) suggest that to obtain a valid solution for $\alpha > 0.5$ restrictions on the flow apply. The oscillations in $\Psi_{xx}$ observed in Figure 37h and the odd circularity curve for $\alpha = 1.0$ indicate that $\alpha = 1.0$ is not a good parameter choice for our simulation.

6.2. Three-dimensional test cases

6.2.1. Test case "TC7-Pillapakkam"

This first three-dimensional test case attempts to reproduce the results published by Pillapakkam et al. in 2007 Pillapakkam et al. (2007). The computations by Pillapakkam et al. were done with a finite element code using a level-set method to capture the phase boundary. The parameters, summarized in the table in Figure 38, correspond to the ones Pillapakkam et al. used to produce Figures 6 - 9 in the previously mentioned publication. The simulation geometry is also taken over, with the only modification, that we place the initial bubble in the middle of the computational domain as we are using a moving computational domain.

The mesh used for this computation consists of 43718 first order tetrahedrons, with 19602 nodes in total. 1635 nodes discretize the bubble interface. In order to maximize the computational accuracy with a limited number of degrees of freedom, the mesh is refined in two steps. First, the maximum side length of a tetrahedron is decreased linearly form $l_c = 0.2 \, \text{cm}$ in the far-field to $l_c = 0.0377 \, \text{cm}$ on the phase boundary. Second, the characteristic mesh length is further decreased towards the lower end of the
bubble, where $l_c = 0.00942 \text{ cm}$. Such a mesh decently resolves the large gradients in the components of $\Psi$ at the trailing end of the bubble. The computation ran in parallel on 32 CPUs and required 10 hours for 1260 time steps of 0.0005 seconds. The general bubble behavior is similar to the previously discussed two-dimensional test cases "TC3-Zainali" and "TC6".

Comparing our results to the above mentioned publication, one finds several differences: Regarding the rise velocity shown in Figure 39, we witness the maximum rise velocity of $6.78 \text{ cm/s}$ at $t = 0.02 \text{ s}$. This is about half the value of the maximum rise velocity reported in Pillapakkam et al. (2007). From there, our bubble decelerates much longer and further reaching a local minimum of $2.57 \text{ cm/s}$ at $t = 0.1175 \text{ s}$. Our second maximum is reached with $2.87 \text{ cm/s}$ at $t = 0.1975 \text{ s}$. The bubble simulated by Pillapakkam et al. has a second maximum of roughly $12.5 \text{ cm/s}$ at $t = 0.095 \text{ s}$.

Figure 40 shows the bubble shapes and the flow around the bubble at the three aforementioned extremal points of the velocity curve (first maximum, local minimum, second maximum). The two later bubble shapes differ from the ones depicted in Figure 8 and 9 in Pillapakkam et al. (2007). A common observation is the negative wake at the trailing end of the bubble at later times. A significant difference, however, is the position of the stagnation point, where the vertical flow meets and is redirected upwards and downwards. In the Figures of Pillapakkam et al., this point lies right at the end of the bubble. In contrast, we find the stagnation point shifted upwards into the bubble (Figure 40b). Likely, this shift causes the different bubble shapes and also some of the difference in rise velocity at later times. In the simulations of Pillapakkam et al., the negative wake additionally accelerates the bubble. In our simulations, the lowest part of the bubble is moved downwards with the negative wake, which decreases the total rise velocity. In Pillapakkam et al. (2007), Pillapakkam et al. claim that the bubble attains a steady state at $t = 0.15 \text{ s}$ when the simulation was stopped. In contrast, we
do not find a steady state for this parameter set and time period. Even at $t = 0.6 \, s$, the sphericity is still changing at a high rate.

The simulation is repeated using the Giesekus model for the viscoelastic fluid; the mobility parameter is chosen as $\alpha = 0.3$. As observed in previous simulations, the choice of the viscoelastic fluid model has no influence on the rise velocity or sphericity in the early phase of the simulation. At approximately $t = 0.05 \, s$, the curves start to deviate (see Figure 39). The higher rise velocity in the Giesekus fluid is explained by shear thinning effects. Bubble shapes and flow situations are similar for $\alpha = 0.3$ and $\alpha = 0.0$. Summarizing test case "TC7-Pillapakkam", it is noticed that the difference between the published results and ours prevail the similarities.

