

THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

A Lagrangian–Eulerian simulation method for viscoelastic flows
applied to adhesive joining

SIMON INGELSTEN

Department of Industrial and Materials Science

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, 2022

A Lagrangian–Eulerian simulation method for viscoelastic flows applied to adhesive joining
SIMON INGELSTEN

© SIMON INGELSTEN, 2022

ISBN 978-91-7905-642-1

Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie nr 5108
ISSN 0346-718X

Department of Industrial and Materials Science
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 (0)31-772 1000

Department of Computational Engineering and Design
Fraunhofer–Chalmers Research Centre for Industrial Mathematics
Chalmers Science Park
SE-412 88, Gothenburg
Sweden
Telephone: +46 (0)31-772 42 78

E-mail: simon.ingelsten@fcc.chalmers.se

Cover: Simulations of robot-carried extrusion of a viscoelastic adhesive. The adhesive is injected to the computational domain with injection cells (solid cubes) and flows on the product surface.

Chalmers Digital Printing
Gothenburg, Sweden 2022

A Lagrangian–Eulerian simulation method for viscoelastic flows applied to adhesive joining

Simon Ingelsten

Department of Industrial and Materials Science

Chalmers University of Technology

Abstract

Viscoelastic flows are important for many industrial processes, such as adhesive joining, polymer extrusion and additive manufacturing. Numerical simulations enable virtual evaluation and product realization, which can support the design phase and reduce the amount of costly physical testing. However, such applications are challenging to simulate. Thus, efficient, robust and user-friendly simulation methods are needed.

In this thesis, a Lagrangian–Eulerian simulation framework for viscoelastic flow is presented. The constitutive equation is solved at Lagrangian nodes, convected by the flow, while the momentum and continuity equations are discretized with the finite volume method. The volume of fluid method is used to model free-surface flow, with an injection model for extrusion along arbitrary nozzle paths. The solver combines an automatic and adaptive octree background grid with implicit immersed boundary conditions. In contrast to boundary-conformed mesh techniques, the framework handles arbitrary geometry and moving objects efficiently. Furthermore, novel coupling methods between the Lagrangian and Eulerian solutions as well as unique treatment of the Lagrangian stresses at the fluid-fluid interface are developed. Consequently, the resulting method can simulate the complex flows associated with the intended applications, without the need for advanced stabilization techniques.

The framework is validated for a variety of flows, including relevant benchmarks as well as industrial adhesive joining applications. The latter includes robot-carried adhesive extrusion onto a car fender as well as a hemming application. The results agree with the available experimental data. As such, the research presented in this thesis can contribute to enable virtual process development for joining applications.

Keywords: viscoelastic flow, computational fluid dynamics, immersed boundary methods, volume of fluid, adhesive joining

List of papers

This thesis is based on the research in the following papers:

I. A Lagrangian–Eulerian framework for simulation of transient viscoelastic fluid flow

S. Ingelsten, A. Mark, F. Edelvik.

Journal of Non-Newtonian fluid mechanics, 266 (2019), 20-32

II. Computationally efficient viscoelastic flow simulation using a Lagrangian–Eulerian method and GPU-acceleration

S. Ingelsten, A. Mark, K. Jareteg, R. Kádár, F. Edelvik.

Journal of Non-Newtonian fluid mechanics, 279 (2020), 104264

III. A Backwards-Tracking Lagrangian–Eulerian Method for Viscoelastic Two-Fluid Flows

S. Ingelsten, A. Mark, R. Kádár, F. Edelvik.

Applied Sciences, 11 (1) (2021), 439

IV. Simulation of viscoelastic squeeze flows for adhesive joining applications

S. Ingelsten, A. Mark, R. Kádár, F. Edelvik.

Journal of Non-Newtonian fluid mechanics, 300 (2022), 104722

V. Simulation of viscoelastic adhesive joining processes

S. Ingelsten, A. Mark, R. Kádár, F. Edelvik.

Manuscript (2022)

Other notable contributions

- **Simulating the dispensing of complex rheological fluids on arbitrary geometries using the immersed boundary method**
S. Ingelsten, J. Göhl, A. Mark, F. Edelvik
Annual Transactions of the Nordic Rheology Society (2018)
- **A numerical framework for simulation of swirled adhesive application**
S. Ingelsten, A. Mark, K. Jareteg, R. Kádár, F. Edelvik.
Annual Transactions of the Nordic Rheology Society (2019)
- **A backwards-tracking Lagrangian-Eulerian method for viscoelastic free surface flow simulation**
S. Ingelsten, A. Mark, K. Jareteg, R. Kádár, F. Edelvik.
Annual Transactions of the Nordic Rheology Society (2020)
- **Thixotropy of cellulose nanocrystal suspensions**
M. Fazilati, S. Ingelsten, S. Wojno, T. Nypelö, R. Kádár
Journal of Rheology, 65 (5) (2021), 1035-1052

Acknowledgments

I would like to thank my main supervisor Roland Kádár and my co-supervisors Andreas Mark and Fredrik Edelvik. I am very grateful for all the support, all the fruitful discussions and for believing in my ideas. These have all been vital for making my ideas come to life as well as for writing the papers and this thesis.

I would also like thank to my colleagues at the department of Computational Engineering and Design at the Fraunhofer-Chalmers Centre. Thank you for all the interesting discussions on programming, numerics and fluid mechanics as well as for creating a friendly and welcoming work environment.

I would like to thank our colleagues at RISE IVF in Mölndal for the good collaboration over the years and for supporting with experimental results, with special mentions to Per-Johan Wahlborg, Fredrik Wandebäck and Ola Albinsson.

Thank you my beloved family, for always supporting and believing in me. Thank you all my friends for making life fun and interesting.

Finally, thank you Louise for the endless love and support.

This work has been supported in part by the Swedish Governmental Agency for Innovation Systems, VINNOVA, through the FFI Sustainable Production Technology program, and in part by the Production Area of Advance at Chalmers University of Technology. The research has been partly carried out in a Centre for Additive Manufacturing – Metal (CAM2) in a joint project financed by Swedish Governmental Agency of Innovation Systems (Vinnova), coordinated by Chalmers University of Technology. The support is gratefully acknowledged.

Contents

Nomenclature	xi
1 Introduction	1
1.1 Scope of this thesis	3
1.2 Outline of thesis	5
1.3 Background of research project	5
1.4 Software framework	6
1.5 Viscoelasticity	6
1.6 Viscoelastic flow	8
1.6.1 The Lagrangian and Eulerian frames of reference	8
1.6.2 The momentum and continuity equations	8
1.6.3 Convected derivatives	10
1.6.4 Constitutive models	11
1.6.5 Conformation tensor	14
1.6.6 Multiple modes	14
1.6.7 Normal stress differences	15
1.6.8 The Reynolds, Weissenberg and Deborah numbers	15
1.7 Numerical stability	16
1.8 Motivation for the chosen approach	18
2 Lagrangian–Eulerian framework	21
2.1 Governing equations	21
2.2 Eulerian finite volume solver	22
2.2.1 Immersed boundary method	23
2.2.2 Volume of fluid method	23
2.2.3 Injection model	24

CONTENTS

2.3	Lagrangian-Eulerian formulation	26
2.4	Forwards-tracking method	27
2.4.1	Distribution of Lagrangian nodes	27
2.4.2	ODE systems	28
2.4.3	Radial basis function interpolation	29
2.5	Backwards-tracking method	30
2.6	Viscoelastic contribution to momentum equation	32
2.7	ODE solver	33
2.7.1	Forwards-tracking method	33
2.7.2	Backwards-tracking method	34
2.8	Implementation	34
2.8.1	ODE solver	35
2.8.2	Radial basis function interpolation	35
2.8.3	Numerical stability	36
3	Results and discussion	37
3.1	Single-phase flow	37
3.2	Computational performance	41
3.3	Free-surface flow	46
3.4	Adhesive joining flows	51
3.4.1	Adhesive extrusion	51
3.4.2	Joining operations	56
4	Conclusions	65
	References	69

Nomenclature

Acronyms

ALPM	Adaptive Lagrangian particle method
BDF	Backward differentiation formula
BLPM	Backward-tracking Lagrangian particle method
BSD	Both sides diffusion
CPU	Core processing unit
FEM	Finite element method
FENE	Finitely extensible nonlinear elasticity
FSI	Fluid-structure interaction
FVM	Finite volume method
GPU	Graphics processing unit
HWNP	High Weissenberg number problem
LCR	Log-conformation representation
LPM	Lagrangian particle method
MAC	Marker-and-cell (method)
ODE	Ordinary differential equation
PDPS	Positive definiteness preserving scheme
PTT	Phan Thien Tanner (model)
RBF	Radial basis function
SIMPLEC	Semi-implicit method for pressure-linked equations – consistent
SIMPLE	Semi-implicit method for pressure-linked equations
SPD	Symmetric positive definite
SPTT	Simplified Phan Thien Tanner (model)
SRCR	Square-root conformation representation
UCM	Upper-convected Maxwell (model)

Nomenclature

VOF Volume of fluid (method)

Roman Symbols

$a_{n,i}$	BDF formula function value coefficient
A	Gordon-Schowalter derivative velocity gradient terms
a_s	Convected derivative slip parameter
b_n	BDF formula time derivative coefficient
B	Log-conformation representation symmetric tensor
B	RBF interpolation polynomial contribution matrix
b	Square-root conformation tensor
c	Conformation tensor
d	Number of spatial dimensions
F	Body force
\mathcal{Y}	Conformation tensor equation scalar-valued function
f	General scalar function
f	RBF interpolation right hand side vector
\mathcal{F}	Constitutive equation model function
G	Elastic modulus
g	ODE Right hand side vector
G_k	Constitutive equation right hand side (ODE form) for k th stress mode
H	Conformation tensor equation tensor-valued function
I	Unit tensor
K_i	Runge-Kutta coefficient vector i
l	Extensibility in FENE constitutive models
L	General second order tensor
M	RBF interpolation matrix
M	Square-root conformation representation antisymmetric tensor
N_c	Number of close points in RBF interpolation
N_{cells}	Number of cells in Eulerian grid
N_{loc}	Number of local time steps in ODE solver
N_m	Number of viscoelastic stress modes
n_{max}	Maximum number of Lagrangian nodes in a subcell
n_{split}	Number of subcells in each direction for node distribution
P	RBF interpolation polynomial term in

p	Pressure
\mathbf{R}	Matrix of conformation tensor eigenvectors
\mathbb{R}	Real numbers
\mathbf{S}	Strain rate tensor
t	Time
\mathbf{u}	Velocity vector
\mathbf{u}_{app}	Injection nozzle velocity
\mathbf{u}_{flow}	Injection velocity based on flow rate
\mathbf{u}_{inj}	Total injection velocity
\mathbf{v}	RBF interpolation polynomial coefficient vector
V	Volume
V_{inj}	Injected volume
V_{nom}	Nominal injected volume
\dot{V}	Injection volume flow rate
\dot{V}_{corr}	Injection volume flow rate correction
\dot{V}_{nom}	Nominal injection volume flow rate
\mathbf{w}	RBF interpolation weight vector
\mathbf{x}	Position vector
\mathbf{x}_0	Material point
\mathbf{y}	ODE solution vector

Greek Symbols

α	Fluid volume fraction
α_G	Dimensionless parameter in Giesekus constitutive model
α_i	General linear viscoelastic model coefficient
$\alpha_{\text{lim},1}$	Viscoelastic subset larger volume fraction threshold
$\alpha_{\text{lim},2}$	Viscoelastic subset smaller volume fraction threshold
β	Viscosity ratio
β_i	General linear viscoelastic model coefficient
ε	Dimensionless parameter in PTT constitutive models
$\varepsilon_{\text{neigh}}$	Node distribution neighborhood tolerance
η	Polymeric viscosity
η_t	Total viscosity
γ	Scalar strain

Nomenclature

γ	Strain tensor
$\dot{\gamma}_c$	Characteristic shear rate
Λ	Fluid color function
λ	Relaxation time
λ_r	Retardation time
μ	Solvent viscosity
μ_a	Artificial viscosity
μ_c	Characteristic viscosity
Ω	Computational domain
$\Omega_{v,1}$	Viscoelastic subset (larger volume fraction)
$\Omega_{v,2}$	Viscoelastic subset (smaller volume fraction)
$\mathbf{\Omega}$	Log-conformation representation antisymmetric tensor
ϕ	General quantity
ψ	Radial basis function
ρ	Density
σ	Scalar stress
$\boldsymbol{\sigma}$	Cauchy stress tensor
$\boldsymbol{\tau}$	Viscoelastic stress
$\boldsymbol{\tau}_d$	Deviatoric stress
$\boldsymbol{\Theta}$	Log-conformation tensor
ξ	Gordon-Schowalter convected derivative parameter
ξ_l	Injection flow rate correction lower limit factor
ξ_r	Injection flow rate correction relaxation factor
ξ_s	Injection flow rate correction smoothing factor
ζ_s	RBF interpolation scaling parameter

Dimensionless Numbers

De	Deborah number
Re	Reynolds number
Wi	Weissenberg number

Other

$\mathbf{0}$	Zero matrix or vector
--------------	-----------------------

Chapter 1

Introduction

Viscoelastic fluid flows are of great importance for many industrial processes. This includes adhesive joining, polymer extrusion, additive manufacturing and seam sealing. In many cases the process repeatedly needs to be adjusted for new products and optimized in terms of the product quality, production cycle time and material consumption. This typically involves substantial manual work, including time-consuming and expensive physical testing.

Numerical simulation tools enable virtual verification and product realization, which can aid a reduction of the physical testing required. They can also provide valuable insight early in design phase through virtual concept evaluation. As such, user-friendly and efficient simulation tools can offer a process to optimize the production cost and contribute to sustainability. However, many industrial applications involve complex geometry, moving or deforming objects and viscoelastic free-surface flow, making them challenging to simulate numerically.

A variety of numerical methods for viscoelastic flow can be found in the literature. An extensive review on the subject was recently published by Alves *et al.* (2021). Commonly, the governing equations are discretized in the Eulerian frame of reference with the finite volume method (FVM) (Alves *et al.*, 2001, 2003; Oliveira *et al.*, 1998; Pimenta & Alves, 2017) or the finite element method (FEM) (Baaijens *et al.*, 1995; Hulsen *et al.*, 2005). The Eulerian description is suitable for transport equations with diffusion, including transport of momentum, heat and mass. However, viscoelastic constitutive equations are typically hyperbolic and lack physical diffusion. Specialized high-order discretization schemes may therefore be necessary to minimize numerical

diffusion (Alves *et al.*, 2003). Thus, the Lagrangian frame of reference constitutes an appealing alternative.

Some Lagrangian techniques involves deformation of the computational mesh. As such, repeated re-meshing is necessary to avoid excessive distortion. The Lagrangian method by Rasmussen & Hassager (1995) and the split Lagrangian–Eulerian method by Harlen *et al.* (1995), which are both based on FEM, constitute such examples. However, a Lagrangian–Eulerian approach does not necessarily require a deforming mesh. In the Lagrangian particle method (LPM) by Halin *et al.* (1998), the constitutive equation was integrated along the trajectories of massless particles in the flow, while the momentum and continuity equations were solved on a stationary mesh with FEM. The element integrals were evaluated through polynomial approximation of the Lagrangian stress solution. Thus, a minimum of three particles per element was required for the polynomials to be admissible. In the adaptive Lagrangian particle method (ALPM) (Gallez *et al.*, 1999), adaptive creation and deletion of particles were introduced. In LPM and ALPM a relatively large amount of particles was required to achieve numerical stability. Later, the improved backward-tracking Lagrangian particle method (BLPM) (Wapperom *et al.*, 2000) was proposed.

For viscoelastic free-surface flow, different numerical techniques are conceivable. An early example is the mixed FEM method to simulate the viscoelastic die swell effect by Crochet & Keunings (1982). A recent example with a similar approach was reported by Spanjaards *et al.* (2019). An extensively used method is the marker-and-cell (MAC) method (de Paulo *et al.*, 2007; Oishi *et al.*, 2011, 2008; Tomé *et al.*, 2002, 2008, 2010, 2012). In this method, a finite difference discretization is used for the governing equations and the free surface is tracked by marker particles. The method has been proven useful for certain flows, although it is typically implemented for uniform spatial discretization.

A popular method in FVM frameworks, for Newtonian as well as viscoelastic free-surface flow, is the volume of fluid (VOF) method. VOF is a diffuse-interface method, in which the local fluid volume fraction is convected with an algebraic or a geometric scheme. A single set of governing equations is solved, for which the fluid properties are locally averaged with the volume fraction. A sharp fluid interface may be reconstructed from the computed volume fraction field. Examples of viscoelastic VOF applications include Habla *et al.* (2011), Comminal *et al.* (2018) and Niethammer *et al.* (2019). A

different approach related to VOF was reported by Bonito *et al.* (2006). They performed a predictor step, convecting the solution in the cells of a cubic lattice in a Lagrangian manner. A finite element method was then used to correct the solution and solve the flow in the liquid region.

In addition to the above mentioned approaches, other free-surface techniques include e.g. front-tracking methods (Izbassarov & Muradoglu, 2015), phase-field methods (Zografos *et al.*, 2020) and level-set methods (Pillapakam & Singh, 2001; Stewart *et al.*, 2008).

A majority of the discussed methods assume a boundary-conformed computational mesh, i.e. the flow geometry is described through the external boundary conditions of the domain. Alternatively, structured grids are used, limiting the description to non-complex geometries. While a boundary-conformed description is useful for many applications, including detailed study of benchmark flows, it lacks flexibility for certain types of flow. Typically, a substantial manual effort is required to generate a mesh, particularly in complex geometries. Flows with moving objects are particularly challenging, requiring advanced techniques with deforming, overlapping or sliding meshes. Alternative numerical methods are therefore needed, which can efficiently handle complex geometry, moving objects and viscoelastic free surface flow. Immersed boundary methods, as for example used in this thesis, can be a suitable approach.

1.1 Scope of this thesis

In this thesis a new Lagrangian–Eulerian framework for numerical simulation of viscoelastic flow is presented. The viscoelastic constitutive equation is solved along the trajectories of Lagrangian fluid nodes, which are convected by the flow. The fluid momentum and continuity equations are solved with an Eulerian finite volume method, discretized on an automatic and adaptive octree grid. Boundary conditions on solid objects in the computational domain are imposed with implicit immersed boundary conditions and free-surface flow modeled with the volume of fluid method.

The combination of the immersed boundary method with the automatic grid generation enables efficient handling of arbitrary geometry and moving objects with minimal user-input compared to boundary-conformed mesh methods. An injection model de-

scribes the inflow of a viscoelastic fluid from a moving nozzle, which enables simulation of adhesive extrusion along industrial robot paths.

The main scientific contribution of this thesis consists of the development, implementation and validation of two novel versions of a Lagrangian–Eulerian algorithm to solve the viscoelastic constitutive equation within the immersed boundary finite volume-framework. This includes the development of unique, tailored methods to couple the Lagrangian and the Eulerian fields.

In the forwards-tracking method, a robust radial basis function method is used for unstructured interpolation of the viscoelastic stress to the Eulerian grid, such that the viscoelastic term in the momentum equation can be integrated in the cells. In contrast to similar methods with local polynomial approximation of the stress, the interpolation results in a method which is versatile and insensitive to the local number of Lagrangian nodes. In the backwards-tracking method, the viscoelastic stress is stored in a structured, staggered-like arrangement in the Eulerian grid nodes. The viscoelastic term in the momentum equation is therefore integrated numerically over the Eulerian cell surface directly from the Lagrangian solution. Hence, no additional interpolation is required.

The Lagrangian–Eulerian framework is intended for viscoelastic free-surface flow and utilizes a unique treatment of the Lagrangian stress solution near the fluid-fluid interface. The Lagrangian solver is localized to the viscoelastic phase and the influence of large velocities in the low-viscous Newtonian phase is reduced. Consequently, the computational performance as well as the robustness is enhanced.

To the best of the authors knowledge, this type of Lagrangian–Eulerian formulation in a finite volume framework has not previously been reported. The same holds for the treatment of the coupling between the Lagrangian and Eulerian solution fields and the treatment of the Lagrangian solution in free-surface flow. As will be shown in the thesis, these properties combined results in a numerical method which can simulate the complex viscoelastic flows in the intended applications, including moving, deforming objects and extrusion from a moving adhesive nozzle. Furthermore, no advanced numerical stabilization or enhanced stress-velocity coupling scheme, such as those similar to Rhie-Chow interpolation used in Eulerian finite volume methods, is necessary.

1.2 Outline of thesis

In the following sections of this chapter, the background of the research project is presented and the software framework is summarized. This is followed by an introduction to viscoelasticity and an overview of viscoelastic flow, with focus on numerical simulation. The chapter is concluded with a motivation for the choice of numerical method.

In the second chapter, the numerical framework is presented in detail with focus on the Lagrangian algorithms developed to solve the viscoelastic constitutive equation and the coupling to the Eulerian momentum equation. The chapter is concluded with a few remarks on the corresponding software implementation.

In the third chapter the numerical studies conducted with the Lagrangian–Eulerian framework are presented. This includes relevant benchmark flow simulations, a computational performance assessment and a numerical case study of a viscoelastic adhesive joining application. Finally, in the fourth chapter the thesis is concluded and the outlook on future research is discussed.

1.3 Background of research project

The research presented in this thesis is part of the long-term aim at the Fraunhofer–Chalmers Research Centre for Industrial Mathematics to develop efficient and user-friendly numerical tools for simulation and optimization of various production processes. An important focus area is the adhesive joining process. This includes the flows involved in the adhesive extrusion and joining, as well robot programming and optimization. The aim is to enable virtual product realization through simulation. Other focus areas include surface treatment processes, such as a seam sealing, electrostatic spray painting and oven curing, as well as heat transfer processes for electronics cooling and additive manufacturing. The research has been carried out in publicly funded projects as well as industrial collaboration projects.

In previous research, shear-thinning viscosity models, e.g. the Carreau model, proved insufficient to correctly predict the adhesive bead profile in certain adhesive extrusion applications. Furthermore, the use of viscoelastic constitutive models was identified

as a promising approach. Thus, an accurate, robust and efficient numerical method suitable for such flows was desired.

1.4 Software framework

The Lagrangian–Eulerian framework presented in this thesis is implemented in the in-house software platform IPS IBOFlow[®] (IPS IBOFlow, 2022), an incompressible fluid flow solver implemented in C++ at the Fraunhofer-Chalmers Centre for Industrial Mathematics in Gothenburg, Sweden. The key features of the solver are the use of implicit immersed boundary methods (Mark & van Wachem, 2008; Mark *et al.*, 2011) and the automatic and adaptive octree discretization. Prior to the current work, the software has been employed for simulation of free-surface flow of shear thinning fluids in automotive seam sealing (Edelvik *et al.*, 2017; Mark *et al.*, 2014), adhesive application (Svensson *et al.*, 2016) and 3D-bioprinting (Göhl *et al.*, 2018), as well as for fluid-structure interaction (FSI) applications (Svenning *et al.*, 2014) and conjugated heat transfer (Andersson *et al.*, 2018; Mark *et al.*, 2013; Nowak *et al.*, 2020). Within these areas, the software framework is utilized as a research platform but is also commercially available.

1.5 Viscoelasticity

Mechanical elements constitute a useful means of illustrating viscoelastic material properties. The discussion in this section is based on Barnes *et al.* (1989) and Bird *et al.* (1987b). A Hookean spring and a viscous damper, shown in Figure 1.1, serve as models of ideal elastic and viscous materials.