6.2.2. Test case "TC8"

This final test case can be seen as three-dimensional version of the test case "TC6". For the initial geometry and the parameter set see Figure 41. Solely the polymeric concentration was increased compared to test case "TC6". The viscosity of the viscoelastic fluid now consists of $\mu_s = 1.5$ and $\mu_p = 8.75$. A variety of previous experiments and simulations observed a large influence of the bubble volume on the bubble behavior. To study this dependence, the bubble radius is varied in six – not equidistant – steps form $r = 0.1 \, cm$ to $r = 0.3 \, cm$, thereby covering a volume range from $V = 0.00419 \, cm^3$ to $V = 0.113 \, cm^3$. This includes the volume, which Liu et al. identified experimentally as "critical volume where the velocity discontinuity occurs" Liu et al. (1995). Also, the velocity jump witnessed by Pillapakkam et al. Pillapakkam et al. (2007), was found in this volume range.
The sphere is initially centered at $x = 1$ cm, $y = 1$ cm, $z = 2$ cm. In this test case, the radius is varied from $r = 0.1$ cm to $r = 0.3$ cm. The subscript 1 denotes the heavier matrix fluid, 2 the bubble.

When varying the bubble radius, this requires to generate an individual mesh for each computation. In order to obtain a comparable approximation of the bubble interface the number of nodes on the interface is held approximately constant. It varies only from 1621 to 1645. Hence, different bubble behavior for different bubble radii is attributed to the underlying physics and not to different mesh accuracy. The simulations were executed on 32 CPUs and ran for 24 hours.

Figure 42 shows the temporal evolution of rise velocity, sphericity and normalized vertical length of the bubbles with varying radius over two seconds. The similarity of this test case’s parameters to the ones of test case "TC6" is rediscovered in very similar velocity curves. The rise velocity curves of all the differently sized bubbles progress almost in parallel. The sphericity curves on the contrary are not parallel at all. Larger bubbles deform much longer and much more.

A quantity that describes the observed kind of bubble shape evolution very precisely is the normalized vertical bubble length. It is computed as the distance of the upper-most to the lower-most point of the bubble in vertical direction, normalized with the initial bubble diameter:

$$L_v = \frac{x_{top}(3) - x_{bottom}(3)}{2r}$$

(44)

The global curves of normalized vertical bubble length and sphericity are similar. The curves of larger bubbles leave the initial value faster and further. The right lower plot of Figure 42 shows the first 0.1 $s$ of the bubble length evolution. One can see that the bubbles are first compressed vertically for a short period of time, before they are stretched. The spacing between the curves indicates a linear influence of the bubble radius on the normalized vertical bubble length over the first 0.1 s.
Figure 42: Test case "TC8": Rise velocity, sphericity, and normalized vertical bubble length

Figure 43: Test case "TC8": Bubble shapes at $t = 2\, s$, labeled with the initial bubble radius
Figure 43 displays the shapes of the bubbles with initial radius $r = 0.15\,cm$ to $r = 0.3\,cm$ at $t = 2\,s$. The smallest bubble ($r = 0.1\,cm$) is not displayed. As can be seen in the sphericity plot, its shape remains very similar to a sphere during the entire simulation. Comparing the smallest and the largest displayed bubble, we see two completely different shapes. The slightly prolate spherical shape in Figure 43a corresponds to a sphericity value of $\Psi \approx 0.9982$, while the bubble with long and narrow tail in Figure 43f has a sphericity $\Psi \approx 0.89$. As reference, a cube has a sphericity $\Psi = \frac{2\pi}{3(\frac{4}{3})^\frac{2}{3}} \approx 0.806$.

Instead, looking at the bubbles from left to right, a step in size always corresponds to a moderate change in the shape. The shapes start as almost spherical only slightly prolate for the smallest bubble. The bubbles with $r = 0.2\,cm$ and $r = 0.225\,cm$ attain an egg-like shape. The next bubble has a rounded tip, which then becomes more pronounced, before the largest bubble has a long and narrow tail.

Still, behind the smooth transitions hides a qualitative difference. All bubbles with an initial radius of $r = 0.25\,cm$ or less have a purely convex shape throughout the simulation. The bubbles with larger radii have a region where the bubble surface is concave. This distinction allows us to divide the bubbles into two groups. On the one hand bubbles with a purely convex shape reach a quasi-steady state in which the bubble shape and the bubble velocity run into a constant limit. On the other hand, bubbles that once have a concave region on their surface, develop an ever growing tail. For such bubbles the simulations did not attain a quasi-steady state.