Figure 1.1: Mechanical elements.

The spring obeys Hooke’s law of elasticity

$$\sigma = G\gamma, \tag{1.1}$$

where σ is the stress in the spring, γ the strain and G the elastic shear modulus, in case of shear strain. The damper obeys Newton's viscosity law

$$\sigma = \eta \dot{\gamma}, \quad (1.2)$$

where η is the viscosity and $\dot{\gamma}$ is the strain rate or, in case of simple shear strain, the shear rate.

Viscoelastic models can be generated through combination of the spring and damper elements. Combination in series yields the Maxwell element and in parallel the Kelvin-Voigt element, both shown in Figure 1.2.

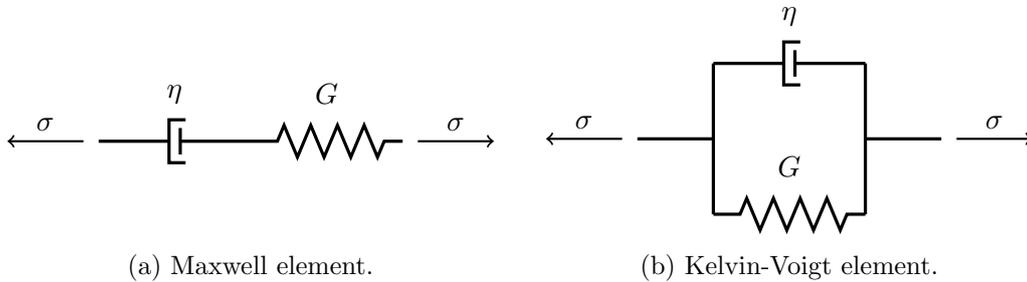


Figure 1.2: Viscoelastic elements.

For the Maxwell element, the stresses in the damper and the spring are equal, while the total strain is the sum of the element strains. This results, after some manipulation, in

$$\lambda \dot{\sigma} + \sigma = \eta \dot{\gamma}, \quad (1.3)$$

where $\lambda = \eta/G$ is the Maxwell relaxation time. The corresponding equation for the Kelvin-Voigt element reads

$$\sigma = G\gamma + \eta \dot{\gamma}, \quad (1.4)$$

which can be derived by assuming equal strains in the damper and the spring and that the total stress is the sum of the respective stresses.

The Maxwell and Kelvin-Voigt elements represent models of linear viscoelasticity. Generally, all linear viscoelastic models obey linear ordinary differential equations (ODE) on the form (Barnes *et al.*, 1989)

$$\left(1 + \sum_{i=1}^n \alpha_i \frac{d^i}{dt^i}\right) \sigma = \left(\beta_0 + \sum_{k=1}^m \beta_k \frac{d^k}{dt^k}\right) \gamma, \quad (1.5)$$

where d^i/dt^i is the i th order time derivative, and $\{\alpha_i\}_{i=1}^n$ and $\{\beta_k\}_{k=0}^m$ are constants. For example, by letting $\alpha_1 = \lambda$ and $\beta_1 = \eta$ be the only nonzero constants, the Maxwell model is obtained. For $\beta_0 = G$ and $\beta_1 = \eta$ the Kelvin-Voigt model is obtained. Generalization of linear viscoelastic models to constitutive equations for viscoelastic flow is discussed in Section 1.6.4.

1.6 Viscoelastic flow

Relevant aspects of viscoelastic flow are discussed in this section, including the governing equations as well different frames of reference and convected derivatives.

1.6.1 The Lagrangian and Eulerian frames of reference

This thesis considers a Lagrangian–Eulerian formulation of the governing equations for viscoelastic flow. An introduction to the corresponding frames of reference is therefore given.

In the Lagrangian frame of reference, properties are described in material points which move with the material due to deformation and flow. In contrast, in the Eulerian frame of reference properties are described at fixed spatial locations. The Lagrangian and Eulerian frames are also denoted the material and spatial descriptions, respectively. Let $\mathbf{x}(\mathbf{x}_0, t)$ be the location of the material point at time t , such that $\mathbf{x}(\mathbf{x}_0, t_0) = \mathbf{x}_0$. Let $\frac{\partial\phi}{\partial t}$ denote the rate of change of ϕ at a fixed location $\mathbf{x}(\mathbf{x}_0, t)$ at some time t . By application of the chain rule

$$\frac{d}{dt}(\phi(\mathbf{x}(\mathbf{x}_0, t), t)) = \frac{\partial\phi}{\partial t} + \frac{\partial\phi}{\partial\mathbf{x}} \cdot \frac{\partial\mathbf{x}}{\partial t} = \frac{\partial\phi}{\partial t} + \mathbf{u} \cdot \nabla\phi = \frac{D\phi}{Dt}, \quad (1.6)$$

where \mathbf{u} is velocity. Here, $D\phi/Dt$ is the Lagrangian time derivative of ϕ and $\partial\phi/\partial t$ the Eulerian time derivative.

1.6.2 The momentum and continuity equations

For continuum flow, the conservation of mass is described by the continuity equation (Schlichting & Gersten, 2000)

$$\frac{\partial\rho}{\partial t} + \nabla \cdot (\rho\mathbf{u}) = 0, \quad (1.7)$$

where ρ is the mass density. For incompressible flows, the density is constant and (1.7) reduces to the incompressible continuity equation

$$\nabla \cdot \mathbf{u} = 0, \quad (1.8)$$

i.e. the velocity field is divergence-free.

Conservation of linear momentum is described by Cauchy's first law of continuum mechanics (Truesdell & Rajagopal, 1999)

$$\rho \frac{D\mathbf{u}}{Dt} = \nabla \cdot \boldsymbol{\sigma} + \mathbf{F}, \quad (1.9)$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor and \mathbf{F} a body force, including e.g. gravity. Furthermore, Cauchy's second law of continuum mechanics states

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^T, \quad (1.10)$$

i.e. the stress tensor is symmetric. For fluid flow, the stress can be expressed as

$$\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau}_d, \quad (1.11)$$

where p is the isotropic pressure $p = -\frac{1}{3}\text{Tr}(\boldsymbol{\sigma})$, \mathbf{I} is the identity tensor and $\text{Tr}(\boldsymbol{\sigma})$ the trace of $\boldsymbol{\sigma}$. The tensor $\boldsymbol{\tau}_d$ is called the deviatoric stress (Schlichting & Gersten, 2000).

For a Navier-Stokes fluid, or Newtonian fluid, the deviatoric stress for incompressible flow reads (Truesdell & Rajagopal, 1999)

$$\boldsymbol{\tau}_d = 2\mu\mathbf{S}, \quad (1.12)$$

where μ is the viscosity and \mathbf{S} is the strain rate tensor, i.e. the symmetric part of the velocity gradient, and reads

$$\mathbf{S} = \frac{1}{2} (\nabla\mathbf{u} + \nabla\mathbf{u}^T). \quad (1.13)$$

Insertion of (1.11) and (1.12) into the Cauchy momentum equation (1.9) yields the well-known incompressible Navier-Stokes equation

$$\rho \left(\frac{\partial\mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla\mathbf{u} \right) = -\nabla p + \mu\nabla^2\mathbf{u} + \mathbf{F}. \quad (1.14)$$

In (1.14) the right hand side is expressed in the Eulerian frame of reference. Furthermore, it has been used that for incompressible flow with constant viscosity μ

$$\nabla \cdot (2\mu\mathbf{S}) = \mu\nabla^2\mathbf{u}. \quad (1.15)$$

For a viscoelastic fluid, the deviatoric stress can be expressed as the Newtonian counterpart with an additional term as

$$\boldsymbol{\tau}_d = 2\mu\mathbf{S} + \boldsymbol{\tau}, \quad (1.16)$$

where $\boldsymbol{\tau}$ is the viscoelastic stress. The corresponding momentum equation reads

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \nabla \cdot \boldsymbol{\tau} + \mathbf{F}. \quad (1.17)$$

In the context of polymeric fluids, the Newtonian viscosity $\mu \geq 0$ is commonly referred to as the solvent viscosity. The viscoelastic stress $\boldsymbol{\tau}$ is described by a constitutive equation, discussed in Section 1.6.4.

1.6.3 Convected derivatives

Replacing the scalar stress σ and strain rate $\dot{\gamma}$ by their tensorial counterparts $\boldsymbol{\tau}$ and $2\mathbf{S}$, e.g. in the Maxwell model equation (1.3), may appear reasonable to construct a constitutive equation for viscoelastic flow. However, it can be shown that the result is not frame invariant (Morozov & Spagnolie, 2015). For a constitutive equation to be admissible, it must be independent of the frame of reference in which it is expressed, including time-dependent frames (Barnes *et al.*, 1989). This is sometimes expressed in terms of a material objectivity condition, see for example Lodge (1974).

To derive formulation principles for admissible constitutive equations, Oldroyd (1950) introduced a convected coordinate system which is embedded and deforms with the material, commonly referred to as the covariant base vectors. Another set of convected base vectors are the contravariant base vectors, which are perpendicular to the material surfaces (Bird *et al.*, 1987b). Frame-invariant convected time derivatives can be identified by expressing the stress tensor $\boldsymbol{\tau}$ in terms of the convected base vectors and applying the time derivative. After some manipulation, the covariant frame leads to the upper-convected time derivative, which for a second order tensor \mathbf{L} reads

$$\overset{\nabla}{\mathbf{L}} = \frac{D\mathbf{L}}{Dt} - \mathbf{L} \cdot \nabla \mathbf{u} - \nabla \mathbf{u}^\top \cdot \mathbf{L}, \quad (1.18)$$

which is common in viscoelastic constitutive models. The contravariant frame leads to the lower-convected time derivative,

$$\overset{\Delta}{\mathbf{L}} = \frac{D\mathbf{L}}{Dt} - \mathbf{L} \cdot \nabla \mathbf{u}^\top - \nabla \mathbf{u} \cdot \mathbf{L}. \quad (1.19)$$

Linear combinations of the upper- and lower-convected derivatives also yield frame-invariant time derivatives, as

$$\square \mathbf{L} = \left(\frac{1 + a_s}{2} \right) \nabla \mathbf{L} + \left(\frac{1 - a_s}{2} \right) \overset{\Delta}{\mathbf{L}}, \quad (1.20)$$

where $a_s \in [-2, 2]$ is a slip parameter. A different form is the Gordon-Schowalter derivative $\square \mathbf{L}$ (Larson, 1988)

$$\square \mathbf{L} = \frac{D\mathbf{L}}{Dt} - \mathbf{L} \cdot \nabla \mathbf{u} - \nabla \mathbf{u}^\top \cdot \mathbf{L} + \xi (\mathbf{L} \cdot \mathbf{S} + \mathbf{S} \cdot \mathbf{L}), \quad (1.21)$$

where $\xi \in [0, 2]$ is related to the slip parameter a_s .

Finally, higher order convected derivatives may be constructed by successive application of the convected derivative, such that (Bird *et al.*, 1987b)

$$\mathbf{L}_{[n+1]} = (\mathbf{L}_{[n]})_{[1]}, \quad (1.22)$$

where $\mathbf{L}_{[n]}$ is the n th order convected derivative of \mathbf{L} and $\mathbf{L}_{[1]} = \square \mathbf{L}$.

1.6.4 Constitutive models

An intuitive method to construct admissible constitutive equations is through generalization of linear viscoelasticity, introducing tensorial quantities and convected time derivatives, resulting in so-called quasi-linear constitutive models. For such models, experimental data is necessary for an appropriate choice of convected derivative (Bird *et al.*, 1987b). A general such model reads

$$\boldsymbol{\tau} + \sum_{i=1}^n \alpha_i \boldsymbol{\tau}_{[i]} = \beta_0 \boldsymbol{\gamma} + 2 \sum_{k=1}^m \beta_k \mathbf{S}_{[k-1]}. \quad (1.23)$$

Compared to the linear viscoelastic model (1.5), σ is replaced by $\boldsymbol{\tau}$ and γ by the strain tensor $\boldsymbol{\gamma}$. Furthermore, the property $\boldsymbol{\gamma}_{[1]} = \square \boldsymbol{\gamma} = \nabla \mathbf{u} + \nabla \mathbf{u}^\top = 2\mathbf{S}$ (Bird *et al.*, 1987b) is used.

As discussed in Section 1.5, the Maxwell element equation is obtained with the nonzero constants $\alpha_1 = \lambda$ and $\beta_1 = \eta$ in (1.5). Similarly, the same choice of constants in (1.23) with the upper-convected derivative yields the upper-convected Maxwell (UCM) model

$$\lambda \overset{\nabla}{\boldsymbol{\tau}} + \boldsymbol{\tau} = 2\eta \mathbf{S}, \quad (1.24)$$

1. INTRODUCTION

where λ is the relaxation time and η the polymeric viscosity. Another constitutive model is the Oldroyd-B model, which may be obtained from (1.23) with the nonzero constants $\alpha_1 = \lambda$, $\beta_1 = \eta_0$ and $\beta_2 = \eta_0 \lambda_r$, resulting in the constitutive equation (Larson, 1988)

$$\lambda \overset{\nabla}{\boldsymbol{\tau}} + \boldsymbol{\tau} = 2\eta_0 \left(\mathbf{S} + \lambda_r \overset{\nabla}{\mathbf{S}} \right), \quad (1.25)$$

where η_0 is the total viscosity and λ_r is the retardation time. An alternative formulation of the Oldroyd-B model results from the decomposition

$$\boldsymbol{\tau}_d = 2 \frac{\lambda_r}{\lambda} \eta_0 \mathbf{S} + \boldsymbol{\tau}, \quad (1.26)$$

where $\boldsymbol{\tau}$ has the constitutive equation

$$\lambda \overset{\nabla}{\boldsymbol{\tau}} + \boldsymbol{\tau} = 2 \left(1 - \frac{\lambda_r}{\lambda} \right) \eta_0 \mathbf{S}. \quad (1.27)$$

In this form, the Oldroyd-B model corresponds to the UCM model with the nonzero solvent viscosity $\mu = \frac{\lambda_r}{\lambda} \eta_0$ and the polymeric viscosity $\eta = \left(1 - \frac{\lambda_r}{\lambda} \right) \eta_0$.

Quasi-linear models, including the UCM and Oldroyd-B models, impose no upper limit on the physical stretching of polymer molecules. This may lead to unbounded normal stresses. Nonlinear models, however, can provide a physically correct description for a wider range of flows. Admissible constitutive equations can be constructed through combination of convected derivatives with empirical expressions (Bird *et al.*, 1987b). The Giesekus model is such an example, obtained by adding a quadratic term to the UCM model (Morozov & Spagnolie, 2015),

$$\lambda \overset{\nabla}{\boldsymbol{\tau}} + \boldsymbol{\tau} + \frac{\alpha_G \lambda}{\eta} \boldsymbol{\tau} \cdot \boldsymbol{\tau} = 2\eta \mathbf{S}, \quad (1.28)$$

where $\alpha_G \in [0, 1/2]$ is a dimensionless parameter.

A different approach is to derive constitutive equations from molecular theory (Bird *et al.*, 1987a). A well-known example is the Phan Thien Tanner (PTT) model (Thien & Tanner, 1977), derived from network theory assuming non-affine motions between the strands, reading

$$\lambda \overset{\square}{\boldsymbol{\tau}} + \left(1 + \frac{\varepsilon \lambda}{\eta} \text{Tr}(\boldsymbol{\tau}) \right) \boldsymbol{\tau} = 2\eta \mathbf{S}, \quad (1.29)$$

where ε is a dimensionless parameter. A version with an exponential nonlinear term was later proposed (Phan-Thien, 1978), such that

$$\lambda \overset{\square}{\boldsymbol{\tau}} + \exp \left(\frac{\varepsilon \lambda}{\eta} \text{Tr}(\boldsymbol{\tau}) \right) \boldsymbol{\tau} = 2\eta \mathbf{S}. \quad (1.30)$$

If the Gordon-Schowalter derivative $\overset{\square}{\boldsymbol{\tau}}$ is reduced to the upper convected derivative $\overset{\nabla}{\boldsymbol{\tau}}$, the model is also referred to as the simplified PTT (SPTT) model.

In the finitely extensible nonlinear elasticity (FENE) models (Herrchen & Öttinger, 1997), the viscoelastic fluid is considered a dilute solution of nonlinear dumbbells, i.e. beads connected by nonlinear springs as shown in Figure 1.3.

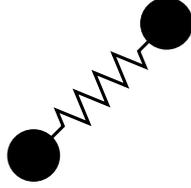


Figure 1.3: Dumbbell.

Different constitutive equations can be obtained for the continuum-level stress, depending on the choice of closure approximation. The Peterlin approximation yields the FENE-P constitutive equation (Bird *et al.*, 1987b)

$$Z(\text{Tr}(\boldsymbol{\tau}))\boldsymbol{\tau} + \lambda\overset{\nabla}{\boldsymbol{\tau}} - \lambda\left(\boldsymbol{\tau} - \left(1 - \frac{2}{2+l}\right)\frac{\eta}{\lambda}\mathbf{I}\right)\frac{D \ln Z}{Dt} = -2\left(1 - \frac{b}{b+2}\right)\eta\mathbf{S}, \quad (1.31)$$

where l is a model parameter related to the maximum dumbbell extension and the function Z reads

$$Z(\text{Tr}(\boldsymbol{\tau})) = 1 + \frac{3}{l}\left(1 - \frac{\lambda}{3\eta}\text{Tr}(\boldsymbol{\tau})\right). \quad (1.32)$$

Another closure approximation by Chilcott & Rallison (1988) results in the FENE-CR constitutive equation

$$\lambda\overset{\nabla}{\boldsymbol{\tau}} + \left(\frac{l^2 + \frac{\lambda}{\eta}\text{Tr}(\boldsymbol{\tau})}{l^2 - 3}\right)\boldsymbol{\tau} = 2\eta\left(\frac{l^2 + \frac{\lambda}{\eta}\text{Tr}(\boldsymbol{\tau})}{l^2 - 3}\right)\mathbf{S}. \quad (1.33)$$

In addition to those discussed, other models derived from molecular theory include the Pom-Pom model (McLeish & Larson, 1998) for branched polymer melts and the Rolie-Poly model (Likhtman & Graham, 2003). A different class of models are integral models, in which the deformation history is accounted for through a memory function, e.g. the K-BKZ model (Mitsoulis, 2013).

1.6.5 Conformation tensor

The viscoelastic state of a material may be expressed in terms of the molecular configuration state through the conformation tensor

$$\mathbf{c} = \frac{\lambda}{\eta} \boldsymbol{\tau} + \mathbf{I}. \quad (1.34)$$

In physical terms, the conformation tensor is the second moment of the dimensionless end-to-end vector of polymer chains (Morozov & Spagnolie, 2015). Many constitutive models may be formulated for \mathbf{c} on the form (Chen *et al.*, 2013)

$$\overset{\nabla}{\mathbf{c}} = \frac{1}{\lambda} \mathcal{Y}(\mathbf{c}) \mathcal{H}(\mathbf{c}), \quad (1.35)$$

where $\mathcal{Y}(\mathbf{c})$ is a scalar-valued function and $\mathcal{H}(\mathbf{c})$ a tensor-valued function. The conformation tensor equation is commonly used in stability enhancement techniques, discussed in Section 1.7. In Table 1.1 the expressions for \mathcal{Y} and \mathcal{H} are listed for different constitutive models.

Model	$\mathcal{Y}(\mathbf{c})$	$\mathcal{H}(\mathbf{c})$
UCM/Oldroyd-B	1	$\mathbf{I} - \mathbf{c}$
Giesekus	1	$\mathbf{I} - \mathbf{c} - \alpha_G(\mathbf{I} - \mathbf{c})$
(S)PTT (linear form)	$1 + \varepsilon(\text{Tr}(\mathbf{c}) - 3)$	$\mathbf{I} - \mathbf{c}$
(S)PTT (exponential form)	$\varepsilon(\text{Tr}(\mathbf{c} - 3))$	$\mathbf{I} - \mathbf{c}$
FENE-P	1	$\mathbf{I} - \mathbf{c} / (1 - \text{Tr}(\mathbf{c}) / l^2)$
FENE-CR	$(1 - \text{Tr}(\mathbf{c}) / l^2)^{-1}$	$\mathbf{I} - \mathbf{c}$

Table 1.1: Functions $\mathcal{Y}(\mathbf{c})$ and $\mathcal{H}(\mathbf{c})$ in (1.35) for different constitutive models.

1.6.6 Multiple modes

In general, the viscoelastic stress tensor may be considered to be a sum of multiple stress modes as

$$\boldsymbol{\tau} = \sum_k^{N_m} \boldsymbol{\tau}_k, \quad (1.36)$$

where $\boldsymbol{\tau}_k$ is the k th stress mode and N_m the number of modes. Each stress mode $\boldsymbol{\tau}_k$ is described by a constitutive equation with a unique set of parameters. Thus, viscoelastic fluids with multiple relaxation times can be modeled.

1.6.7 Normal stress differences

The normal stress behavior is important for viscoelastic flow. This is illustrated by simple shear flow, shown in Figure 1.4. Here, $u_{2,3} = 0$ and $u_1 > 0$ varies linearly in the x_2 -direction. Thus, the only nonzero component of the velocity gradient is $\dot{\gamma} = \partial u_1 / \partial x_2$. For this case, if any other component of the deviatoric stress tensor $\boldsymbol{\tau}_d$ than the shear stress τ_{12} is nonzero, the flow is by definition non-Newtonian (Barnes *et al.*, 1989).

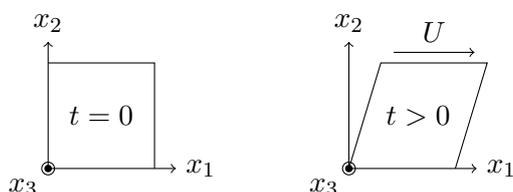


Figure 1.4: Simple shear flow.

The normal stresses τ_{11} , τ_{22} and τ_{33} cannot be measured directly in rheometry experiments. However, the differences

$$N_1 = \tau_{11} - \tau_{22}, \quad (1.37)$$

$$N_2 = \tau_{22} - \tau_{33}, \quad (1.38)$$

can be measured (Larson, 1999). N_1 and N_2 are called the first and second normal stress differences, respectively.

1.6.8 The Reynolds, Weissenberg and Deborah numbers

The Reynolds number quantifies the ratio of inertial to frictional forces, and reads (Schlichting & Gersten, 2000)

$$\text{Re} = \frac{\rho U L}{\mu_c}, \quad (1.39)$$

where U and L are characteristic velocity and length scales, respectively and μ_c a characteristic viscosity.

The Deborah and Weissenberg numbers quantify the effects of elasticity in a flow. While their definitions are similar for many flows, they should be interpreted differently. The Deborah number was originally proposed by Reiner (1964) as

$$\text{De} = \frac{\text{Characteristic material time scale}}{\text{Observation time scale}}. \quad (1.40)$$

A small De corresponds to fluid-like behavior while a large De corresponds to solid-like behavior. For practical reasons, a commonly used definition is

$$\text{De} = \frac{t_{\text{relaxation}}}{t_{\text{process}}}, \quad (1.41)$$

where $t_{\text{relaxation}}$ is the material relaxation time and t_{process} the time scale of the deformation process. For certain flows which are steady in the Lagrangian sense, e.g fully developed pipe and channel flows or steady simple shear, the deformation time scale is infinite. Hence, the Deborah number is zero for such flows (Poole, 2012).