In order to confirm that this distinction also holds for later times, the simulations of bubbles with initial radii $r = 0.25\,cm$ and $r = 0.275\,cm$ were continued for 2000 more time steps. Figure 44a depicts the normalized vertical bubble lengths observed in these
Figure 45: Test case "TC8": Rise velocity at $t = 2\, s$ (as approximation of the terminal bubble velocity) as function of the bubble radius and volume

simulations. For the first simulated second, the bubbles develop similar. In the second second, the vertical extension of the smaller bubble levels of, the larger bubble grows at approximately constant rate. Finally in the third second, the small bubble changes very little, in contrast to the larger bubble that almost doubles its length. Figure 44b and 44c compare the bubble shapes at $t = 2\, s$ and $t = 3\, s$. The shape of the bubble with $r = 0.25\, cm$ does not change visibly, while at the trailing end of the larger bubble a thin tail is pulled out.

A quantity that research activities since Liu et al. (1995) frequently focus on is the terminal rise velocity of the bubble (see Section 1). The terminal velocity is the rise velocity, which the bubble has in steady state. As in our simulations only bubbles with a radius of $r \leq 0.25\, cm$ reached a steady state, only for those a terminal velocity is directly available. For the larger bubbles, the terminal velocity is approximated with the rise velocity at $t = 2\, s$. Figure 45 shows the terminal velocity (or its approximation) as function of the bubble radius and the bubble volume. Both curves closely follow the linear regression line. Despite the above discussed dependence of the bubble shape on the bubble volume, no discontinuity in the velocity volume curve is observed. This is in agreement with recent experimental studies such as the ones published by Amirnia et al. in 2013 Amirnia et al. (2013) and Ohta et al. in 2015 Ohta et al. (2015b). Other experimental studies, e.g., the one by Liu et al. in 1995 Liu et al. (1995) and the numerical study by Pillapakkam et al. Pillapakkam et al. (2007) observe a jump in the terminal velocity.

In the following, the bubble shapes found in test case "TC8" are compared to experimentally observed ones. This is only a qualitative comparison, because the physical parameters used in our simulation differ from the experimental ones. In particular, the viscosity of the Newtonian bubble is different by several orders of magnitude (and this parameter influences the outcome of our simulation (see Section 6.1.4)).
Figure 46 displays three experimentally found bubble shapes next to three of the ones observed in test case "TC8". The experimental bubble shapes are published in Ohta et al. (2015a). A sodium acrylate polymer (SAP) solution with 0.6 wt% is used as matrix fluid. In the experimental paper, the Eötvös number is defined as:

$$E_0 = \frac{g(\rho_1 - \rho_2)d^2}{\gamma}.$$  \hspace{1cm} (45)

Therein $g$ is the gravitational acceleration, $\rho_{1/2}$ is the density of matrix fluid and bubble. The initial bubble diameter is denoted with $d$. (We label the bubbles with initial radius $r$). $\gamma$ is the surface tension coefficient. In the experiment, the different Eötvös numbers are obtained by varying the bubble volume from $V = 200 \text{ mm}^3$ to $V = 500 \text{ mm}^3$ and $V = 1600 \text{ mm}^3$. In our simulation, the bubble volumes are $V = 47.7 \text{ mm}^3$, $V = 65.4 \text{ mm}^3$ and $V = 87.1 \text{ mm}^3$. However, the Eötvös numbers lay in a similar range. Roughly, one can say that varying the bubble volume in the experiment and varying the bubble volume in our simulation led to similar changes in the bubble shape. Note that the bubble shape in Figure 46f is not a steady state bubble shape. Also, the agreement between the bubble shapes is limited to this range. The shapes observed for larger volumes in the experiment differ from the ones we find when increasing the bubble volume.