The Weissenberg number Wi was identified by White (1964) as the number quantifying the ratio of viscoelastic to viscous forces. Following Poole (2012), the elastic forces in steady simple shear flow of an UCM fluid are characterized by the first normal stress difference $N_1 = \tau_{11} - \tau_{22} = 2\lambda\eta\dot{\gamma}_c^2$ and the viscous forces by the shear stress $\tau_{12} = \eta\dot{\gamma}_c$, where $\dot{\gamma}_c$ is a characteristic shear rate. The Weissenberg number is then

$$\text{Wi} = \frac{N_1}{\tau_{12}} = \frac{2\lambda\eta\dot{\gamma}_c^2}{\eta\dot{\gamma}_c} = 2\lambda\dot{\gamma}_c = 2\lambda\frac{U}{L}. \quad (1.42)$$

From (1.42) it is clear that Wi may be interpreted as the ratio of viscoelastic to viscous forces. It is remarked that in some cases the Deborah and Weissenberg numbers are equal or are related through a geometrical factor. However, this not is the case for all flows (Poole, 2012).

1.7 Numerical stability

In viscoelastic flow simulations, numerical instabilities and convergence issues can arise for moderate Weissenberg or Deborah numbers, commonly referred to as the high Weissenberg number problem (HWNP) (Keunings, 2000). Convergence issues at limiting values of Wi or De arise either due to model limitations or numerical approximation errors. Some semi-analytical evidence of limiting values exist. However, what can be interpreted as a limiting Wi or De for the discrete problem could arguably instead stem

from numerical artifacts (Owens & Phillips, 2002). Numerical evidence suggests that the origin of the numerical breakdown is a loss of resolution near large stress gradients. In regions where the stresses grow exponentially, the commonly used polynomial approximations constitute inappropriate representations of the stress profiles (Fattal & Kupferman, 2004).

Different remedies to the HWNP have been proposed. A comparison between different stabilization methods was conducted by Chen *et al.* (2013). As a general remark, a pointed out by Keunings (2000), schemes or modifications to implicitly or explicitly smooth the stress profiles could effectively change the problem being solved. Thus, while numerical stability is achieved, the obtained solution corresponds to a different problem than the original one.

A straightforward means of increasing the numerical stability is to enhance the ellipticity of the problem through artificial diffusion in the momentum equation, referred to as both sides diffusion (BSD). The corresponding momentum equation reads

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \nabla \cdot \mathbf{u} \right) - 2(\mu + \mu_a) \nabla \cdot \mathbf{S} = -\nabla p - 2\mu_a \nabla \cdot \mathbf{S} + \nabla \cdot \boldsymbol{\tau} + \mathbf{F}, \quad (1.43)$$

where μ_a is the artificial viscosity. In the continuum sense, the additional terms $2\mu_a \nabla \cdot \mathbf{S}$ cancel each other. However, in the discretized equations, the left hand side term is treated implicitly and the right hand side term explicitly. Consequently, numerical diffusion which enhances the numerical stability is introduced. In light of the remark by Keunings (2000), BSD may falsely diffuse the solution in time and is preferably avoided for transient simulations, noted by Xue *et al.* (2004). On the other hand, the stabilization method can be suitable for steady flow applications.

The conformation tensor \mathbf{c} is symmetric and positive definite (SPD) and thus have real eigenvalues and the diagonalization $\tilde{\mathbf{c}} = \mathbf{R}^T \mathbf{c} \mathbf{R}$, where \mathbf{R} consists of the eigenvectors of \mathbf{c} (Alves *et al.*, 2021). As such, the constitutive equation may be reformulated for a tensor with improved numerical properties, e.g. inherent preservation of the positive definiteness of \mathbf{c} or reduction of steep stress gradients. A popular such method is the Log-conformation representation (LCR) proposed by Fattal & Kupferman (2004, 2005). As the name suggests, an the constitutive equation is transformed to an equation for $\boldsymbol{\Theta} = \log(\mathbf{c})$, reading

$$\frac{D\boldsymbol{\Theta}}{Dt} - (\boldsymbol{\Omega}\boldsymbol{\Theta} - \boldsymbol{\Theta}\boldsymbol{\Omega}) - 2\boldsymbol{\mathcal{B}} = \frac{1}{\lambda} \mathcal{Y}(e^{\boldsymbol{\Theta}}) e^{-\boldsymbol{\Theta}} \mathcal{H}(e^{\boldsymbol{\Theta}}), \quad (1.44)$$

where $\boldsymbol{\Omega}$ is an antisymmetric tensor and $\boldsymbol{\mathcal{B}}$ a tensor which commutes with \mathbf{c} and $\text{Tr}(\boldsymbol{\mathcal{B}}) = 0$. The tensors $\boldsymbol{\Omega}$ and $\boldsymbol{\mathcal{B}}$ stem from a decomposition of the velocity gradient $\nabla\mathbf{u}$, assuming $\nabla \cdot \mathbf{u} = 0$. The functions \mathcal{Y} and \mathcal{H} are those listed in Table 1.1 for the conformation tensor equation (1.35).

A formulation conceptually similar to LCR is the square-root conformation representation (SRCR) by Balci *et al.* (2011). The square-root conformation tensor \mathbf{b} is defined such that $\mathbf{c} = \mathbf{b} \cdot \mathbf{b}$. Balci *et al.* (2011) originally stated the equations for \mathbf{b} for the Oldroyd-B and FENE-P models, which for a general constitutive reads (Palhares Junior *et al.*, 2016)

$$\frac{D\mathbf{b}}{Dt} = \mathbf{b} \cdot \nabla\mathbf{u}^T + \boldsymbol{\mathcal{M}} \cdot \mathbf{b} + \frac{1}{2\lambda} \mathcal{Y}(\mathbf{b}^2)\mathbf{b}^{-1} \cdot \mathcal{H}(\mathbf{b}^2). \quad (1.45)$$

The tensor $\boldsymbol{\mathcal{M}}$ is antisymmetric and can be calculated explicitly.

The LCR and SRCR both reduce steep gradients in the solution variables and preserve the positive definiteness of the conformation tensor, enhancing numerical stability for strongly elastic flows. Certain challenges for accuracy at high De remain, however, due to demanding requirements of grid refinements Alves *et al.* (2021). A generalized mathematical framework to transform the constitutive equation in terms of any continuous, invertible and matrix transformation, was formulated by Afonso *et al.* (2012), called the kernel-conformation method. For appropriate choices of the matrix transformation, e.g. the LCR and SRCR equations can be recovered.

Finally, a stabilization scheme which should be mentioned is the positive definiteness preserving scheme (PDPS) proposed by Stewart *et al.* (2008), in which the discretization scheme for the constitutive equation preserves the positive definiteness \mathbf{c} per design.

It is noted that the stabilization methods discussed are commonly applied to Eulerian solution algorithms. However, reformulations such as e.g. the LCR and SRCR are feasible for Lagrangian or semi-Lagrangian frameworks as well.

1.8 Motivation for the chosen approach

The current chapter is concluded with a motivation for the Lagrangian–Eulerian approach chosen for the developed numerical method.

Eulerian finite volume discretization is a well-established method for viscous flow as well as for transport of heat and mass. For the viscoelastic constitutive equation, as previously discussed, Eulerian discretization may require specialized high-order schemes

to minimize the numerical diffusion (Alves *et al.*, 2003). For collocated grids, enhanced stress-velocity coupling schemes are necessary, similar to Rhie-Chow interpolation (Alves *et al.*, 2021). Furthermore, stabilization through e.g. LCR is frequently utilized for numerical stability. Thus, a Lagrangian algorithm to solve the constitutive equation is a viable approach.

The solution of the Lagrangian constitutive equation is straightforward to parallelize. Consequently, the resulting algorithm can be executed in parallel on the core processing unit (CPU) or the graphics processing unit (GPU). In comparison, for Eulerian discretization of the constitutive equation, three or six large coupled linear systems must be solved, respectively for 2D and 3D. Furthermore, for free-surface flow applications in which the viscoelastic phase only occupies a small part of the computational domain, the Lagrangian algorithm only needs to be executed in those areas.

The Lagrangian–Eulerian framework presented in this thesis extends the Eulerian finite volume solver described in Section 1.4 with a Lagrangian method to solve the viscoelastic constitutive equation. In contrast to some Lagrangian methods, e.g. Rasmussen & Hassager (1995) or Harlen *et al.* (1995), the computational grid is stationary and therefore not subject to deformation. The current approach has certain similarities to the FEM-based Lagrangian particle methods by Gallez *et al.* (1999); Halin *et al.* (1998); Wapperom *et al.* (2000). Important differences compared to their work include the use of FVM discretization, the automatic octree grid and the immersed boundary method. This combination enables efficient handling of complex flow geometry and moving or deforming objects, particularly compared to boundary-conformed mesh descriptions. Thus the method is suitable for e.g. adhesive joining applications. In addition, the coupling between the Lagrangian and the Eulerian solution variables through interpolation requires considerably fewer Lagrangian nodes per cell for stable results and does not automatically fail if the number of nodes is too small. Finally, in contrast to the referenced Lagrangian methods, the current numerical framework supports simulation of free-surface flow with the VOF method.

In summary, the chosen approach aims to utilize the advantages of the Lagrangian constitutive equation and the Eulerian momentum and continuity equations. The resulting framework can describe complex industrial applications in an efficient and user-friendly manner, without the need for advanced stabilization techniques, enhanced stress-velocity coupling schemes or a boundary-conformed computational mesh.

1. INTRODUCTION

Chapter 2

Lagrangian–Eulerian framework

In this chapter, the numerical method developed is described, with main focus on the Lagrangian algorithms for the viscoelastic stress. Two versions of the Lagrangian–Eulerian method have been developed. The forwards-tracking method was developed mainly during earlier part of the project [Paper I-II], while the improved backwards-tracking method was developed during the later part [Paper III-V].

This chapter is structured as follows. The governing equations are stated, followed by a description of the finite volume method used to solve the Eulerian transport equations. The Lagrangian methods to solve the constitutive equation and couple the results to the Eulerian momentum equation are then described in detail. The chapter is concluded with a short discussion on the software implementation of the numerical framework.

2.1 Governing equations

The viscoelastic fluid flow is described by the incompressible momentum equation (1.17) and the continuity equation (1.8), stated again here for completeness

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \nabla \cdot \boldsymbol{\tau} + \mathbf{F}, \quad (1.17)$$

$$\nabla \cdot \mathbf{u} = 0. \quad (1.8)$$

For N_m viscoelastic stress modes, the constitutive equation is assumed to have the general form

$$\lambda_k \tau_k^\square + \mathcal{F}_k(\tau_k) \tau_k = 2\eta_k \mathbf{S}, \quad k = 1, \dots, N_m, \quad (2.1)$$

where \mathcal{F}_k is a scalar-valued function, defined by the choice of constitutive model.

2.2 Eulerian finite volume solver

The momentum and continuity equations are discretized with the finite volume method on a collocated octree background grid. A computational domain is defined as the Cartesian box

$$\Omega = \begin{cases} I_x \times I_y, & d = 2 \\ I_x \times I_y \times I_z, & d = 3 \end{cases} \quad (2.2)$$

where I_x, I_y and I_z are intervals in the respective coordinate directions and d the number of spatial dimensions considered. The discretization of the domain Ω is defined through an equidistant subdivision of the domain in the respective coordinate direction, i.e. a base cell size. The grid may be anisotropic, i.e. the cell sizes may vary between the coordinate directions. The grid is automatically generated and can be adaptively refined, e.g. around moving objects, in narrow channels and at fluid interfaces. Refinements are generated by recursively dividing cells in the coordinate directions to the desired level, illustrated in Figure 2.1 for $d = 2, 3$.

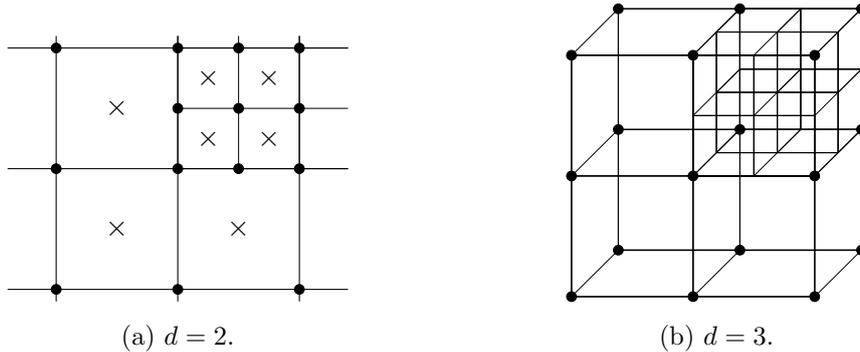


Figure 2.1: Examples of octree grids with one refinement level, with cell centers (\times) and grid nodes (\bullet).

The momentum equation (1.17) provides transport equations for the velocity components, which are coupled to the pressure field through the pressure gradient. However,

the incompressible continuity equation (1.8) does not include the pressure. A pressure-velocity coupling algorithm is thus necessary to couple the fields. In this work the Semi-implicit method for pressure-linked equations – consistent (SIMPLEC) (Doormaal & Raithby, 1984) is used, which is based on the original SIMPLE method by Patankar (1980).

2.2.1 Immersed boundary method

Interior objects in the computational domain Ω are represented by surface triangulations. Boundary conditions from such objects are imposed using the mirroring immersed boundary method (Mark & van Wachem, 2008; Mark *et al.*, 2011). As the linear system for the momentum equation is assembled, the velocity field is implicitly mirrored across the boundary surface such that the prescribed boundary condition is satisfied for the converged solution. The mirroring technique introduces a fictitious flow field inside the object, which is replaced by the object velocity in all flux calculations to ensure zero mass flow across the boundary.

2.2.2 Volume of fluid method

Two-fluid flow is modeled with the volume of fluid method. A color function $\Lambda \in \{0, 1\}$ is defined such that

$$\Lambda = \begin{cases} 1, & \text{In the viscoelastic phase} \\ 0, & \text{In the Newtonian phase} \end{cases} . \quad (2.3)$$

The discrete counterpart to Λ is the fluid volume fraction $\alpha \in [0, 1]$, which constitutes the local volume average of Λ in a control volume. The transport of α is described by the convection equation Tryggvason *et al.* (2011)

$$\frac{\partial \alpha}{\partial t} + \mathbf{u} \cdot \nabla \alpha = 0. \quad (2.4)$$

By definition, α is the relative volume of viscoelastic fluid in a cell. Thus, for $\alpha = 1$ the cell is completely occupied by the viscoelastic fluid and for $\alpha = 0$ by air. For $0 < \alpha < 1$, the cell is intersected by the interface between the two phases.

The transport equation (2.4) is discretized on the Eulerian grid with the finite volume method using the compact CICSAM scheme (Ubbink & Issa, 1999). This

2. LAGRANGIAN–EULERIAN FRAMEWORK

compressible convective scheme is developed to minimize the numerical diffusion of the interface.

Although two fluids may be present, the flow is described by a single set of momentum, continuity and constitutive equations. The local contributions from the respective phases are accounted for through the volume averaging

$$\phi = \alpha\phi_v + (1 - \alpha)\phi_N, \quad (2.5)$$

where ϕ_v is the property of the viscoelastic phase and ϕ_N that of the Newtonian phase. The averaging (2.5) is applied to μ , ρ , η , λ and $\boldsymbol{\tau}$.

For practical reasons, two viscoelastic subsets of the computational domain Ω are defined as

$$\Omega_{v,1}(t) = \{\mathbf{r} \in \Omega : \alpha(t, \mathbf{r}) \geq \alpha_{\text{lim},1}\}, \quad (2.6)$$

$$\Omega_{v,2}(t) = \{\mathbf{r} \in \Omega : \alpha(t, \mathbf{r}) \geq \alpha_{\text{lim},2}\}, \quad (2.7)$$

where $\alpha_{\text{lim},1} > \alpha_{\text{lim},2}$ and, thus, $\Omega_{v,1} \subseteq \Omega_{v,2}$. The viscoelastic constitutive equation is only solved at points $\mathbf{r} \in \Omega_{v,1}$. Other points are considered to lie outside the viscoelastic phase. Furthermore, the contribution from points $\mathbf{r} \notin \Omega_{v,2}$ is excluded from the velocity gradient in the constitutive equation.

The first condition, defined by $\Omega_{v,1}$, reduces the problem size as the constitutive equation is only solved in the viscoelastic phase, in contrast to the whole domain. The set $\Omega_{v,2}$ is mainly introduced to reduce the influence on the constitutive equation from large velocities in the Newtonian phase, which often has low viscosity and density. Thus, the requirement to resolve all velocity scales in the Newtonian phase near the interface is relaxed. In all results presented in this thesis, the threshold values $\alpha_{\text{lim},1} = 0.1$ $\alpha_{\text{lim},2} = 0.01$ are used.

2.2.3 Injection model

Certain applications involve continuous inflow of a viscoelastic fluid from a moving nozzle, as illustrated in Figure 2.2. An injection model is therefore included in the numerical framework, with the objective to inject the viscoelastic fluid to the computational for the given process conditions.

The adhesive injection is carried out with the following steps:

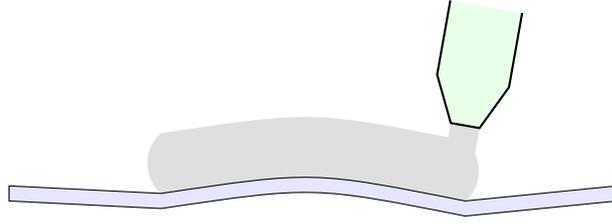


Figure 2.2: Application with a moving nozzle.

- The grid is refined in the injection zone (current nozzle location)
- Injection cells which approximate the nozzle geometry are identified.
- Adhesive is injected by modifying the volume fraction α in the injection cells.

The procedure is illustrated in Figure 2.3. Note that the nozzle geometry is not included as an immersed boundary object in the simulation. The nozzle geometry is accounted for solely by the injection cells.

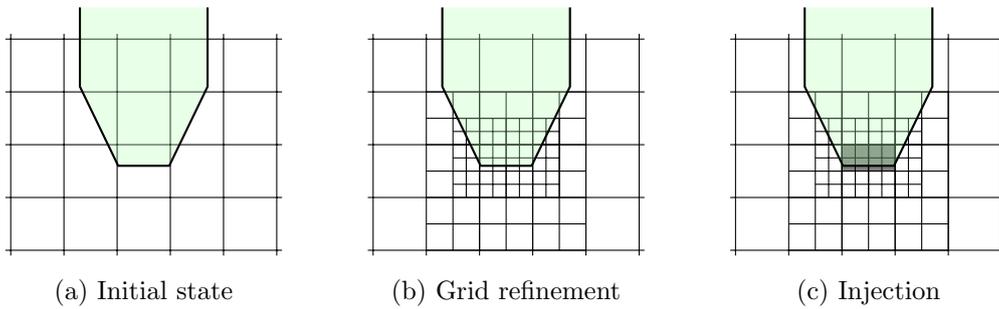


Figure 2.3: Schematic of injection step.

The velocity condition imposed for the injection cells is based on the volume flow rate and the applicator velocity and is treated as an immersed boundary condition. The velocities in the injection cells are set to

$$\mathbf{u}_{\text{inj}} = \mathbf{u}_{\text{flow}} + \mathbf{u}_{\text{app}}, \quad (2.8)$$

where \mathbf{u}_{flow} is the velocity based on the volume flow rate and \mathbf{u}_{app} is the nozzle movement velocity. To account for discrepancies between the nominal nozzle geometry and

the discrete approximation by the injection cells, a flow rate correction is calculated as

$$\dot{V}_{\text{corr}} = \max \left(\xi_s \frac{V_{\text{nom}} - V_{\text{inj}}}{\Delta t}, \xi_l \dot{V}_{\text{nom}} \right), \quad (2.9)$$

where V_{inj} is the injected volume and V_{nom} the nominal volume with respect to the flow rate history. $\xi_s \in (0, 1]$ is introduced to smoothen flow variations over time and $\xi_l \in (0, 1)$ to impose a lower limit on the flow rate. The volume flow rate used for the injection is calculated as

$$\dot{V} = \xi_r \left(V_{\text{nom}} + \dot{V}_{\text{corr}} \right) + (1 - \xi_r) \dot{V}_{\text{old}}, \quad (2.10)$$

where $\xi_r \in (0, 1]$ is a relaxation factor and \dot{V}_{old} the flow rate in the previous time step. In this work, $\xi_s = 0.2$, $\xi_l = 0.1$ and $\xi_r = 0.1$ have been found to be suitable values and are used in the simulations.

2.3 Lagrangian-Eulerian formulation

The constitutive equation (2.1) is expressed as

$$\frac{D\boldsymbol{\tau}_k}{Dt} = \frac{2\eta_k}{\lambda_k} \mathbf{S} - \frac{\mathcal{F}(\boldsymbol{\tau}_k)}{\lambda_k} \boldsymbol{\tau}_k + \mathbf{A}(\boldsymbol{\tau}_k), \quad k = 1, \dots, N_m, \quad (2.11)$$

where $\mathbf{A}(\boldsymbol{\tau}_k)$ expands to

$$\mathbf{A}(\boldsymbol{\tau}_k) = \boldsymbol{\tau}_k \cdot \nabla \mathbf{u} + \nabla \mathbf{u}^T \cdot \boldsymbol{\tau}_k - \xi (\boldsymbol{\tau}_k \cdot \mathbf{S} + \mathbf{S} \cdot \boldsymbol{\tau}_k), \quad k = 1, \dots, N_m. \quad (2.12)$$

For a fluid element, (2.11) constitutes an ODE system for the local viscoelastic stress. However, the fluid element is convected by the flow and (2.11) involves the velocity gradient $\nabla \mathbf{u}$. Thus, to solve (2.11) for the element its trajectory $\mathbf{x}(t)$ is required. An ODE system for the trajectory is expressed as

$$\dot{\mathbf{x}}(t) = \mathbf{u}(t, \mathbf{x}), \quad (2.13)$$

$$\dot{\boldsymbol{\tau}}_k(t, \mathbf{x}) = G_k(\boldsymbol{\tau}_k, \nabla \mathbf{u}(t, \mathbf{x})), \quad k = 1, \dots, N_m, \quad (2.14)$$

where G_k expands to the right hand side of (2.11).

Two methods to solve (2.13) and (2.14) throughout the flow and add the contribution to the momentum equation (1.17) have been developed. The forwards-tracking

method was developed in the earlier part of the research project. The backwards-tracking method was later developed as an improved version of the method, introducing structured storage of the Lagrangian stresses and eliminating the need for unstructured interpolation. Consequently, the backwards-tracking improved the general robustness, as well as the support for refined Eulerian grids and free-surface flow. The two methods are described in the following sections.

2.4 Forwards-tracking method

In the first method developed, denoted the forwards-tracking method, the ODE systems (2.13) and (2.14) are solved simultaneously forwards in time for Lagrangian nodes distributed throughout the viscoelastic flow. As the resulting stress field is stored in an unstructured set of points, radial basis function (RBF) (Iske, 2004) interpolation is used to calculate the stresses at the cell centers of the Eulerian grid. The procedure can be summarized as:

1. Distribute/redistribute Lagrangian nodes
2. Convect nodes and compute updated stresses by solving ODE systems
3. Interpolate viscoelastic stresses to Eulerian grid
4. Calculate viscoelastic contribution to the discretized momentum equation

The different steps are discussed in detail below.