Our larger bubble shapes rather match the bubble shapes published in a different paper Ohta et al. (2015b). Here hydrophobically modified alkali-soluble emulsion polymer (HASE) solution of 1.6 wt% is the matrix fluid. The experimental bubble with an Eötvös number of $E_0 = 24$ (and a volume of $V = 600 \text{ mm}^3$) is very similar to the shape we observe for our bubble with comparable Eötvös number of $E_0 = 26.7$ (and $V = 87.1 \text{ mm}^3$) at $t = 3 \text{ s}$ (Figure 47). One can conclude that the bubble shape observed in the simulation is physical, although not numerically stable.
7. Conclusion

The aim of this study was to adapt the available mesh movement technique to better suit simulations of rising bubbles. After an initial literature study moderately deformed bubble shapes (circularity $\phi \geq 0.8$, sphericity $\Psi \geq 0.94$) – as obtained in the simulations of Hysing et al. Hysing et al. (2009), Adelsberger et al. Adelsberger et al. (2014), and Pillapakkam et al. Pillapakkam et al. (2007) – were expected. The adapted interface tracking based mesh movement technique has proven to be very well suited for moderately deformed bubbles. In Section 4, the methods have been validated with widely accepted two- and three-dimensional benchmark computations.

In cases of strongly deformed bubbles shapes – for example described in Sections 6.1.4 and 6.2.2 – the method reaches its limits. A mature version of the error adaptive mesh update (Section 3.4), edge flips, or remeshing might be able to provide a good mesh quality also for strongly deformed bubble shapes. To allow for break-up, larger modifications of the interface tracking method would be required.

The code intended for viscoelastic simulations was for the first time employed to simulate the rise of a bubble. Quantitative agreement with previously published viscoelastic simulations was not achieved. Qualitative accordance was partially observed. The detailed description of the setup and results of the various test cases in Section 6 might encourage other groups to test their implementations based on those.

The three-dimensional simulations of rising bubbles in viscoelastic fluids (Section 6.2) encountered bubble shapes similar to experimentally observed ones. A goal for future simulations is the systematic approach of a quantitative comparison with experiments,
e.g., based on terminal velocity vs. volume curves of the rising bubbles. Bridging the gap between the above described simulations, e.g., test case "TC8", and physical experiments, e.g., Ohta et al. (2015b), involves decreasing the polymeric concentration by a factor 100, increasing the density ratio by a factor 100, and increasing the viscosity ratio by a factor $1.6 \cdot 10^5$.

8. Acknowledgments

I wish to thank my supervisors Dipl.-Phys. Philipp Knechtges, Dr.-Ing. Stefanie Elgeti, and Prof. Marek Behr, Ph.D. from the Chair for Computational Analysis of Technical Systems at RWTH Aachen University for their suggestions and advise in the course of this study.
References


A. Poiseuille flow \texttt{xns} input file

```plaintext

title Poiseuille

# Governing equations
steady off
stokes on
moving on
space-time on
errorrest on

source minf

nts 100
ntsbout 1
dt 0.2

## Solver
iquadqh 7
iquadph 3
iquadoh 7
iquadgh 3

# Precondition
dprecond ilut 140 1e-4
npml rcm

# GMRES loop for fluid
ninner 100
nouter 3
epsilon 3 1e-4

# GMRES loop for mesh
ninnerx 100
nouterx 3
epsilon 2 1e-4

## Physics
material 1 viscosity 1
density 1

## Boundary conditions
# Inflow
rngdset 2 1 1 0
rngdex 2 1 1-4*(y-0.5)*(y-0.5)

# Outflow
rngdset 3 0 1 0

# Walls
noslip 4 5

## Stabilization parameters
tau_momentum franca
tau_momentum_factor 2.0
tau_continuity_factor 0.0
element_length aspectratio
relativeu on

## Newton-Raphson tolerance
epsilon 1 1e-9
nit 5
```

B. Two-dimensional test case "TC1-Hysing" \texttt{xns} input file

```plaintext

title TC1-Hysing

# Governing equations
steady off
stokes off
moving on
space-time on
periodic on
errorrest on

source minf

nts 3000
ntsbout 10
dt 0.001

## Solver
iquadqh 7
iquadph 3
iquadoh 7
iquadgh 3

# Precondition
dprecond ilut 140 1e-4
npml rcm

# GMRES loop for fluid
ninner 100
nouter 3
epsilon 3 1e-4

# GMRES loop for mesh
ninnerx 100
nouterx 3
```

```
C. Three-dimensional test case "TC2-Adelsberger" xns input file

title TC2-Adelsberger
# Governing equations
steady off
stokes off
moving on
space-time on
periodic on
errorrest on

source minf

nts 3000
dt 0.001
ntsbout 10

## Solver
#Quadrature
iquadgh 5
iquadph 7
iquadgh 5
iquadgh 7
# Precondition
rprecond ilut 140 1e-4
npml rcm
# GMRES loop for fluid
ninner 100
nouter 4
epsilon 3 1e-4
# GMRES loop for mesh
ninnerx 100
nouterx 4
epsilon 2 1e-4

# Boundary conditions
# Fluid boundary conditions
xslip 3 4
noslip 5 6

# Mesh movement
# Vertical walls
xslipx 3 4
# Bottom and Top
noslipx 5 6
rngxexp 5 2 mat2.meanvel.v*dt
rngxexp 6 2 mat2.meanvel.v*dt
# Bubble interface
slipx 7 9
surface_movement normal 9
# Default mesh parameters are used
# Output
matmeanvel 2
ac_force_scale 9 1.0
D. Test case "TC3-Zainali" xns input file