2.4.1 Distribution of Lagrangian nodes

At initialization, Lagrangian nodes are distributed in the domain through a subdivision of the Eulerian cells. Each cell is split into n_{split} smaller segments in each coordinate direction and a Lagrangian node is placed in each sub-volume, as shown in Figure 2.4.

Since the Lagrangian nodes move with the flow, the distribution is subject to variation. The distribution is therefore maintained through addition and deletion of nodes. A node is added if the neighborhood of a cell sub-volume does not contain a node. The neighborhood is defined as the box centered on the sub-volume and with the side $(1 + \varepsilon_{\text{neigh}})\Delta x/n_{\text{split}}$, where $\varepsilon_{\text{neigh}} = 0.1$ in the current work, as shown in Figure 2.5. In this work $\varepsilon_{\text{neigh}} = 0.1$ is used, which has been found a suitable value. The

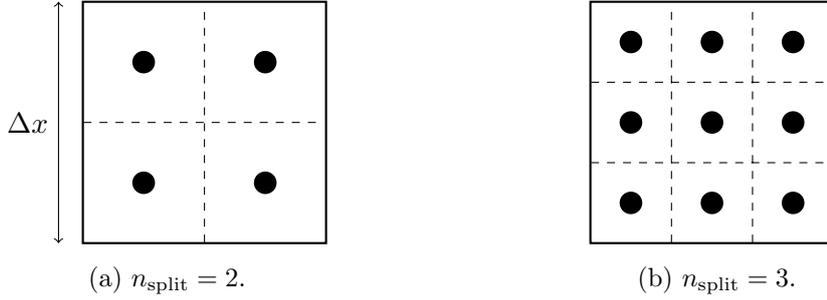


Figure 2.4: Subdivision of a two-dimensional Eulerian cell for Lagrangian node distribution.

stress of the newly added node is interpolated from the surrounding nodes with the RBF method, discussed in detail in Section 2.4.3.

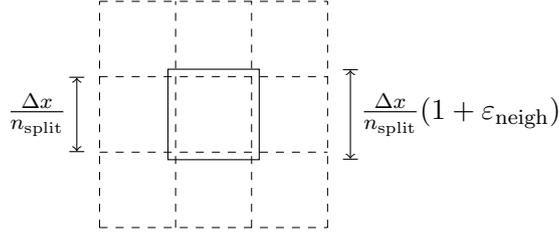


Figure 2.5: Box covering subcell.

Deletion of nodes occurs when the number of nodes in a sub-volume exceeds the limit $n_{\text{max}} \geq 1$. The pairwise closest nodes are replaced by a new node with their mean position and stress until the number of nodes is within the allowed range.

2.4.2 ODE systems

In the forwards-tracking method the ODE systems (2.13) and (2.14) are solved forwards in time for each Lagrangian node, as illustrated in Figure 2.6. The ODE solver algorithm is discussed in Section 2.7. When solving the ODE systems, \mathbf{u} and $\nabla \mathbf{u}$ are interpolated from the Eulerian grid with bilinear or trilinear interpolation, respectively for two and three spatial dimensions, from the Eulerian grid nodes.

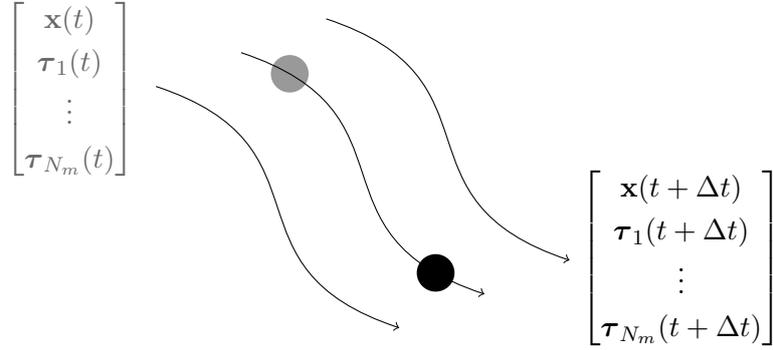


Figure 2.6: Lagrangian node trajectory in flow field.

2.4.3 Radial basis function interpolation

Viscoelastic stresses are stored in the Lagrangian nodes and interpolated to the Eulerian cell centers using radial basis functions. In general, let f be a scalar-valued function which is known at the points $\mathbf{x}_1, \dots, \mathbf{x}_{N_c}$. The approximate value \hat{f} at a point \mathbf{r} is then calculated as (Iske, 2004)

$$\hat{f}(\mathbf{r}) = \sum_{n=1}^{N_c} w_n \psi(\zeta_s |\mathbf{r} - \mathbf{x}_n|) + P(\mathbf{r}), \quad (2.15)$$

where w_1, \dots, w_{N_c} are weights, $\psi : \mathbb{R} \mapsto \mathbb{R}$ a radial basis function and ζ_s a scaling parameter. The term $P(\mathbf{r})$ is the polynomial

$$P(\mathbf{r}) = v_0 + \sum_{s=0}^d v_s r_s, \quad (2.16)$$

where v_0, \dots, v_d are polynomial coefficients and r_0, \dots, r_d the components of \mathbf{r} . The coefficients in (2.15) and (2.16) constitute the solution to the system

$$\begin{bmatrix} \mathbf{M} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix}, \quad (2.17)$$

where

$$M_{ij} = \psi(\zeta_s |\mathbf{x}_i - \mathbf{x}_j|), \quad (2.18)$$

$$\mathbf{B} = \begin{bmatrix} 1 & \dots & 1 \\ \mathbf{x}_1 & \dots & \mathbf{x}_{N_c} \end{bmatrix}^T, \quad (2.19)$$

$$\mathbf{f} = [f(\mathbf{x}_1) \quad \cdots \quad f(\mathbf{x}_{N_c})]^\top, \quad (2.20)$$

$$\mathbf{w} = [w_1 \quad \cdots \quad w_{N_c}]^\top, \quad (2.21)$$

$$\mathbf{v} = [v_0 \quad \cdots \quad v_d]^\top. \quad (2.22)$$

It is noted that the matrices \mathbf{M} and \mathbf{B} depend only on the points $\mathbf{x}_1, \dots, \mathbf{x}_{N_c}$, while the vector \mathbf{f} depends on the function f .

To interpolate the viscoelastic stress to a given point, the Lagrangian nodes close to the point are found. The system (2.17) is then assembled once and solved for each stress component. The close nodes are identified efficiently with a search tree data structure. Depending on the type of implementation, the search tree is either an R-tree structure (Guttman, 1984) or a grid-based structure.

2.5 Backwards-tracking method

The second method developed is denoted the backwards-tracking method. The fundamental strategy of this method is to maintain the structure of the Lagrangian node set by choosing the location of the nodes at the end of each time step a priori. Specifically, the locations are chosen to be at the Eulerian grid nodes. The strategy has been partly inspired the BLP method of Wapperom *et al.* (2000).

Given the structured arrangement of the viscoelastic stress solution, the need for unstructured stress interpolation as well as to maintain the node set is eliminated. As such, the overall robustness is enhanced and the total number of Lagrangian nodes is reduced compared to the forwards-tracking method. Furthermore, the support for refined octree grids is improved.

The viscoelastic stress calculation is carried out as follows. Assuming a known velocity field for $t \in [t_n, t_{n+1}]$ and a stress field for $t = t_n$, the stress at $t = t_{n+1}$ at a given grid node is desired. Hence, the trajectory of the fluid element which resides at the grid node at $t = t_{n+1}$ is needed. Letting $\mathbf{x}(t)$ be the position the fluid element, the trajectory may be expressed as

$$\mathbf{x}(t) = \mathbf{x}(t_{n+1}) - \int_t^{t_{n+1}} \mathbf{u}(t', \mathbf{x}(t')) dt', \quad t \in [t_n, t_{n+1}], \quad (2.23)$$

which is calculated numerically by solving (2.13) backwards in time, starting from $\mathbf{x}(t_{n+1})$. The viscoelastic stress is interpolated from the nodes at time t_n to $\mathbf{x}(t_n)$. Finally, (2.14) is solved forwards in time along the trajectory, such that the viscoelastic stress $\boldsymbol{\tau}(t_{n+1}, \mathbf{x}(t_{n+1}))$, i.e. at the grid node, is obtained.

An illustration of the backwards-tracking procedure is given in Figure 2.7. The steps can be summarized as

1. Calculate the Lagrangian node trajectory by solving (2.13) backwards in time, starting at the Eulerian grid node at $\mathbf{x}(t_{n+1})$.
2. Interpolate the stress $\boldsymbol{\tau}(t_n, \mathbf{x}(t_n))$ to the Lagrangian node from the known stress field at time t_n .
3. Solve (2.14) forwards in time along the trajectory $\mathbf{x}(t)$, $t \in [t_n, t_{n+1}]$.

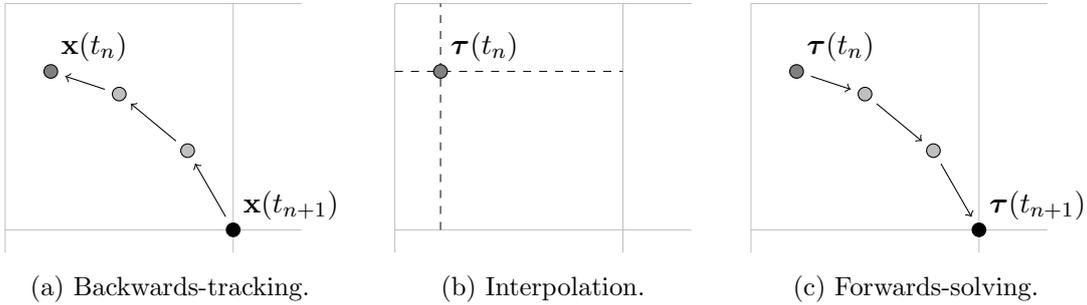


Figure 2.7: Backwards-tracking procedure.

When (2.13) and (2.14) are solved, the velocity along the trajectory is calculated with bilinear or trilinear interpolation, respectively for two and three dimensions, from the cell centers. Similarly, the velocity gradient is calculated as the gradient of the bilinear or trilinear interpolation expressions. For interpolation to a point near grid refinements, the smallest local cell size is used to form the interpolation box. For corners of the interpolation box which do not coincide with a cell center, the velocities are calculated with a first order polynomial approximation from the cells intersecting the box. The interpolation box is visualized in Figure 2.8.

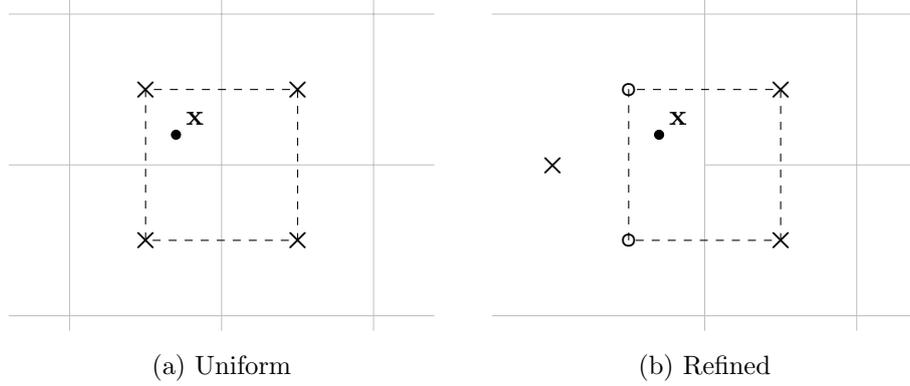


Figure 2.8: Basis for interpolating properties stored at the Eulerian cell centers to Lagrangian nodes in (a) areas with uniform grid spacing and (b) near refinements.

2.6 Viscoelastic contribution to momentum equation

In the momentum equation (1.17) the viscoelastic stress affects the flow through the term $\nabla \cdot \boldsymbol{\tau}$. As discussed in Section 2.2.2, the volume fraction averaging (2.5) is applied to the viscoelastic stress. Thus, the viscoelastic stress contribution is calculated as

$$\nabla \cdot (\alpha \boldsymbol{\tau}) = \alpha \nabla \cdot \boldsymbol{\tau} + \boldsymbol{\tau} \cdot \nabla \alpha, \quad (2.24)$$

in which the product rule has been applied. This is interpreted as a separation of a pure interfacial contribution of the stress divergence term and the remainder part (Niethammer *et al.*, 2019). The first term of (2.24) is integrated with Gauss' divergence theorem as

$$\int_{\text{c.v.}} \nabla \cdot \boldsymbol{\tau} dV = \int_{\text{c.s.}} \hat{\mathbf{n}} \cdot \boldsymbol{\tau} dS = \sum_f \int_{\text{f.s.}} \hat{\mathbf{n}}_f \cdot \boldsymbol{\tau} dA, \quad (2.25)$$

where c.v. denotes the cell volume, c.s. the cell boundary surface and $\hat{\mathbf{n}}$ the outwards surface normal. In the second step of (2.25) the surface integral is expressed as the sum of the cell face integrals, for which the normal vectors $\hat{\mathbf{n}}_f$ are constant. The volume integral of the second term in (2.24) is approximated as

$$\int_{\text{c.v.}} (\boldsymbol{\tau} \cdot \nabla \alpha) dV \approx (\overline{\boldsymbol{\tau}} \cdot \overline{\nabla \alpha}) \Delta V, \quad (2.26)$$

where ΔV is the cell volume and $\overline{(\bullet)}$ denotes the volume average. The quantity $\overline{\nabla \alpha}$ is calculated using central differences.

In the forwards-tracking method, the cell face integrals are approximated through linear interpolating of the viscoelastic stress to the face centers. For faces at the domain boundary or inside immersed boundaries the stress is linearly extrapolated from using the neighboring cell at the opposite face.

In the backwards-tracking method, the face integrals are approximated with the trapezoidal rule, directly using the stresses stored at the grid nodes. For cell faces adjacent to smaller cells, e.g. as shown in Figure 2.1, the sub-faces are integrated separately such that the contribution from each grid node is included.

For two-fluid flow, the threshold volume fraction threshold $\alpha_{\text{lim},2}$ is used, see Section 2.2.2. Cells in which $\alpha < \alpha_{\text{lim},2}$ are assumed to lie outside the viscoelastic phase and, hence, the viscoelastic contribution is assumed to vanish.

2.7 ODE solver

Let \mathbf{y} be the solution vector to the ODE system

$$\begin{cases} \dot{\mathbf{y}}(t) = \mathbf{g}(t, \mathbf{y}) \\ \mathbf{y}(t_0) = \mathbf{y}_0 \end{cases}, \quad (2.27)$$

The system (2.27) is solved for a global time step of length Δt with $N_{\text{loc}} > 0$ local steps of size $\Delta t_1, \dots, \Delta t_{N_{\text{loc}}}$, where $\Delta t_1 + \dots + \Delta t_{N_{\text{loc}}} = \Delta t$.

2.7.1 Forwards-tracking method

Two ODE solution methods are employed for the forwards-tracking method, depending on the implementation. The first method is a second order backwards-differentiation formula (BDF). An approximate solution \mathbf{y}_n at time t_n is calculated from

$$b_n \Delta t_n \dot{\mathbf{y}}_n - \mathbf{y}_n + a_{n,1} \mathbf{y}_{n-1} - a_{n,2} \mathbf{y}_{n-2} = \mathbf{0}, \quad (2.28)$$

where $\Delta t_n = t_n - t_{n-1}$ is of variable length and the coefficients b_n , $a_{n,1}$ and $a_{n,2}$ are uniquely determined given the recent step size history (Hindmarsh *et al.*, 2018).

The second method is the more simple implicit Euler method, for which an approximate solution at time t_n is calculated from

$$\mathbf{y}_n = \mathbf{y}_{n-1} + \Delta t_n \dot{\mathbf{y}}_n, \quad (2.29)$$

where Δt_n are constant for $n = 1, \dots, k$.

2.7.2 Backwards-tracking method

For the backwards-tracking method, the fourth order Runge-Kutta RK4 method (Tahir-Kheli, 2018) is used to solve the ODE systems. An approximate solution at $t = t_n$ is then calculated explicitly as

$$\mathbf{y}(t_n) = \mathbf{y}(t_{n-1}) + \frac{1}{6} (\mathbf{K}_1 + 2\mathbf{K}_2 + 2\mathbf{K}_3 + \mathbf{K}_4), \quad (2.30)$$

with the coefficient vectors $\mathbf{K}_1, \dots, \mathbf{K}_4$

$$\mathbf{K}_1 = \mathbf{g}(t_{n-1}, \mathbf{y}_{n-1}) \Delta t_n, \quad (2.31)$$

$$\mathbf{K}_2 = \mathbf{g}\left(t_{n-1} + \frac{\Delta t_n}{2}, \mathbf{y}_{n-1} + \frac{\mathbf{K}_1}{2}\right) \Delta t_n, \quad (2.32)$$

$$\mathbf{K}_3 = \mathbf{g}\left(t_{n-1} + \frac{\Delta t_n}{2}, \mathbf{y}_{n-1} + \frac{\mathbf{K}_2}{2}\right) \Delta t_n, \quad (2.33)$$

$$\mathbf{K}_4 = \mathbf{g}(t_n, \mathbf{y}_{n-1} + \mathbf{K}_3) \Delta t_n. \quad (2.34)$$

2.8 Implementation

An important aspect of the research presented in this thesis is the software implementation of the proposed methodology. The algorithms presented are mainly implemented in C++ code and are executed in parallel on the CPU. The backwards-tracking method is implemented entirely in CPU code, while for the forwards-tracking method a GPU-accelerated version has also been developed. In this version, the ODE solver and the RBF interpolation are parallelized on the GPU. The algorithms are implemented in CUDA C++ code and utilize the Thrust library (Bell & Hoberock, 2011) for the parallelization. A summary of the available implementations of the ODE solver algorithms and the RBF interpolation is given in Table 2.1.

The forwards-tracking method is implemented in three versions with the intent to enable performance comparison between the CPU-based and GPU-accelerated versions. The implemented versions, including the backwards-tracking method, are summarized in Table 2.2. CPU-BDF constitutes the original implementation of the forwards-tracking method, while CPU-Euler is mainly used to enable fair comparison between

Routine	CPU-impl.	GPU-impl.
BDF ODE solver	✓	
Implicit Euler ODE solver	✓	✓
Runge-Kutta RK4 ODE solver	✓	
RBF interpolation	✓	✓

Table 2.1: Summary of CPU and GPU implementations of ODE solvers and interpolation routines.

the CPU-based and the GPU-based algorithms, with respect to computational performance.

Implementation	ODE solver	RBF interpolation
CPU-BDF	BDF, CPU	CPU
CPU-Euler	Implicit Euler, CPU	CPU
GPU	Implicit Euler, GPU	GPU
Backwards-tracking	RK4, CPU	-

Table 2.2: Combinations of ODE solver and interpolation routines implemented for the Lagrangian–Eulerian methods.

2.8.1 ODE solver

The BDF formula is implemented for execution on the CPU with the available solvers in the Sundials CVode library (Hindmarsh *et al.*, 2018; Sundials, 2020), while the implicit Euler and Runge-Kutta RK4 methods, respectively, are implemented in our in-house code. The implicit Runge-Kutta RK4 method is implemented for execution on the CPU, while the implicit Euler method is implemented for execution either on the CPU or the GPU.

2.8.2 Radial basis function interpolation

The RBF interpolation procedure consists of two main steps:

1. Find all Lagrangian nodes close to the point.

2. Solve the system (2.17) and calculate the interpolated stress components.

The first step is performed differently on the CPU and GPU. In the CPU implementation the Lagrangian node positions are stored in an R-tree data structure, enabling efficient identification of the nodes within a given distance to a point. The R-Tree structure used is implemented in the Boost C++ libraries (Dawes & Abraham, 2020). In the GPU implementation, the Lagrangian nodes are instead stored in a lattice grid structure. When interpolating the viscoelastic stress to a given cell center, all nodes residing in the cell are simply included.

While the R-tree implementation arguably is a more general, the lattice grid structure used for the GPU-implementation is more efficient for the GPU-architecture. This is the main argument for using different search-strategies for the different implementations in the current work.

2.8.3 Numerical stability

Both sides diffusion is implemented in the numerical framework, but is only used for the study of steady flows due to its inherent disadvantages for transient flows, discussed in Section 1.7.

The Lagrangian-Eulerian framework is compatible with reformulation techniques, e.g. the log-conformation representation. LCR is implemented but is not used for the any of the numerical studies presented in this thesis. The intent is instead to evaluate the numerical method without advanced stabilization techniques unless necessary.

Chapter 3

Results and discussion

In this chapter simulation results obtained with the proposed Lagrangian–Eulerian methods are summarized and discussed. Since the focus of the thesis is to present new simulation methodology for viscoelastic flow, the results are focused on validation of the method in different aspects.

Firstly, results for viscoelastic single-phase flows are presented for the proposed methods. A discussion around the performance enhancements obtained for GPU calculations is included in this section. This section summarizes Paper I–II and partly Paper III. Secondly, multiphase flows are discussed, summarizing Paper III–IV. Finally, simulated adhesive joining applications are demonstrated, summarizing the results from Paper V.

For the results presented in this section, the solvent viscosity and polymeric viscosity are quantified by the total viscosity η_t and the viscosity ratio

$$\beta = \frac{\mu}{\mu + \eta}, \quad (3.1)$$

such that $\mu = \beta\eta_t$ and $\eta = (1 - \beta)\eta_t$.

3.1 Single-phase flow

A first validation study of the forwards-tracking method is conducted for a planar Poiseuille flow and for a confined cylinder flow. For the Poiseuille flow, analytic solutions are available. For the confined cylinder flow, the computed results are compared to numerical and experimental data from the literature.

3. RESULTS AND DISCUSSION

The Poiseuille flow, shown in Figure 3.1, is subject to a constant pressure drop Δp and periodic boundary conditions in the streamwise direction. Thus, effectively an infinitely long channel is modeled. The viscoelastic fluid is described with the UCM and Oldroyd-B models, with the constitutive equation

$$\lambda \overset{\nabla}{\boldsymbol{\tau}} + \boldsymbol{\tau} = 2\eta \mathbf{S}, \quad (3.2)$$

with the viscosity ratio $\beta = 0$ for the UCM model and $\beta \in (0, 1)$ for the Oldroyd-B model.

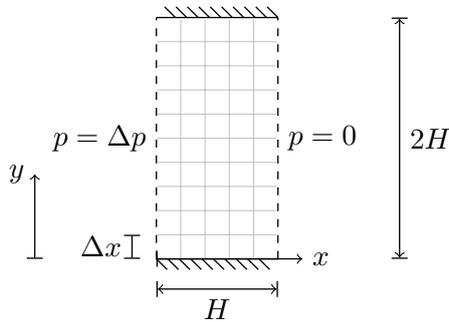


Figure 3.1: Planar Poiseuille flow.

All simulations are transient and initialized with vanishing velocity and stress fields. The accuracy is estimated with the relative errors at steady flow conditions with respect to the analytic solution. The error for a flow quantity ϕ is calculated as

$$E_\phi = \frac{\|\phi - \phi_a\|_{L_2}}{\|\phi_a\|_{L_2}}, \quad (3.3)$$

where $\|\bullet\|_{L_2}$ denotes the L_2 -norm over the fluid cells and ϕ_a the analytic solution. The Weissenberg number is defined as $Wi = \lambda U/H$, where U is the mean steady flow velocity. In Figure 3.2 the errors calculated for the velocity and the viscoelastic normal stress for the UCM model are shown for varying uniform grid spacing Δx and Weissenberg number. The errors converge to zero with second order accuracy for the considered Wi range, which agrees with the order of accuracy of the momentum and continuity discretization for the case.

The simulated transient startup flow compared to the corresponding analytic solution for Oldroyd-B model with varying β and Wi is shown in Figure 3.3. Here, the

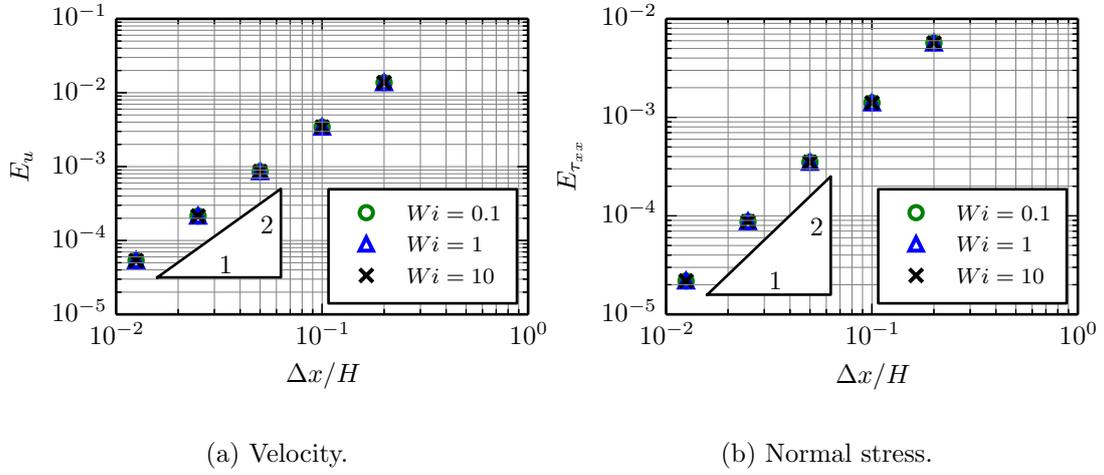


Figure 3.2: Relative errors with respect to analytic solution in planar Poiseuille flow for (a) velocity and (b) normal stress.

streamwise velocity at the center of the channel is shown. Both β and Wi strongly influence the transient flow dynamics. This is captured in the numerical solution, which practically overlap the analytic curves.

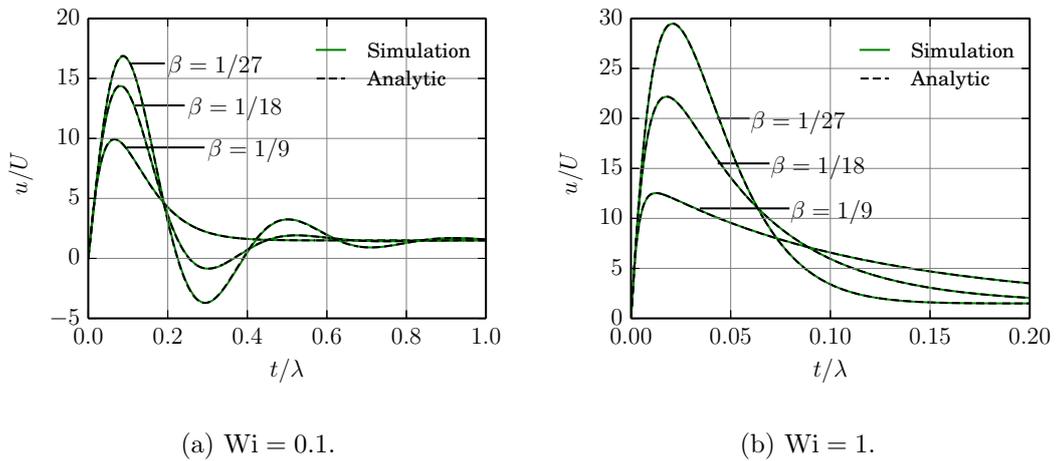


Figure 3.3: Transient streamwise velocity at channel centerline for startup of planar Poiseuille flow.

3. RESULTS AND DISCUSSION

The confined cylinder geometry is shown in Figure 3.4. The flow involves both shear and extension characteristics and is a common benchmark problem for viscoelastic flow simulation, see for example Alves *et al.* (2001); Oliveira *et al.* (1998). In this work, the intent is to validate the proposed numerical method for flows involving realistic viscoelastic fluids as well as complex geometry. As such, the use of the Lagrangian–Eulerian method with immersed boundary conditions is assessed.

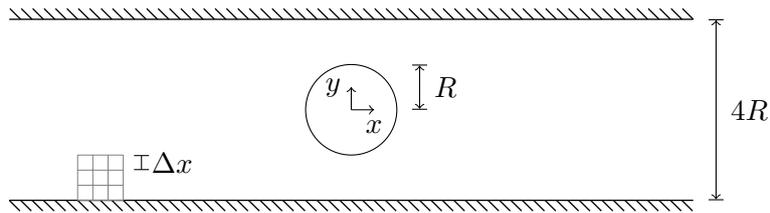


Figure 3.4: Symmetrically confined cylinder flow.

The viscoelastic fluid is modeled by a four-mode linear-form PTT model with the constitutive equation

$$\lambda_k \nabla \cdot \boldsymbol{\tau}_k + \left(1 + \frac{\varepsilon_k \eta_k}{\lambda_k} \text{Tr}(\boldsymbol{\tau}_k) \right) \boldsymbol{\tau}_k = 2\eta_k \mathbf{S}, \quad k = 1, \dots, 4, \quad (3.4)$$

for which the parameters are chosen to match those reported by Baaijens *et al.* (1995) for a viscoelastic polyisobutylene solution, see Table 3.1. In their work, experimental measurements as well as FEM simulation of the flow were reported.

η_k [Pa · s]	0.443	0.440	0.0929	0.00170
λ_k [s]	0.00430	0.0370	0.203	3.00
ε_k	0.39	0.39	0.39	0.39

Table 3.1: Parameters for the SPTT model used in the confined cylinder flow.

A Deborah number is defined as $\text{De} = \bar{\lambda}U/R$ and a Reynolds number as $\text{Re} = \rho RU/\eta_0$, where η_0 is the total viscosity $\eta_0 = \eta_1 + \dots + \eta_4$ and $\bar{\lambda}$ the mean relaxation time $\bar{\lambda} = (\eta_1\lambda_1 + \dots + \eta_4\lambda_4)/\eta_0$

The computed results are compared to the data from Baaijens *et al.* (1995) at steady flow conditions for $\text{De} = 0.25, 0.93, 2.32$, corresponding to $\text{Re} = 0.019, 0.069, 0.174$. The velocity u and first normal stress difference $N_1 = \tau_{xx} - \tau_{yy}$ are compared across the

channel at different locations in Figure 3.5 and along the centerline in Figure 3.6. The velocity is normalized by the mean velocity U and the stress by $\tau_0 = 3\eta_0 U/R$.

Overall, the results agree well with the FEM simulations from Baaijens *et al.* (1995). A few small discrepancies are observed, which could be attributed to the different numerical methods and uncertainties in the data. No grid dependency assessment was reported for the FEM simulations and the raw data from Baaijens *et al.* (1995) have not been available for the comparison. However, it is noted that the discrepancies between the numerical results are small compared to the differences between the numerical and experimental data.

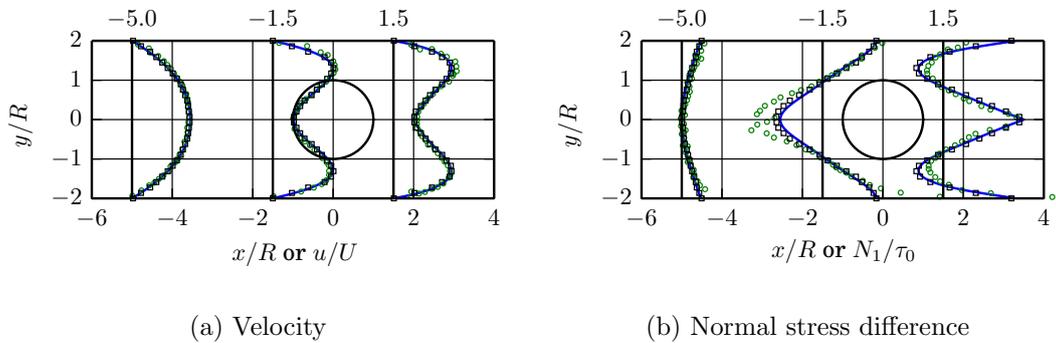


Figure 3.5: Computed quantities across the channel for $De = 2.32$ (—), compared to FEM-simulations (\square) and experiments (\circ) from Baaijens *et al.* (1995).

Combined, the simulations of the Poiseuille and confined cylinder flows show the capability of the proposed numerical method to simulate different viscoelastic flows. This includes realistic viscoelastic fluids and complex geometry described with immersed boundary conditions. Similar studies have been conducted to validate the different implementations of the forwards-tracking method, discussed in Section 2.8, as well as the backwards-tracking method. Although the complete study is not repeated for each method and implementation, a certain overlap is asserted, summarized in Table 3.2.

3.2 Computational performance

An example of relative contributions to the simulation time from the ODE solver, the RBF interpolation and the node redistribution are shown in Figure 3.7. The times

3. RESULTS AND DISCUSSION

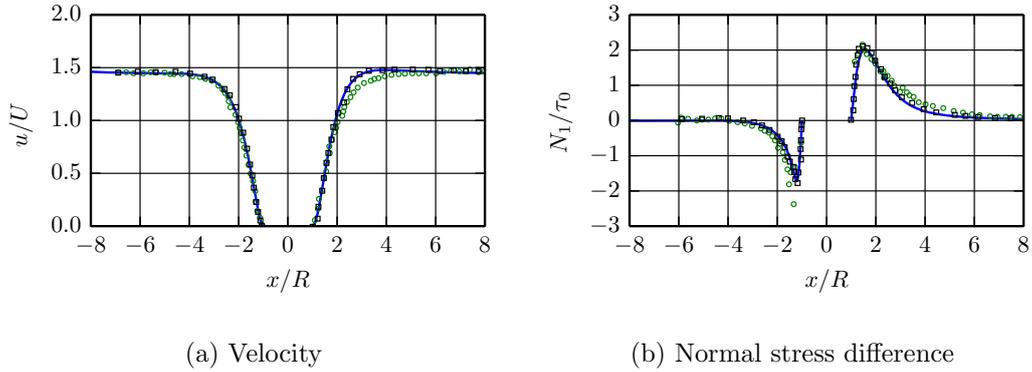


Figure 3.6: Computed quantities along the confined cylinder channel centerline $y = 0$ for $De = 2.32$ (—), compared to FEM-simulations (\square) and experiments (\circ) from Baaijens *et al.* (1995).

	Fwd (CPU-BDF)	Fwd (CPU-Euler)	Fwd (GPU)	Bwd
Poiseuille (steady)	✓	✓	✓	✓
Poiseuille (transient)	✓			✓
Confined cylinder	✓	✓	✓	

Table 3.2: Overview of single-phase validation cases simulated with the versions of the Lagrangian–Eulerian method presented in Chapter 2.

have been measured for 100 time steps of the confined cylinder flow, simulated with four processor cores with the CPU-BDF implementation, discussed in Section 2.8. From Figure 3.7, it is evident that substantial performance gains may be obtained by reducing the computational cost of these operations.

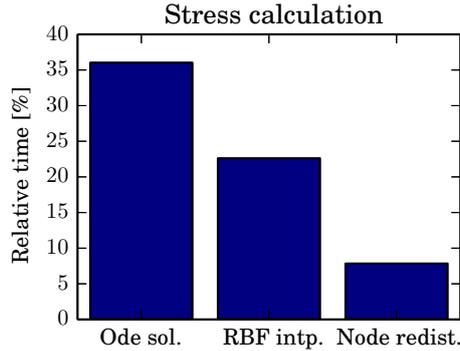


Figure 3.7: Relative computational times in CPU-BDF implementation..

As previously stated, the Lagrangian form of the viscoelastic constitutive equation enables straightforward parallelization of the calculations. Utilizing this property, the GPU-accelerated version of the forwards-tracking method has been implemented. A performance comparison of the implementations of the forwards-tracking method discussed in Section 2.8 is presented in this section.

The confined cylinder flow described in Section 3.1 is employed as the benchmark case for computational performance. Simulations with 100 time steps are performed with varying grid resolution as well as number of CPU cores. At the highest resolution, 512000 Eulerian cells and approximately 2 million Lagrangian nodes are used. All simulations are performed with an Intel(R) Xeon(R) Gold 6134 CPU with 8 3.20 GHz cores and with a Tesla V100 GPU with 32 Gb memory.

The operations which differ between the implementations are the ODE solver and the RBF interpolation. In Figure 3.8 the average time per simulation step for these operations are compared for the different implementations. The GPU implementation is associated with smallest times for both operations. The CPU-based variants share the implementation of the RBF interpolation. Hence, the interpolation times are identical for the two. The ODE solution executes faster for the CPU-Euler implementation than the CPU-BDF version. It is remarked that the implicit Euler method is performed

3. RESULTS AND DISCUSSION

with a single substep in both the CPU and the GPU implementation. The computational cost of the CPU-Euler implementation increases rapidly with the number of substeps, while the GPU implementation is much less sensitive to this parameter. This is discussed in greater detail in Paper II.

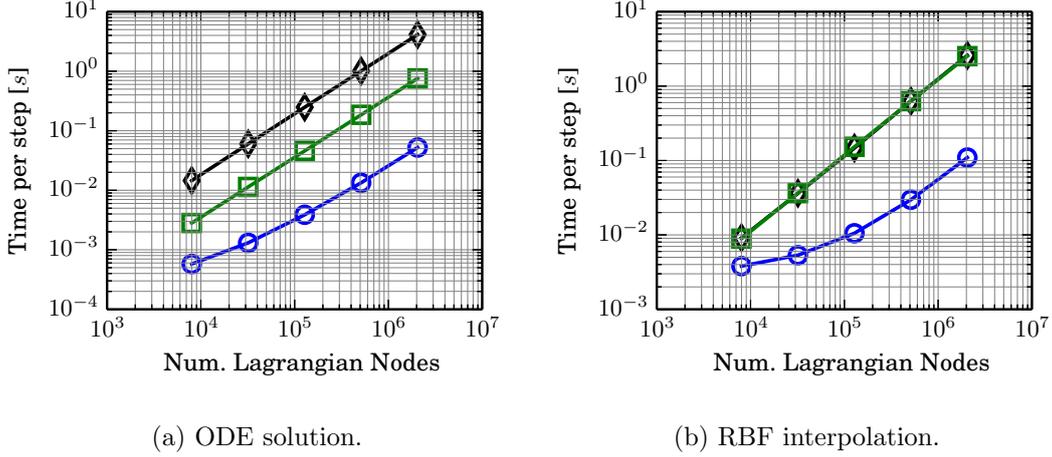


Figure 3.8: Average time for solving (a) ODE systems and (b) RBF interpolation for the four-mode PTT fluid in the confined cylinder channel for GPU (○), CPU-BDF (◇) and CPU-Euler (□) using 4 CPU threads.

In Figure 3.9 the corresponding comparison for varying number of CPU cores is shown for the highest grid resolution. Similar relationships between the simulation times are found. The parallelization on the GPU is not affected by the number of processor cores. Hence, the measured times for the GPU implementation are essentially constant with respect to the number of CPU cores, while remaining substantially smaller than the CPU counterparts.

Ultimately, the performance of the full algorithm is the main property of interest. In Figure 3.10 the measured computation times for the full viscoelastic stress calculation and for a full simulation step are compared. As expected, the GPU-implementation results in a major reduction in computation time, compared to the CPU implementations.

To summarize, the GPU-implementation reduces the computation time for the ODE solver by 98-99.6% and the RBF interpolation by 93-98%, depending on the number of CPU cores used. The corresponding decrease of the total simulation time

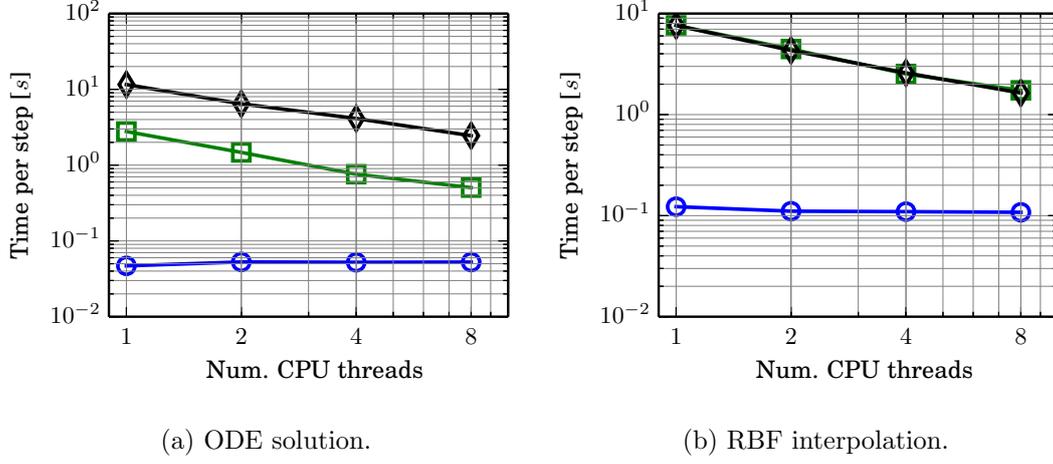


Figure 3.9: Average time for (a) solving ODE systems and (b) RBF interpolation for the four-mode PTT fluid in the confined cylinder channel for GPU (\circ), CPU-BDF (\diamond) and CPU-Euler (\square).

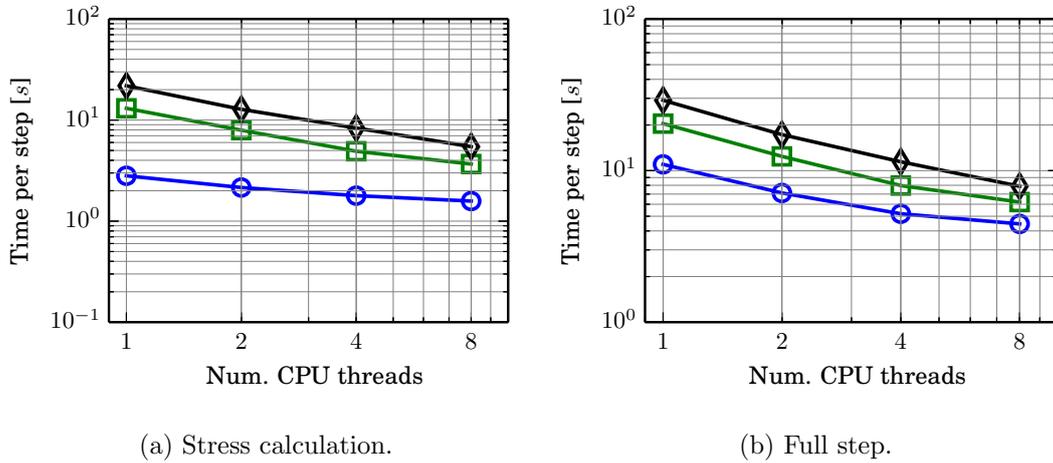


Figure 3.10: Average time for (a) viscoelastic stress calculation and (b) full time step for the four-mode PTT fluid in the confined cylinder channel for GPU (\circ), CPU-BDF (\diamond) and CPU-Euler (\square).

is 44-62%. Consequently, it can be concluded the Lagrangian–Eulerian algorithm is indeed well-suited for GPU-acceleration. Moreover, the reduction of total time for the CPU-Euler implementation is 21-30%, demonstrating the influence of the choice of ODE solver algorithm.

Although the performance has been assessed in detail for the forwards-tracking method specifically, the observed results have important implications for the backwards-tracking method as well. The backwards-tracking method requires a smaller number of Lagrangian nodes. Furthermore, the RBF-interpolation as well as the node redistribution are eliminated. Thus, the backwards-tracking has a lower computational cost than the forwards-tracking method by construction.

3.3 Free-surface flow

So far, the simulations presented have considered single-phase viscoelastic flow, whereas many applications involve free surface flow. Adhesive joining applications, being of particular interest in this thesis, constitute such an example. Other examples include e.g. polymer extrusion and additive manufacturing. In this section the focus is aimed on viscoelastic free surface flow simulations. The cases are selected for their relevance as viscoelastic benchmarks as well as for industrial applications. The simulations are performed with the backwards-tracking Lagrangian–Eulerian method.

The die swell effect is important for e.g. polymer processing applications and has been subject to extensive numerical study, see for example Comminal *et al.* (2018); Crochet & Keunings (1982); Habla *et al.* (2011); Oishi *et al.* (2011, 2008); Spanjaards *et al.* (2019); Tomé *et al.* (2002, 2008, 2010, 2012). In the case of adhesive joining applications, viscoelastic swelling may appear e.g. during adhesive extrusion or for squeeze flows in parts assembly. The effect arises for viscoelastic fluids subject to constrained flow in a pipe or channel, which emerge through a nozzle or die. As an illustration, Barnes *et al.* (1989) described the fluid as a bundle of elastic threads. In constrained flow the threads are stretched by the streamwise normal stress. As the fluid emerges the threads relax and decrease in length, causing the width of the extrudate to increase.

A planar die swell flow is simulated, illustrated in Figure 3.11. The channel is initially filled with the viscoelastic fluid, which is a single-mode Oldroyd-B fluid with

$\beta = 1/9$, while the expansion zone to the right is filled with a Newtonian gas. The box, which effectively creates the channel, is imposed as an immersed boundary condition. Fully developed stress and velocity profiles are imposed at the inlet. The transient flow is simulated until the viscoelastic fluid exits through the outlet.

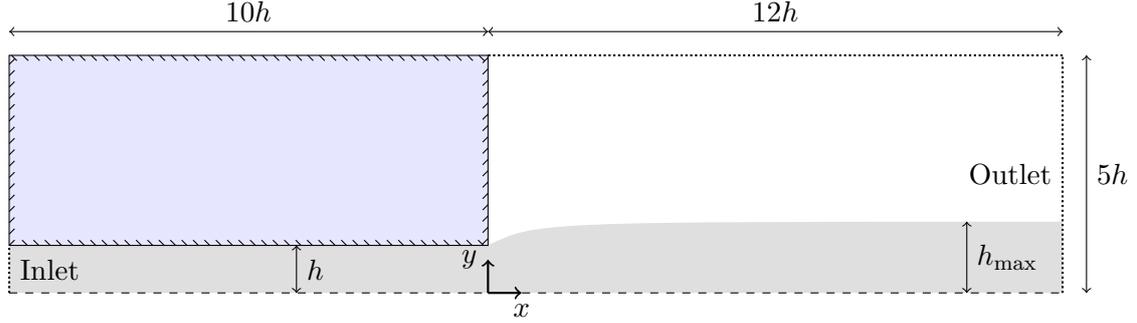


Figure 3.11: Die swell domain.

A Weissenberg number is defined as $Wi = \lambda U/h$ and a Reynolds number as $Re = 2\rho U h/(\mu + \eta)$. The swell ratio is calculated as $S_R = h_{\max}/h$. Furthermore, the so-called recoverable shear S_R is adopted, defined as (Crochet & Keunings, 1982)

$$S_R = \left| \frac{N_1}{2\tau_{xy}} \right|_{y=h} = \left| \frac{\tau_{xx} - \tau_{yy}}{\tau_{xy}} \right|_{y=h} = \frac{3\lambda U}{h} = 3Wi, \quad (3.5)$$

where $N_1 = \tau_{xx} - \tau_{yy}$.