```plaintext
title TC3-Zainali

# Governing equations
steady off
stokes off
moving on
space-time on
periodic on
t
logconf

source minf

nts 600
ntsbout 1
dt 0.0005

## Solver
# Quadrature
iquadgh 7
iquadph 3
iquadoh 7
iquadgh 3

# Precondition
rprecond ilut 140 1e-4
nprml rcm

# GMRES loop for fluid
ninner 100
nouter 3
epsilon 3 1e-4

# GMRES loop for mesh
ninnerx 100
nouterx 3
epsilon 2 1e-4

# Newton-Raphson
epsilon 1 1e-10
nit 20

## Stabilization parameters
tau_momentum franca
tau_momentum_factor 2.0
tau_continuity_factor 0.0
tau_constitutive logconf
tau_constitutive_factor 1.0

## Physics
# Viscoelastic matrix fluid = material 1
material 1 viscosity 9.533
material 1 solvent_viscosity 0.717
material 1 density 1.0
material 1 reltime 0.2
# Newtonian bubble = material 2
material 2 viscosity 0.0
material 2 solvent_viscosity 1.025
material 2 density 0.1
material 2 reltime 1.0
material 1 mobility 0.3
material 2 mobility 0.3

## Initial and boundary conditions
# Fluid boundary conditions
noslip 3 4 5
rngdset 6 1 1 1 1 0

# Mesh movement
# Vertical walls
xslipx 3 4

# Bottom and Top
noslipx 5 6
rngxexp 5 2 mat2.meanvel.v*dt
rngxexp 6 2 mat2.meanvel.v*dt

# Bubble interface
slipx 7 9
surface_movement normal 9

# Default mesh parameters
# Output
matmeanvel 2
ac_force_scale 9 1.0
```
E. Test case "TC4-Vahabi" \texttt{xns} input file

```plaintext
title TC4-Vahabi
steady off
stokes off
moving on
space-time on
periodic on
tup
logconf
errorrest on
source minf
nts 4000
ntsbout 10
dt 0.001
iquadqh 7
iquadph 3
iquadoh 7
iquadgh 3
rprecond ilut 140 le-4
npmlrcm
ninner 100
nouter 3
epsilon 3 le-4
rngxexp 5 2 mat2.meanvel.v*dt
slipx 7 9
surface_movement normal 9

material 1 viscosity 100.0
material 2 viscosity 0.0
material 1 solvent_viscosity 100.0
material 2 solvent_viscosity 200.0
material 1 density 1.0
material 2 density 0.2
material 1 reltime 3.0
material 2 reltime 1.0
#material 1 mobility 0.3
#material 2 mobility 0.3

surface_tension 9 10.0
gravity 0 -980

noslip 3 4 5
rngxset 6 1 1 1 1 1 0
xslipx 3 4
noslpx 5 6
rngxexp 5 2 mat2.meanvel.v*dt
rngxexp 6 2 mat2.meanvel.v*dt
slipx 7 9
surface_movement normal 9

matmeanvel 2
ac_force_scale 9 1.0
```

F. Test case "TC5-Damanik" \texttt{xns} input file