Simulations are performed for $Re = 0.5$ and $S_R = 1, 1.5, 2, 2.5$, varied through λ . In Figure 3.12 and Figure 3.13 the simulations for $S_R = 1$ and $S_R = 2.5$, respectively, are shown at different normalized times $t^* = tU/h$. The viscoelastic fluid is subject to swelling outside the channel, increasing in magnitude with S_R .

To validate the results, the swell ratios are compared to data from the literature, summarized in Table 3.3. The data represents a variety of numerical methods as well as slightly varying setup. For example, the Reynolds are small, but vary slightly between the studies.

The swell ratios obtained from the die swell simulations are compared to the literature data in Figure 3.14. For completeness, the theoretical solution by Tanner (1970, 2005) is included to the comparison. While a certain spread between the data sets

3. RESULTS AND DISCUSSION

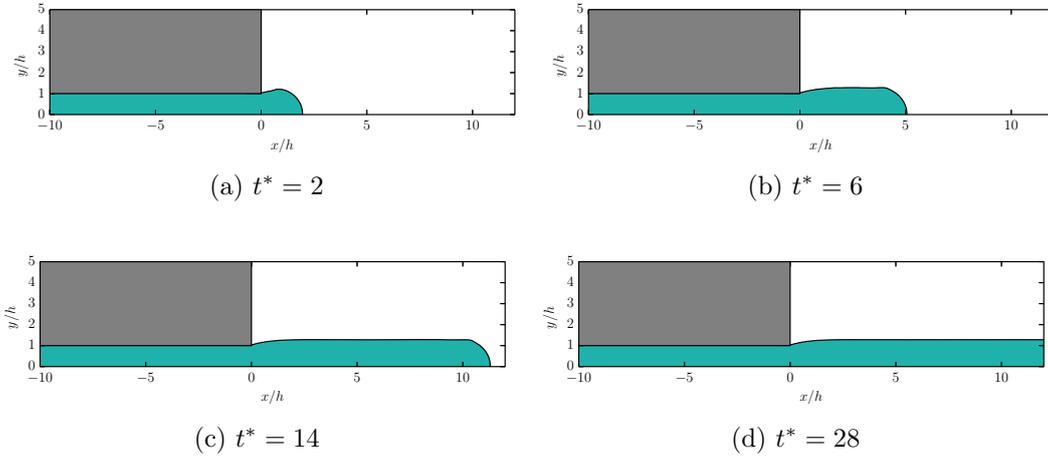


Figure 3.12: Die swell simulation with $S_R = 1$, interface between viscoelastic phase (green) and Newtonian phase (white) visualized for $\alpha = 0.5$.

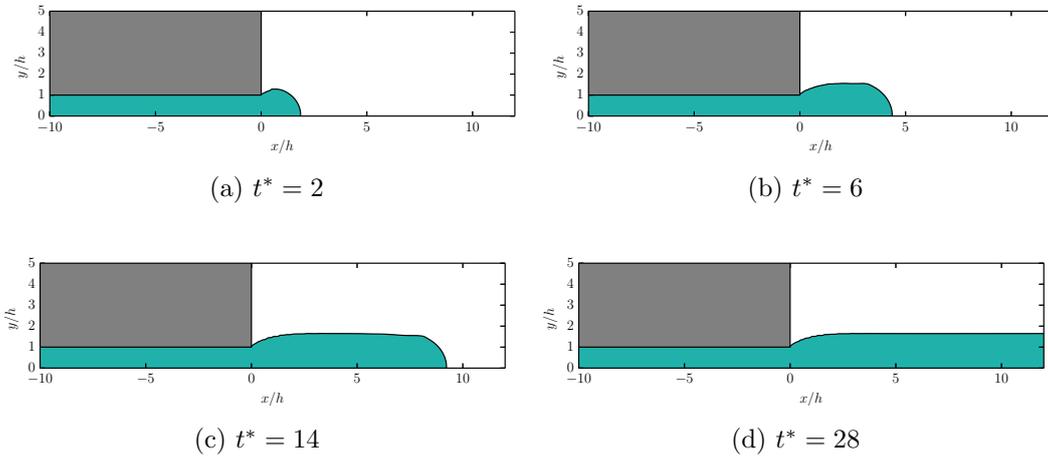


Figure 3.13: Die swell simulation with $S_R = 2.5$, interface between viscoelastic phase (green) and Newtonian phase (white) visualized for $\alpha = 0.5$.

Work	Method	Re
<i>Current work</i>	VOF, Lagrangian-Eulerian	0.5
Crochet & Keunings (1982)	Mixed FEM	0
Tomé <i>et al.</i> (2002)	GENSMAC	0.5
Habla <i>et al.</i> (2011)	pseudo-VOF	0.5
Comminal <i>et al.</i> (2018) (CCU)	VOF, Geometric scheme	0
Comminal <i>et al.</i> (2018) (HRIC)	VOF, Algebraic scheme	0
Comminal <i>et al.</i> (2018) (RheoTool)	VOF, Algebraic scheme (MULES)	0.01

Table 3.3: Summary of numerical data compared for the die swell flow.

exists, the results show similar trends, including those obtained in this work. Hence, the results arguably agree with the literature data, demonstrating the capability to predict the die swell effect correctly.

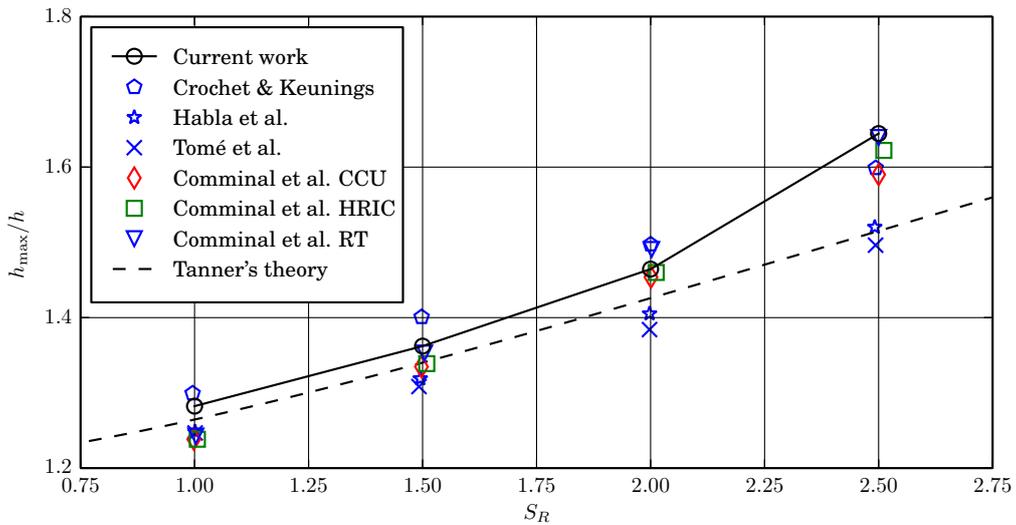


Figure 3.14: Simulated swell ratios compared to data from the literature.

Another well-known phenomenon in viscoelastic free-surface flow is jet buckling. As such, it is a common test case for numerical simulations, see for example Bonito *et al.* (2006); de Paulo *et al.* (2007); Oishi *et al.* (2008); Tomé *et al.* (2002, 2008, 2010, 2012). A more detailed discussion on planar jet buckling is given in Paper III. In

3. RESULTS AND DISCUSSION

Figure 3.15 a three-dimensional jet buckling simulation is shown. The incoming jet has a circular cross section with diameter $D = 5$ mm and inlet velocity $U = 0.5$ m/s. Gravity acts in the negative z -direction. The flow is characterized $Re = \rho DU/\eta_t = 0.25$ and $Wi = \lambda U/D = 10$.

At impact, the jet initially floats and builds upwards. After a certain time, the jet yields and buckles. After the buckling process is initiated, the effect is substantial and continues for the duration of the simulation.

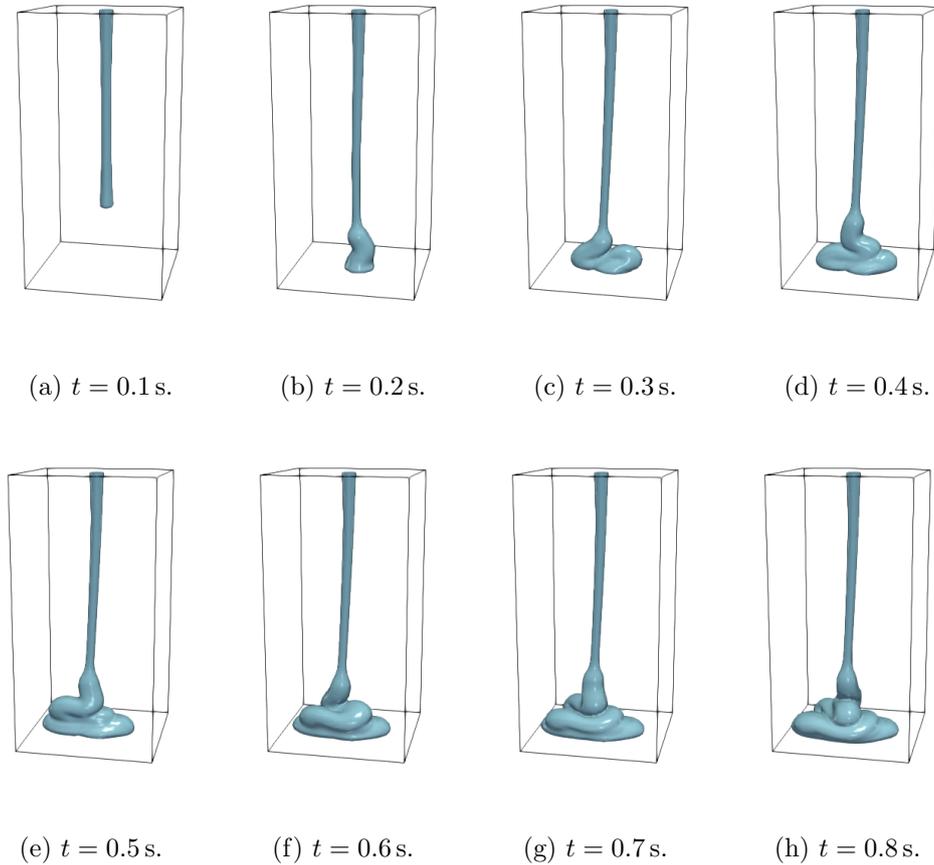


Figure 3.15: Three-dimensional viscoelastic jet buckling simulation, interface between viscoelastic phase visualized for $\alpha = 0.5$.

3.4 Adhesive joining flows

As previously stated, the adhesive joining process is a target application for the developed numerical framework. Typically, this includes a robot-carried adhesive extrusion, in which the adhesive is applied by a nozzle moving along a prescribed path, followed by a joining operation. This section, which concludes the chapter, is focused on such flows.

3.4.1 Adhesive extrusion

Adhesive extrusion is simulated for two cases for which scanned adhesive beads of a structural rubber adhesive are available for comparison. In the first case, as shown in Figure 3.16, adhesive is applied onto a flat plate by a circular nozzle with diameter $d = 2\text{mm}$ moving with constant velocity $u_{\text{app}} = 150\text{mm/s}$ at distance d_{app} to the plate.

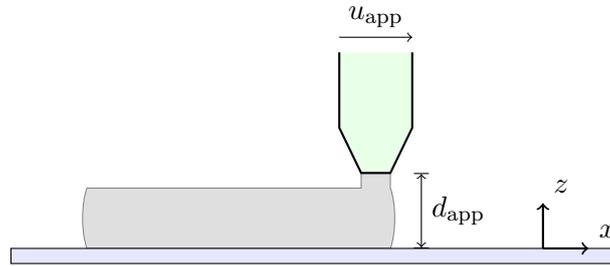


Figure 3.16: Adhesive application on plate .

The extrusion simulation is performed with different magnitudes of the tilt angle θ_t , which refers to a rotation in the plane orthogonal to the nozzle movement direction. The rotation is applied about the target point on the substrate, as shown in Figure 3.17. In this work, the application distance d_{app} refers to the distance between the nozzle center and the target point. Thus, the actual height of the nozzle from the plate, denoted h_{app} , varies as $h_{\text{app}} = d_{\text{app}} \cos \theta_t$.

In this work, the adhesive flow rate is defined through the nominal bead diameter d_{nom} , which constitutes the diameter of a cylindrical bead cross section for a given flow

3. RESULTS AND DISCUSSION

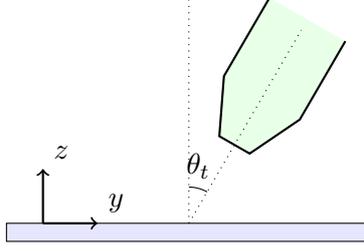


Figure 3.17: Tilt angle in adhesive extrusion, movement direction out-of-plane.

rate and nozzle velocity. The nominal bead cross section area A_{nom} is the volume of adhesive per distance moved by the applicator, i.e. $A_{\text{nom}} = \dot{V}/u_{\text{app}}$. Thus,

$$d_{\text{nom}} = \sqrt{\frac{4A_{\text{nom}}}{\pi}} = \sqrt{\frac{4\dot{V}}{\pi u_{\text{app}}}}. \quad (3.6)$$

The nominal diameter and application distance $d_{\text{nom}} = d_{\text{app}} = 3.5$ mm are used, corresponding to the flow rate $\dot{V} \approx 1.44$ ml/s.

Simulations are carried out with $\theta_t = 0, 10^\circ, 20^\circ, 30^\circ$ for a total bead length of 50 mm. The viscoelastic properties of the adhesive are modeled with a single-mode exponential SPTT model, with $\varepsilon = 0.5$, $\lambda = 0.1$ s, $\eta_t = 10^3$ Pa · s and $\beta = 0.2$.

The inflow of adhesive to the simulation is realized with the injection model, described in Section 2.2.3. In Figure 3.18 the early stages of the simulation with $\theta_t = 0$ are shown, demonstrating the accumulation of adhesive injected to the domain. The full simulation is shown in Figure 3.19.

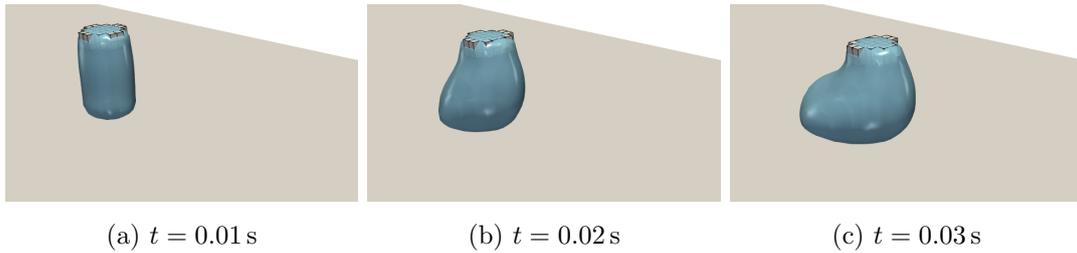


Figure 3.18: Adhesive bead visualized for $\alpha = 0.5$ and injection cells (solid cubes) for extrusion simulation with $d_{\text{nom}} = 3.5$ mm, $u_{\text{app}} = 150$ mm/s and $\theta_t = 0$.

In Figure 3.20 the 3D-scanned beads available for comparison are shown with the simulated beads. Furthermore, the beads are compared in more detail in Figure 3.21.

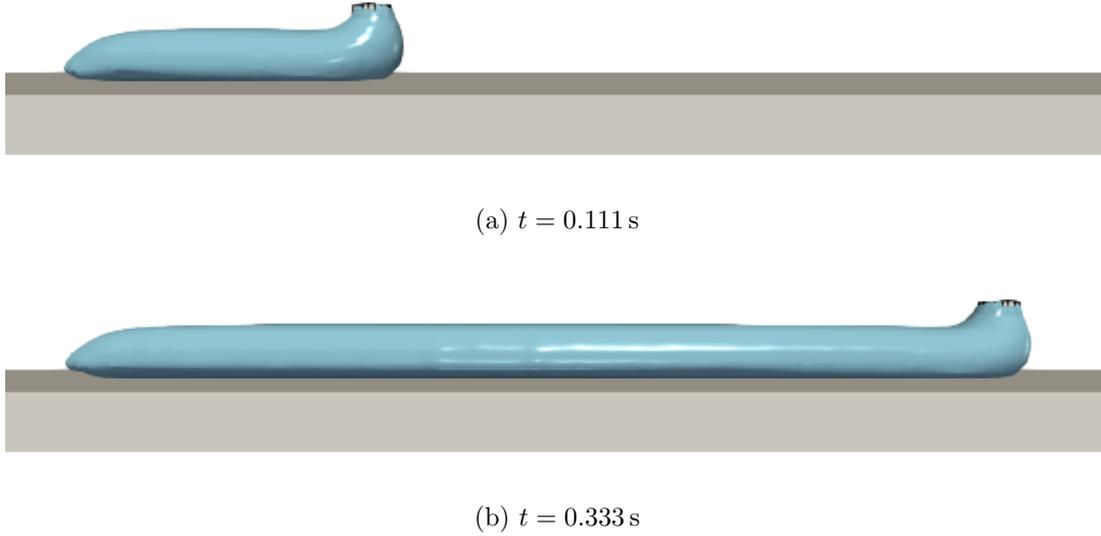


Figure 3.19: Adhesive bead visualized for $\alpha = 0.5$ and injection cells (solid cubes) for extrusion simulation with $d_{\text{nom}} = 3.5$ mm, $\mathbf{u}_{\text{app}} = 150$ mm/s and $\theta_t = 0$.

The comparison includes the three-dimensional bead geometries as well as cross section curves. Overall, a good agreement with the experimental adhesive beads is found for the simulations. For the tilt angles studied, the effect on the bead profile is relatively small. A small asymmetry appears as the tilt angle increases, which is observed both in the simulations and the experimental data.

The next case simulated is the extrusion of adhesive onto a car fender with a robot-carried adhesive nozzle. In contrast to the flat plate case, this application involves a complex product geometry and a nozzle path obtained from a robot program. The motion involves varying application direction, application distance and movement velocity. In Figure 3.22 snapshots from the simulation are shown. A particularly interesting area is shown in detail in Figure 3.23. In this area, the robot decelerates to accurately follow the path. Thus, the local amount of adhesive is large and a minor buckling-like effect occurs.

Also for this case, a 3D-scanned adhesive bead is available for comparison. A comparison between the experimental and simulated beads is given in Figure 3.24.

3. RESULTS AND DISCUSSION

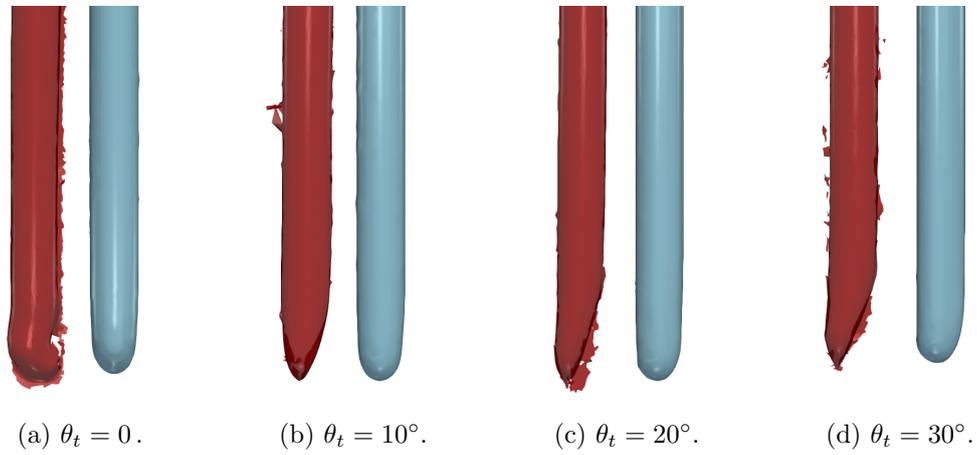


Figure 3.20: 3D-scanned (red) and simulated (blue) adhesive beads on plate for different tilt angles, experiments courtesy of RISE IVF.

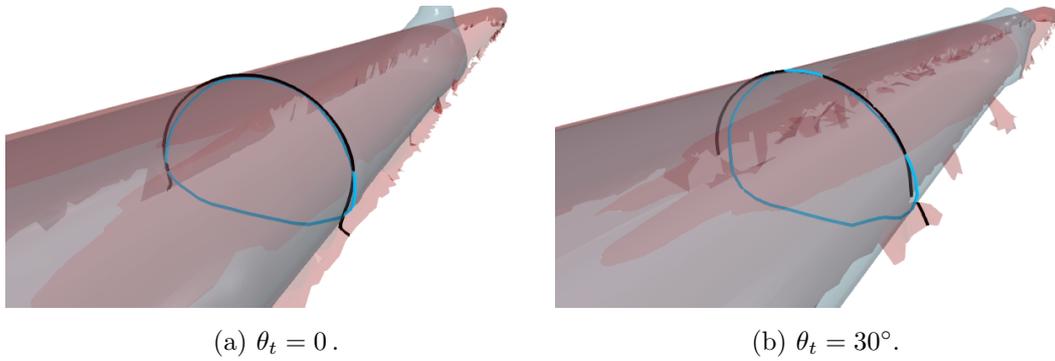


Figure 3.21: Comparison between simulated and scanned adhesive beads, scanned bead (red) with cross section (black), simulated bead (blue) with cross section (blue), experiments courtesy of RISE IVF.

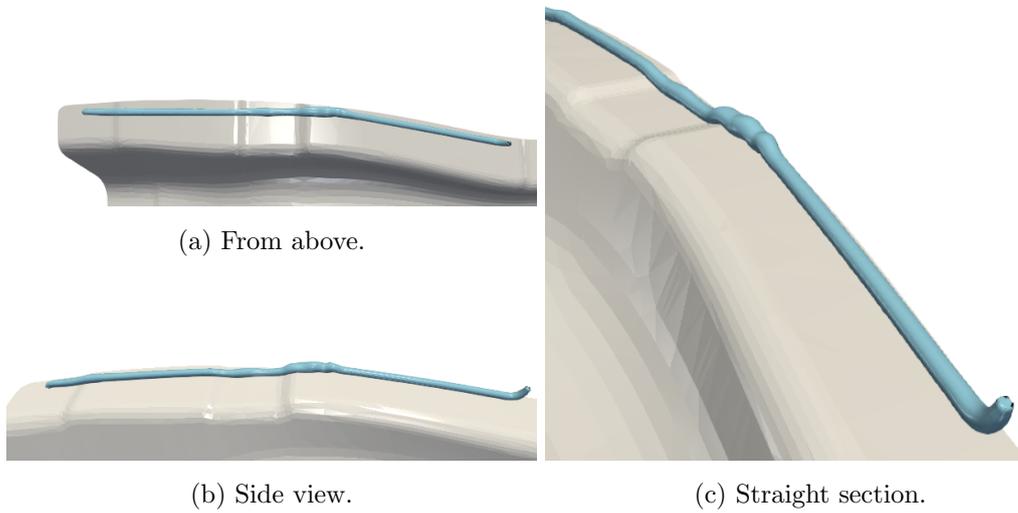


Figure 3.22: Simulated adhesive bead on car fender visualized for $\alpha = 0.5$.

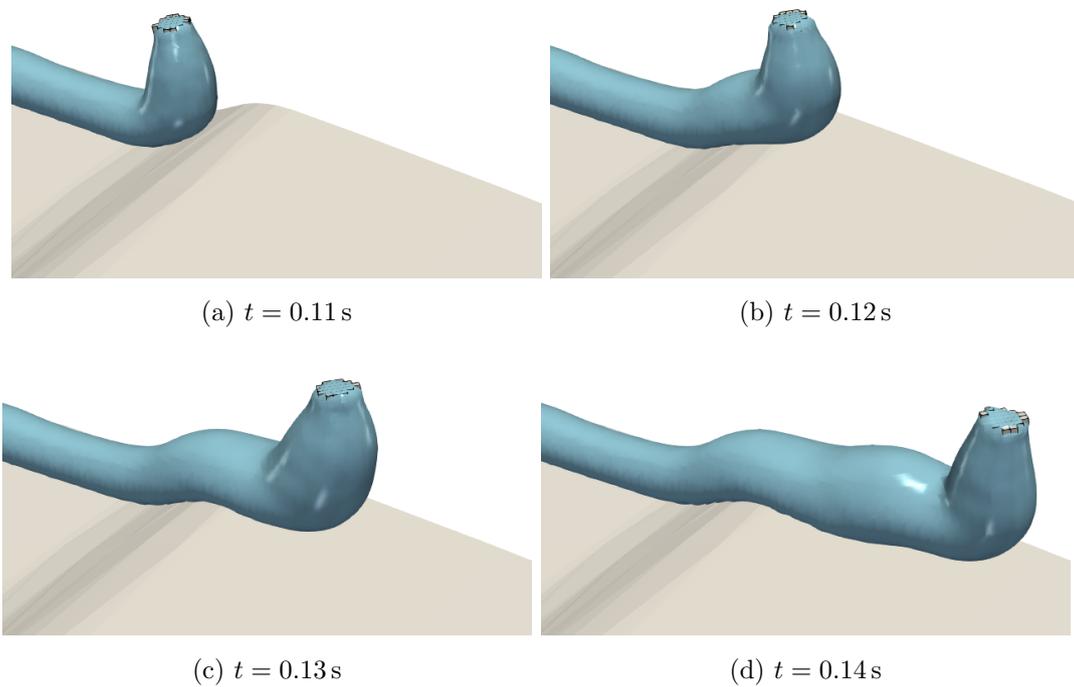


Figure 3.23: Progress of adhesive extrusion on car fender visualized for $\alpha = 0.5$ and injection cells (solid cubes at the top).

3. RESULTS AND DISCUSSION

Due to differences between the nominal fender geometry used for the simulations and the scanned data, the simulated bead has been slightly separated in the figures to enable side-by-side comparison. Good agreement is found between the simulated and experimental adhesive beads. Some volume differences are visible in the first part of the bead. This is likely due to the flow rate in the experiments being higher than the nominal value, possibly attributed to the pre-pressure imposed in the nozzle to initiate flow.

A more detailed comparison is shown in Figure 3.25. In the area where the nozzle decelerates the simulation agrees well with the scanned data. This includes the local increase in adhesive volume as well as the buckling-like effects. In the straight section with near-constant nozzle speed, the agreement is excellent. This is the area where the uncertainties in differences between the nominal conditions and the experiments are typically the smallest.

3.4.2 Joining operations

The adhesive extrusion is followed by a joining operation. This typically involves parts assembly, where the adhesive is squeezed between solid parts, as well as additional steps e.g. screw fastening, riveting or hemming.

In hemming, illustrated in Figure 3.26, two solid parts are first assembled with an adhesive. One part is then folded around the other. At this point spring-back may occur, where the structure recovers a part of the deformation, potentially reducing the join quality.

A conceptual study of the adhesive flow during joining is performed for viscoelastic fluids squeezed between two solids, as shown in Figure 3.27. The upper solid moves with constant velocity V and the lower solid is fixed. The gap between the solids, with initial size h_0 , is initially filled with a viscoelastic fluid and the remainder of the domain with air. A particular focus in the study is the load exerted on the solids, which is of interest in applications, e.g. due to stress limitations in the products or the production equipment. Moreover, accurate load prediction is crucial in FSI applications.

Simulations are performed for the squeeze flow where the upper solid moves downwards, as in Figure 3.27, as well as for the reverse squeeze flow where the solid moves upwards. The flows are characterized by the viscosity ratio $\beta = 1/9$, $\text{Re} = \rho V h_0 / \mu = 0.0009$ and $\text{Wi} \in [0.01, 100]$, varied through λ . In Figure 3.28 the simulation of an

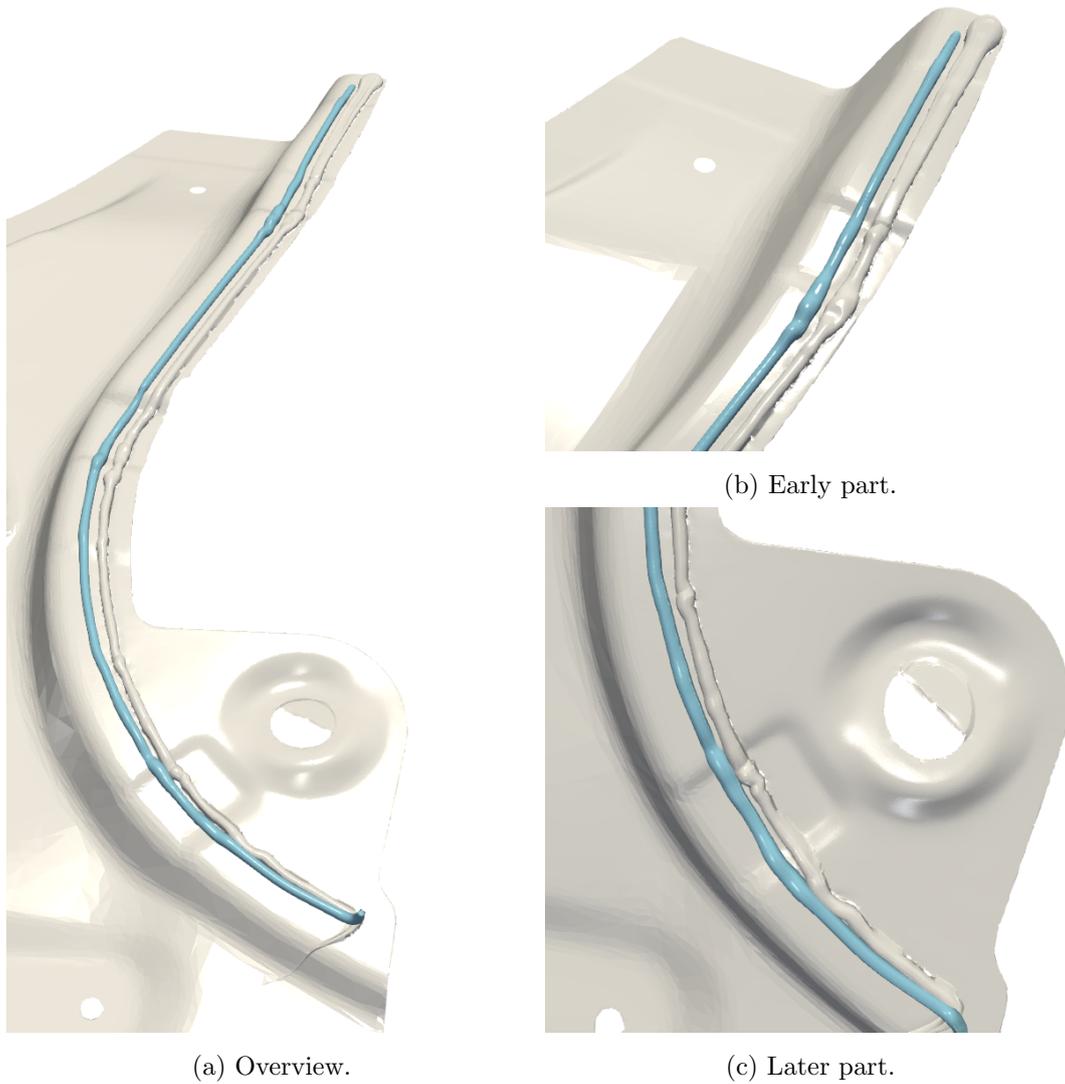
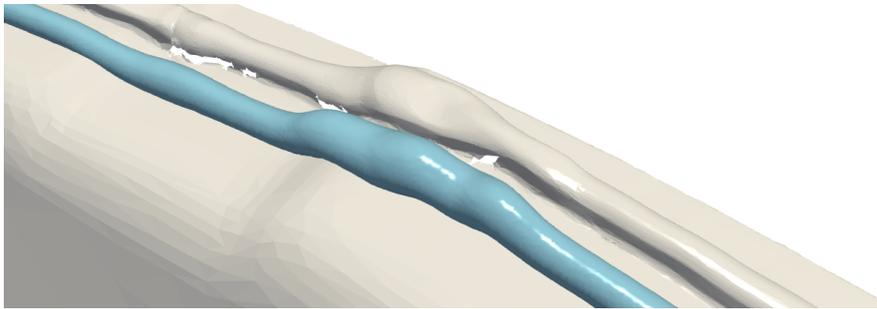
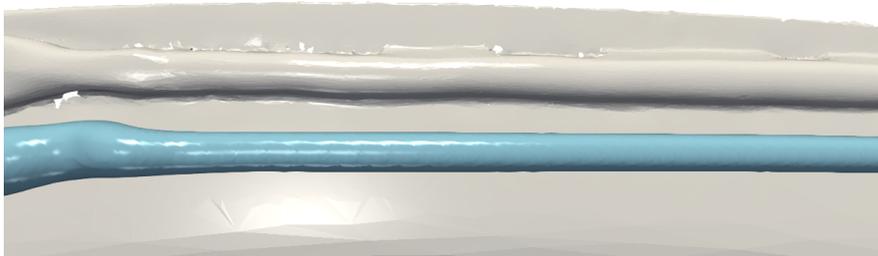


Figure 3.24: Comparison overview between simulated (blue) and scanned (gray) adhesive bead on car fender, experiments courtesy of RISE IVF.

3. RESULTS AND DISCUSSION



(a) Low-velocity section.



(b) Straight section.

Figure 3.25: Comparison overview between simulated (blue) and scanned (gray) adhesive bead on car fender, experiments courtesy of RISE IVF.

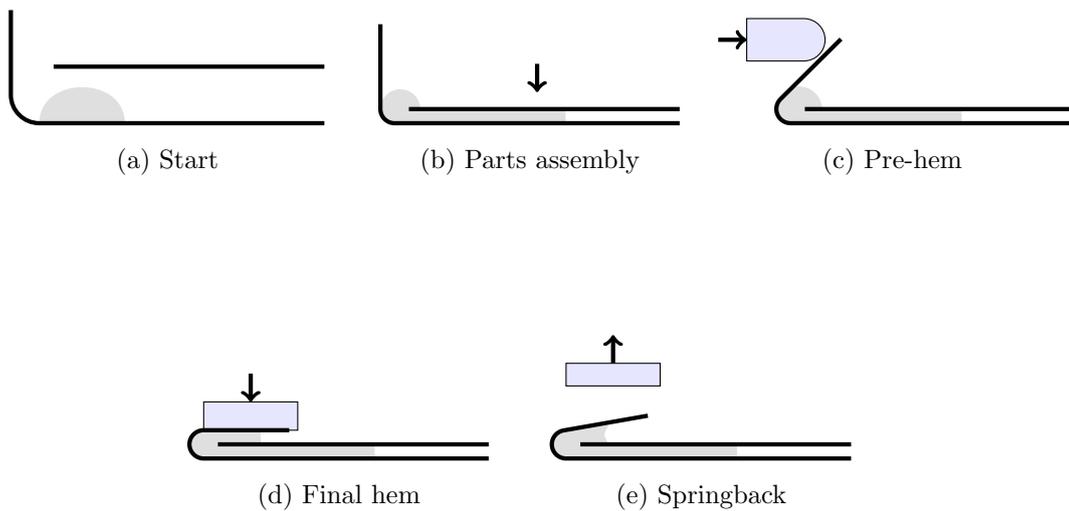


Figure 3.26: Hemming process.

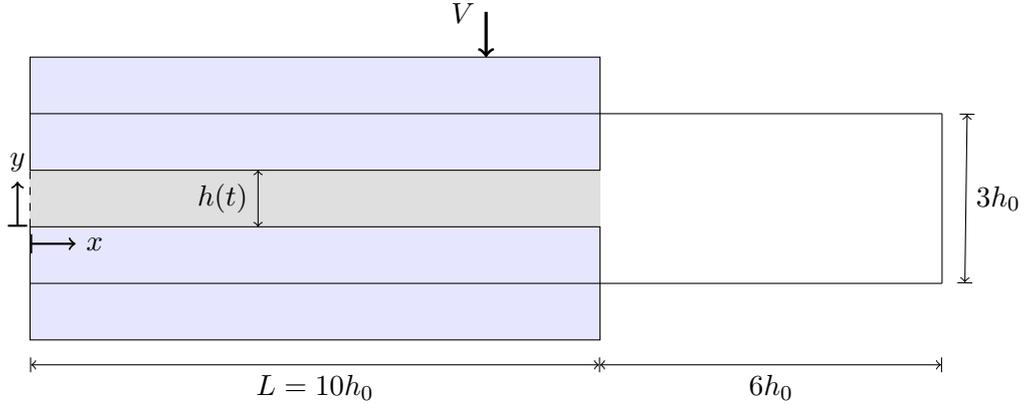


Figure 3.27: Schematic of the squeezing flow between two plates. The gray area between the plates is initially filled with the viscoelastic fluid.

Oldroyd-B fluid with $Wi = 1$ is shown. The dimensionless time t^* is defined as $t^* = h_0/V$. The viscoelastic fluid is squeezed out from the gap and exhibits swelling. This is expected, since the viscoelastic fluid emerges from a constrained flow, similar to the flows discussed in Section 3.3.

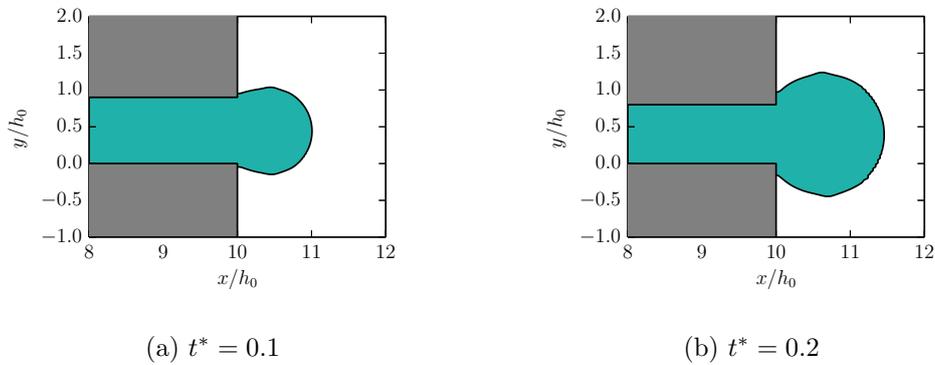


Figure 3.28: Snapshots of squeeze flow simulation with the Oldroyd-B model for $Wi = 1$ with the viscoelastic fluid (green) and the air (white) as visualized for $\alpha = 0.5$.

In Figure 3.29 the corresponding reverse squeeze flow simulation is shown. The volume increase in the gap causes forces the viscoelastic fluid to flow inwards the gap. In the context of hemming, the result demonstrates the potential impact spring-back may have on the adhesive joint.

3. RESULTS AND DISCUSSION

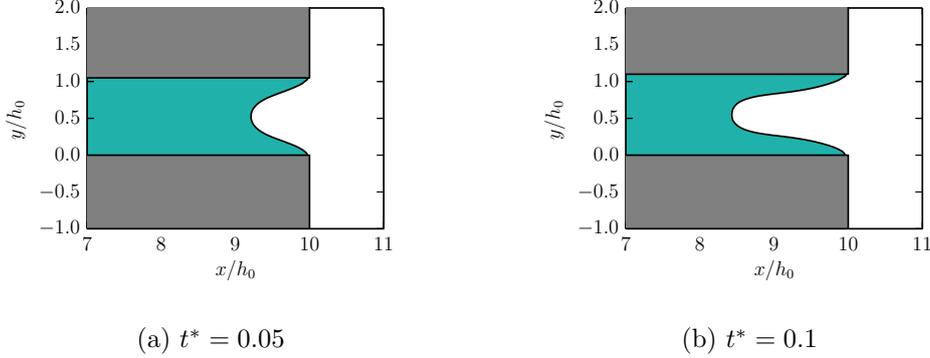


Figure 3.29: Snapshots of reverse squeeze flow simulation with the Oldroyd-B model for $Wi = 1$ with the viscoelastic fluid (green) and the air (white) as visualized for $\alpha = 0.5$.

The computed loads on the upper plate, i.e. the integrated stresses over the solid surface, are shown in Figure 3.30 for the complete range of Wi considered. The loads are normalized by $W_T = 2\mu VL^3/(3h_0^3)$. For the squeeze flow, the theoretical asymptotic loads for $Wi \rightarrow 0$ and $Wi \rightarrow \infty$ are included, corresponding to Newtonian fluids with viscosities $\mu + \eta$ and η , respectively. The loads for the squeeze flow agree with the theoretical predictions and with similar numerical observations reported in the literature by Debbaut (2001) and Phan-Thien *et al.* (1985). The reverse squeeze flow loads show similar trends in terms of the spread for different Wi and the initial transients, which occur on the order of a relaxation time. After the initial transient, the loads decrease with time.

As a next step, the hemming process is studied numerically for a test coupon geometry, with simulated adhesive bead geometries as input. So far, all solid objects in the simulations have been rigid objects. However, due to the nature of the hemming process, the deformation of sheet metal parts is included in the simulations. The deformation is therefore computed using the in-house FEM-based structural mechanics solver LaStFEM. In this work, one-way coupled FSI simulations are used. Thus, solids act on the fluid through immersed boundary conditions, while the fluid forces do not act on the solids.

A geometry designed to be representative for the hemming process is used, shown in Figure 3.31, for which a two-dimensional cross section is simulated. The process consists of a parts assembly step, a pre-hemming step and a final hemming step. The

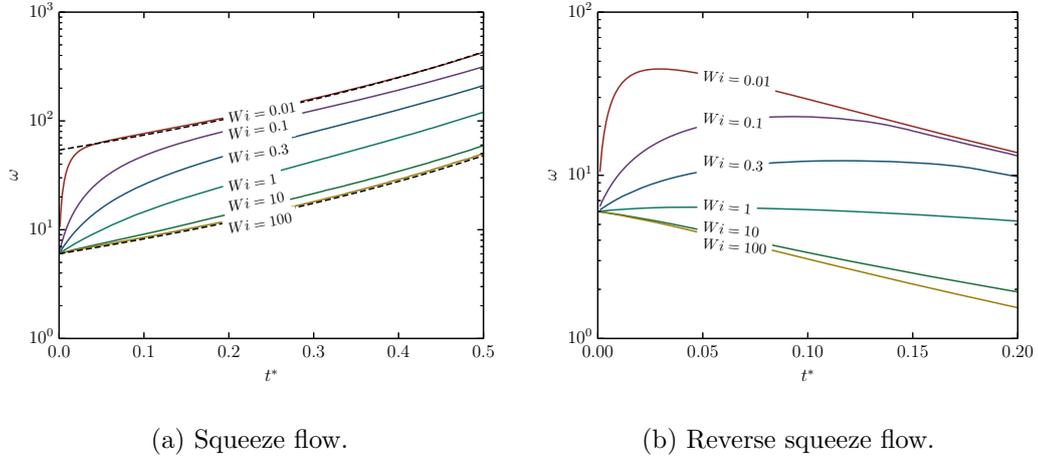


Figure 3.30: Computed loads for the Oldroyd-B model and asymptotic loads (dashed) for $Wi \rightarrow 0$ and $Wi \rightarrow \infty$.

hemming tools are modeled by rigid objects which drive the deformation of the sheet metal parts, which in turn drives the viscoelastic adhesive flow. The initial state of the two-dimensional case is shown in Figure 3.32. During the parts assembly the inner structure moves downward with velocity 10 mm/s until the gap between the sheets is 0.25 mm thick. The tools performing the hemming move with 50 mm/s velocity. The adhesive bead has been extruded with $d_{nom} = 3$ mm.

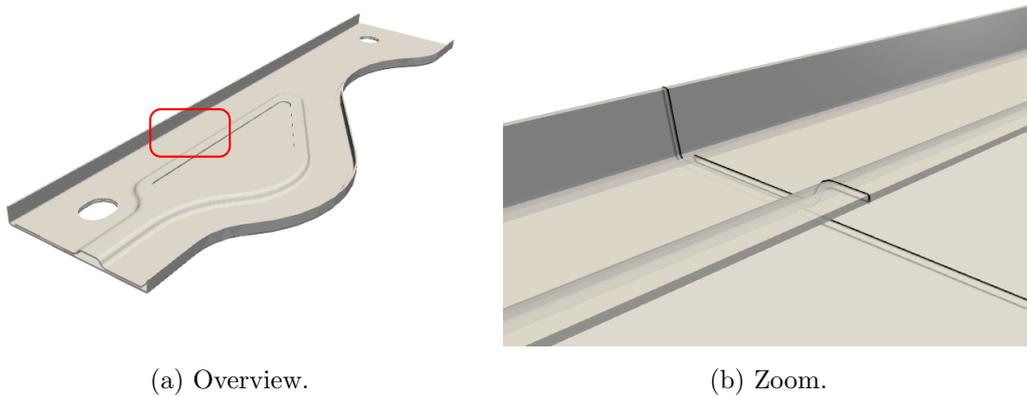


Figure 3.31: Hemming case 3D-geometry (a) overview and (b) zoomed view of cross section used for simulation.

3. RESULTS AND DISCUSSION

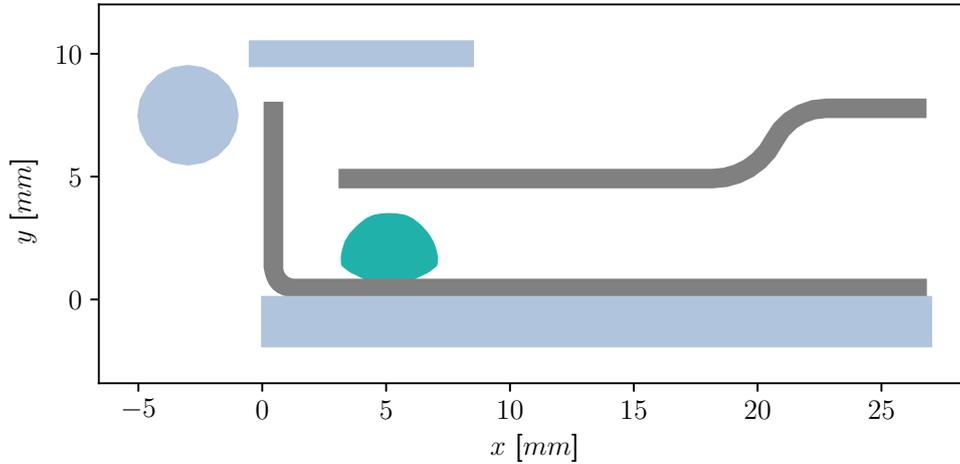


Figure 3.32: Hemming case with sheet metal parts (gray), rigid tools (blue) and initial adhesive geometry (green).