```plaintext
title TC5-Damanik
stead off
stokes off
moving on
space-time on
periodic on
tup
logconf

source minf

nts 900
ntsbout 10
dt 0.01

iquadgh 7
iquadph 3
iquadoh 7
rprecond ilut 140 le-4
npmlrcm
ninner 100
nouter 3
epsilon 3 le-4

rngxexp 5 2 mat2.meanvel.v*dt
slipx 7 9
surface_movement normal 9

matmeanvel 2
ac_force_scale 9 1.0
```
G. Test case "TC6" xns input file

title TC6
steady off
stokes off
moving on
space-time on
periodic on
tup
logconf
source minf
nts 2000
ntsbout 10
dt 0.001
iquadgh 7
iquadph 3
iquadch 7
iquadgh 3
rprecond ilut 140 1e-4
npml rcm
ninnder 100
nouter 3
epsilon 2 1e-4
ninnderx 100
nouterx 3
epsilon 3 1e-4
nit 20
epsilon 1 1e-10
material 1 reltime 10.0
material 2 reltime 1.0
surface_tension 9 0.245
gravity 0 -9.8
xslip 3 4
noslip 6
rngdset 6 1 1 1 1 1 0
xslipx 3 4
noslipx 5 6
rngxexp 5 2 mat2.meanvel.v*dt
rngxexp 6 2 mat2.meanvel.v*dt
slipx 7 9
surface_movement normal 9
matmeanvel 2
ac_force_scale 9 1.0
### H. Test case "TC7-Pillapakkam" \texttt{xns} input file

```plaintext
title TC7-Pillapakkam
steady off
stokes off
moving on
space-time on
periodic on
tup
logconf

source minf
nts 6000
ntsbout 5
dt 0.0005
iquadgh 5
iquadph 7
iquadoh 5
iquadgh 7
rprecond ilut 140 le-4
nprml rcm

inner 120
outer 3
epsilon 2 1e-4

innerx 100
outerx 3
epsilon 3 1e-4

nit 20
epsilon 1 1e-12

tau_momentum franca
tau_momentum_factor 2.0

tau_continuity_factor 0.0
tau_constitutive logconf
tau_constitutive_factor 1.0
element_length aspectratio
relativeu on
dtingls on

material 1 viscosity 9.5
material 2 viscosity 0.0
material 1 solvent_viscosity 0.75
material 2 solvent_viscosity 1.025
material 1 density 1.0
material 2 density 0.1
material 1 retime 0.2
material 2 retime 1.0

# mobility
#material 1 mobility 0.3
#material 2 mobility 0.0
surface_tension 11 10.0
gravity 0 0 -981

noslip 3 4 5 6 7
rngdset 8 1 1 1 1 1 1 1 1 1 1 0

noslipx 3 4 5 6
rngexp 3 3 mat2.meanvel.w*dt
rngexp 4 3 mat2.meanvel.w*dt
rngexp 5 3 mat2.meanvel.w*dt
rngexp 6 3 mat2.meanvel.w*dt

noslipx 7 8
rngexp 7 3 mat2.meanvel.w*dt
rngexp 8 3 mat2.meanvel.w*dt

slipx 9 11
surface_movement normal 11

matmeanvel 2
ac_force_scale 11 1.0
```

### I. Test case "TC8" \texttt{xns} input file

```plaintext
title TC8
dt 0.0005

steady off
stokes off
moving on
space-time on
periodic on
tup
logconf

source minf
nts 6000
ntsbout 5

iquadgh 5
iquadph 7
iquadoh 5
iquadgh 7
rprecond ilut 140 le-4
nprml rcm

inner 120
outer 3
epsilon 3 1e-4
```
ninnerx 100
nouterx 3
epsilon 2 1e-4
epsilon 1 1e-12
nit 20
tau_momentum franca
tau_momentum_factor 2.0
tau_continuity_factor 0.0
tau_constitutive logconf
tau_constitutive_factor 1.0
element_length aspectratio
relativeu on
dtingls on
material 1 viscosity 8.75
material 2 viscosity 0.0
material 1 solvent_viscosity 1.5
material 2 solvent_viscosity 10.25
material 1 density 1.0
material 2 density 0.1
material 1 reltime 0.1
material 2 reltime 1.0
material 1 mobility 0.3
material 2 mobility 0.3
surface_tension 11 10.0
gravity 0 0 -981
noslip 3 4 5 6 7
rngdset 8 1 1 1 1 1 1 1 1 1 1 0
noslipx 3 4 5 6
rngxexp 3 3 mat2.meanvel.w*dt
rngxexp 4 3 mat2.meanvel.w*dt
rngxexp 5 3 mat2.meanvel.w*dt
rngxexp 6 3 mat2.meanvel.w*dt
noslipx 7 8
rngxexp 7 3 mat2.meanvel.w*dt
rngxexp 8 3 mat2.meanvel.w*dt
slipx 9 11
surface_movement normal 11
matmeanvel 2
ac_force_scale 11 1.0