Figure 3.33 shows the results of a hemming simulation. In this particular simulation, the adhesive bead is initially positioned with its center 4.5 mm from the flange, i.e. the vertical part of the outer structure. The results visualize how the deformation of the sheet metal parts drive the adhesive flow, including in the thin gaps, indicating the complexity the numerical simulation.

In Figure 3.34 the results of the hemming simulations for respective bead placements 4.5 mm and 6 mm from the flange are compared to experimental results. The experiments have been performed for the same bead diameter and placements. The experimental results consist of photographs from above, displaying the squeeze-out of adhesive outside the joint. To clarify the comparison, the squeeze-out zone has been highlighted in the figures. In the simulations, the effect of the bead placement on the adhesive distribution in the joint is clear. As expected, a bead placement closer to the flange results in a larger amount of adhesive being squeezed out. The same observation can be made from the experiments. While a quantitative comparison between the simulation and the experiments is difficult, they are in reasonable qualitative agreement as they show the same trend. A more thorough comparison to experimental results is planned to be conducted in the near future. However, at this time the results are an indication that the numerical method is capable to simulate hemming flows.

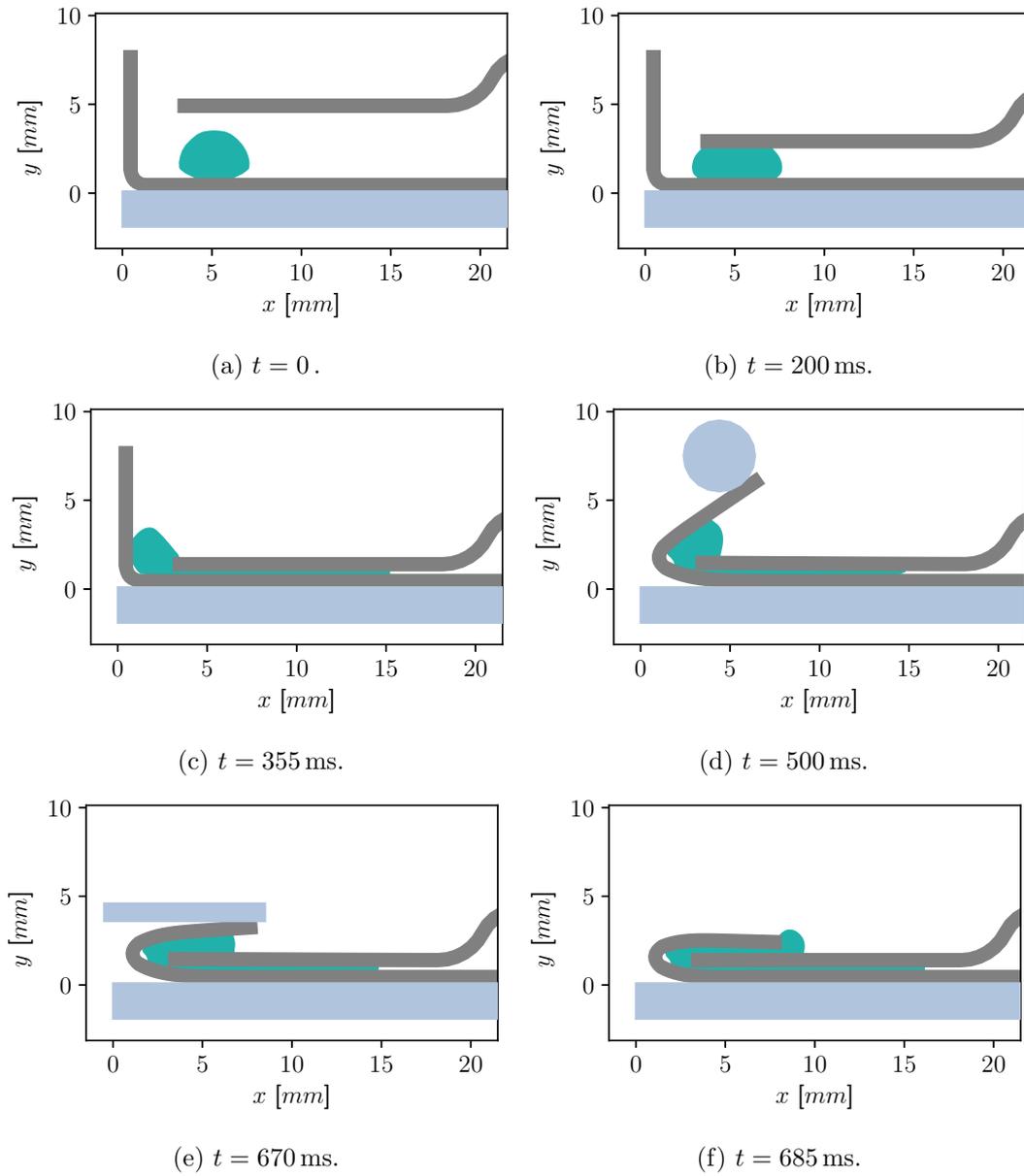


Figure 3.33: Hemming simulation with adhesive placed 4.5 mm from flange.

3. RESULTS AND DISCUSSION

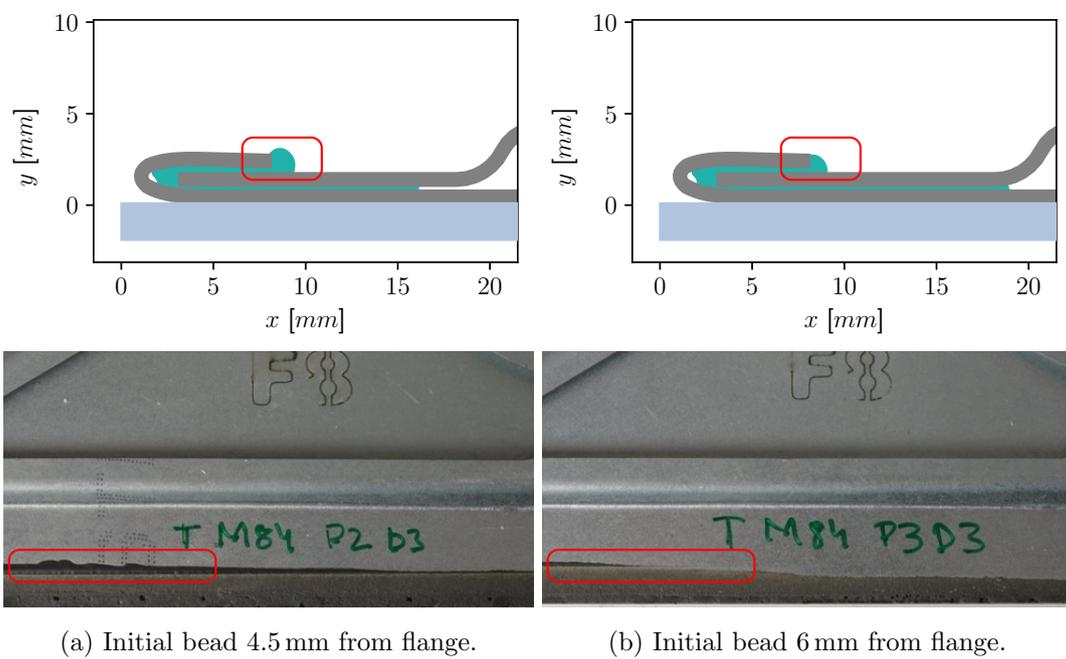


Figure 3.34: Comparison between simulation (top) and experimental hemming (bottom) with squeeze-out zone highlighted (red box), experiments courtesy of RISE IVF.

Chapter 4

Conclusions

Viscoelastic flows are important for many industrial processes, which often involve complex geometry, moving objects and free-surface flow. As such, they are challenging to simulate, which calls for efficient and user-friendly numerical methods. In this thesis, a new Lagrangian–Eulerian framework for viscoelastic flow has been presented. An application of particular interest has been the adhesive joining process, including the adhesive extrusion along a path as well as the subsequent joining step. For this purpose, the numerical framework includes an injection model to model the inflow of the viscoelastic adhesive from a nozzle moving along an arbitrary path.

Two versions of the Lagrangian method to solve the viscoelastic constitutive equations have been proposed, denoted the forwards-tracking and the backwards-tracking method, respectively. The latter was developed as an improvement of the former, enhancing the overall robustness as well as the support for refined computational grids and free-surface flow.

The proposed framework has been validated for relevant benchmarks of viscoelastic single-phase as well as free-surface flow. The results have been found to be in good agreement with available analytic, numerical and experimental data from the literature. Furthermore, the computational performance aspects of the proposed framework have been investigated by comparing computational costs for different implementations of the Lagrangian algorithm to calculate the viscoelastic stress. An important conclusion was that, due to the parallel properties of the Lagrangian constitutive equation, a substantial increase in computational speed was gained by GPU-acceleration of the Lagrangian stress algorithm. Furthermore, it was noted that the backwards-tracking method inherently constitute a further improvement of robustness and computational

4. CONCLUSIONS

cost compared to the forwards-tracking method. This follows from the structured arrangement of the Lagrangian stress solution.

Adhesive beads simulated with different extrusion angles were compared to 3D-scanned experimental beads. The results were found to be in very good agreement with the experimental data. Furthermore, the adhesive extrusion onto an industrial product part along a robot path was simulated. It is noted that, to the best of the authors knowledge, such combination of a product geometry, nozzle path as well as complex fluid rheology has not previously been reported. The ability to perform such simulations is attributed to the unique combination of the immersed boundary octree framework, the coupling between the Lagrangian and Eulerian solution fields, the treatment of the viscoelastic stress at fluid interfaces and the novel injection model. Also for this case, good agreement with 3D-scanned experimental data was observed.

Following the adhesive extrusion, the subsequent joining through hemming was simulated, in which the adhesive geometries were initialized from adhesive extrusion simulations. As such, the simulations demonstrated the important connection between the simulations of the different steps in the joining process. Specifically, this connection can be taken into account with the current numerical framework. Furthermore, the results of the hemming simulations were found to be in qualitative agreement with the available experimental results.

The outlook on future research of the Lagrangian–Eulerian framework includes further case studies as well as introduction of new applications. For adhesive joining flows, an important future development is to enable simulation of two-way coupled fluid–structure interaction. This is important to capture e.g. spring-back effects in hemming applications. Furthermore, additional injection models can be introduced. In a performance context, a GPU-accelerated implementation of the backwards-tracking method constitutes a natural step to explore and improve the performance of the algorithm.

In summary, the results in this thesis have shown that the Lagrangian–Eulerian approach in the presented framework is feasible for the considered flows and suitable for the intended applications, including industrial adhesive joining flows. Thus, the results enable future research and development of numerical tools for applications with viscoelastic flow. The framework will be available for research project partners. Moreover, the adhesive extrusion simulation tool is already available for industrial end-users.

The simulation technology developed within the scope of this thesis may therefore aid virtual processes in manufacturing industry already in the near future.

4. CONCLUSIONS

References

- AFONSO, A.M., PINHO, F.T. & ALVES, M.A. (2012). The kernel-conformation constitutive laws. *Journal of Non-Newtonian Fluid Mechanics*, **167-168**, 30–37. 18
- ALVES, M., PINHO, F. & OLIVEIRA, P. (2001). The flow of viscoelastic fluids past a cylinder: finite-volume high-resolution methods. *Journal of Non-Newtonian Fluid Mechanics*, **97**, 207 – 232. 1, 40
- ALVES, M., OLIVEIRA, P. & PINHO, F. (2021). Numerical methods for viscoelastic fluid flows. *Annual Review of Fluid Mechanics*, **53**, 509–541. 1, 17, 18, 19
- ALVES, M.A., OLIVEIRA, P.J. & PINHO, F.T. (2003). Benchmark solutions for the flow of Oldroyd-B and PTT fluids in planar contractions. *Journal of Non-Newtonian Fluid Mechanics*, **110**, 45 – 75. 1, 2, 19
- ANDERSSON, T., NOWAK, D., JOHNSON, T., MARK, A., EDELVIK, F. & KÜFER, K.H. (2018). Multiobjective Optimization of a Heat-Sink Design Using the Sandwiching Algorithm and an Immersed Boundary Conjugate Heat Transfer Solver. *Journal of Heat Transfer*, **140**, 102002. 6
- BAAIJENS, H.P., PETERS, G.W., BAAIJENS, F.P. & MEIJER, H.E. (1995). Viscoelastic flow past a confined cylinder of a polyisobutylene solution. *Journal of Rheology*, **39**, 1243 – 1277. 1, 40, 41, 42
- BALCI, N., THOMASES, B., RENARDY, M. & DOERING, C.R. (2011). Symmetric factorization of the conformation tensor in viscoelastic fluid models. *Journal of Non-Newtonian Fluid Mechanics*, **166**, 546 – 553, xVIth International Workshop on Numerical Methods for Non-Newtonian Flows. 18

REFERENCES

- BARNES, H.A., HUTTON, J.F. & WALTERS, K. (1989). *An Introduction to Rheology*. Rheology Series, Vol. 3 Elsevier, New York. 6, 7, 10, 15, 46
- BELL, N. & HOBEROCK, J. (2011). Thrust: A productivity-oriented library for cuda. In W. mei W. Hwu, ed., *GPU Computing Gems*, chap. 26, 359–371. 34
- BIRD, R., CURTISS, C., ARMSTRONG, R. & HASSAGER, O. (1987a). *Dynamics of polymeric liquids Vol. 2 Kinetic Theory*. John Wiley and Sons Inc., New York, NY. 12
- BIRD, R.B., ARMSTRONG, R. & HASSAGER, O. (1987b). *Dynamics of polymeric liquids Vol. 1: Fluid mechanics*, vol. 1. John Wiley and Sons Inc., New York, NY, 2nd edn. 6, 10, 11, 12, 13
- BONITO, A., PICASSO, M. & LASO, M. (2006). Numerical simulation of 3d viscoelastic flows with free surfaces. *Journal of Computational Physics*, **215**, 691 – 716. 3, 49
- CHEN, X., MARSCHALL, H., SCHÄFER, M. & BOTHE, D. (2013). A comparison of stabilisation approaches for finite-volume simulation of viscoelastic fluid flow. *International Journal of Computational Fluid Dynamics*, **27**, 229–250. 14, 17
- CHILCOTT, M. & RALLISON, J. (1988). Creeping flow of dilute polymer solutions past cylinders and spheres. *Journal of Non-Newtonian Fluid Mechanics*, **29**, 381 – 432. 13
- COMMINAL, R., PIMENTA, F., HATTEL, J.H., ALVES, M.A. & SPANGENBERG, J. (2018). Numerical simulation of the planar extrudate swell of pseudoplastic and viscoelastic fluids with the streamfunction and the vof methods. *Journal of Non-Newtonian Fluid Mechanics*, **252**, 1 – 18. 2, 46, 49
- CROCHET, M. & KEUNINGS, R. (1982). Finite element analysis of die swell of a highly elastic fluid. *Journal of Non-Newtonian Fluid Mechanics*, **10**, 339 – 356. 2, 46, 47, 49
- DAWES, B. & ABRAHAM, D. (2020). Boost C++ libraries, <https://www.boost.org/>. 36

- DE PAULO, G., TOMÉ, M. & MCKEE, S. (2007). A marker-and-cell approach to viscoelastic free surface flows using the ptt model. *Journal of Non-Newtonian Fluid Mechanics*, **147**, 149 – 174. 2, 49
- DEBBAUT, B. (2001). Non-isothermal and viscoelastic effects in the squeeze flow between infinite plates. *Journal of Non-Newtonian Fluid Mechanics*, **98**, 15–31. 60
- DOORMAAL, J.P.V. & RAITBY, G.D. (1984). Enhancements of the simple method for predicting incompressible fluid flows. *Numerical Heat Transfer*, **7**, 147–163. 23
- EDELVIK, F., MARK, A., KARLSSON, N., JOHNSON, T. & CARLSON, J. (2017). Math-based algorithms and software for virtual product realization implemented in automotive paint shops. In L. Ghezzi, D. Hömberg & C. Landry, eds., *Math for the Digital Factory*, 231–251, Springer-Verlag, Berlin. 6
- FATTAL, R. & KUPFERMAN, R. (2004). Constitutive laws for the matrix-logarithm of the conformation tensor. *Journal of Non-Newtonian Fluid Mechanics*, **123**, 281 – 285. 17
- FATTAL, R. & KUPFERMAN, R. (2005). Time-dependent simulation of viscoelastic flows at high weissenberg number using the log-conformation representation. *Journal of Non-Newtonian Fluid Mechanics*, **126**, 23 – 37. 17
- GALLEZ, X., HALIN, P., LIELENS, G., KEUNINGS, R. & LEGAT, V. (1999). The adaptive lagrangian particle method for macroscopic and micro–macro computations of time-dependent viscoelastic flows. *Computer Methods in Applied Mechanics and Engineering*, **180**, 345 – 364. 2, 19
- GÖHL, J., MARKSTEDT, K., MARK, A., HÅKANSSON, K., GATENHOLM, P. & EDELVIK, F. (2018). Simulations of 3d bioprinting: predicting bioprintability of nanofibrillar inks. *Biofabrication*, **10**. 6
- GUTTMAN, A. (1984). R-trees: A dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, SIGMOD '84, 47–57, Association for Computing Machinery, New York, NY, USA. 30

REFERENCES

- HABLA, F., MARSCHALL, H., HINRICHSSEN, O., DIETSCHKE, L., JASAK, H. & FAVERO, J.L. (2011). Numerical simulation of viscoelastic two-phase flows using openfoam®. *Chemical Engineering Science*, **66**, 5487 – 5496. 2, 46, 49
- HALIN, P., LIELENS, G., KEUNINGS, R. & LEGAT, V. (1998). The lagrangian particle method for macroscopic and micro–macro viscoelastic flow computations dedicated to professor marcel j. crochet on the occasion of his 60th birthday.1. *Journal of Non-Newtonian Fluid Mechanics*, **79**, 387 – 403. 2, 19
- HARLEN, O., RALLISON, J. & SZABO, P. (1995). A split lagrangian-eulerian method for simulating transient viscoelastic flows. *Journal of Non-Newtonian Fluid Mechanics*, **60**, 81 – 104. 2, 19
- HERRCHEN, M. & ÖTTINGER, H.C. (1997). A detailed comparison of various FENE dumbbell models. *Journal of Non-Newtonian Fluid Mechanics*, **68**, 17 – 42. 13
- HINDMARSH, A.C., SERBAN, R. & REYNOLDS, D.R. (2018). *User Documentation for ccode v3.1.0 (sundials v3.1.0)*. Sundials. 33, 35
- HULSEN, M.A., FATTAL, R. & KUPFERMAN, R. (2005). Flow of viscoelastic fluids past a cylinder at high weissenberg number: Stabilized simulations using matrix logarithms. *Journal of Non-Newtonian Fluid Mechanics*, **127**, 27 – 39. 1
- ISKE, A. (2004). *Multiresolution Methods in Scattered Data Modelling*, vol. 37 of *Lecture notes in computational science and engineering*. Springer, 1st edn. 27, 29
- IZBASSAROV, D. & MURADOGLU, M. (2015). A front-tracking method for computational modeling of viscoelastic two-phase flow systems. *Journal of Non-Newtonian Fluid Mechanics*, **223**, 122 – 140. 3
- KEUNINGS, R. (2000). A survey of computational rheology. In *Proceedings of the XIIIth International Congress on Rheology*, vol. 1, 7–14, Citeseer. 16, 17
- LARSON, R. (1999). *The Structure and Rheology of Complex Fluids*. Topics in Chemical Engineering, OUP USA. 15
- LARSON, R.G. (1988). *Constitutive Equations for Polymer Melts and Solutions*. Butterworths series in chemical engineering, Butterworth Publishers. 11, 12

- LIKHTMAN, A.E. & GRAHAM, R.S. (2003). Simple constitutive equation for linear polymer melts derived from molecular theory: Rolie–poly equation. *Journal of Non-Newtonian Fluid Mechanics*, **114**, 1 – 12. 13
- LODGE, A. (1974). *Body Tensor Fields in Continuum Mechanics: With Applications to Polymer Rheology*. Academic Press. 10
- MARK, A. & VAN WACHEM, B.G.M. (2008). Derivation and validation of a novel implicit second-order accurate immersed boundary method. *J. of Comput. Physics*, **227**, 6660 – 6680. 6, 23
- MARK, A., RUNDQVIST, R. & EDELVIK, F. (2011). Comparison between different immersed boundary conditions for simulation of complex fluid flows. *Fluid dynamics & materials processing*, **7**, 241–258. 6, 23
- MARK, A., SVENNING, E. & EDELVIK, F. (2013). An immersed boundary method for simulation of flow with heat transfer. *International Journal of Heat and Mass Transfer*, **56**, 424 – 435. 6
- MARK, A., BOHLIN, R., SEGERDAHL, D., EDELVIK, F. & CARLSON, J.S. (2014). Optimisation of robotised sealing stations in paint shops by process simulation and automatic path planning. *International Journal of Manufacturing Research*, **9**, 4–26. 6
- MCLEISH, T.C.B. & LARSON, R.G. (1998). Molecular constitutive equations for a class of branched polymers: The pom-pom polymer. *Journal of Rheology*, **42**, 81–110. 13
- MITSOULIS, E. (2013). 50 years of the k-bkz constitutive relation for polymers. *ISRN Polymer Science*. 13
- MOROZOV, A. & SPAGNOLIE, S.E. (2015). *Introduction to Complex Fluids*, 3–52. Springer New York, New York, NY. 10, 12, 14
- NIETHAMMER, M., BRENN, G., MARSCHALL, H. & BOTHE, D. (2019). An extended volume of fluid method and its application to single bubbles rising in a viscoelastic liquid. *Journal of Computational Physics*, **387**, 326 – 355. 2, 32

REFERENCES

- NOWAK, D., JOHNSON, T., MARK, A., IREHOLM, C., PEZZOTTI, F., ERHARDSSON, L., STÅHLBERG, D., EDELVIK, F. & KÜFER, K.H. (2020). Multicriteria Optimization of an Oven With a Novel ε -Constraint-Based Sandwiching Method. *Journal of Heat Transfer*, **143**, 012101. 6
- OISHI, C., MARTINS, F., TOMÉ, M., CUMINATO, J. & MCKEE, S. (2011). Numerical solution of the extended pom-pom model for viscoelastic free surface flows. *Journal of Non-Newtonian Fluid Mechanics*, **166**, 165 – 179. 2, 46
- OISHI, C.M., TOMÉ, M.F., CUMINATO, J.A. & MCKEE, S. (2008). An implicit technique for solving 3d low reynolds number moving free surface flows. *Journal of Computational Physics*, **227**, 7446 – 7468. 2, 46, 49
- OLDROYD, J. (1950). On the formulation of rheological equations of state. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 200, 523–541, The Royal Society. 10
- OLIVEIRA, P., PINHO, F. & PINTO, G. (1998). Numerical simulation of non-linear elastic flows with a general collocated finite-volume method. *Journal of Non-Newtonian Fluid Mechanics*, **79**, 1 – 43. 1, 40
- OWENS, R. & PHILLIPS, T. (2002). *Computational Rheology*. Computational Rheology, Imperial College Press. 17
- PALHARES JUNIOR, I.L., OISHI, C.M., AFONSO, A.M., ALVES, M.A. & PINHO, F.T. (2016). Numerical study of the square-root conformation tensor formulation for confined and free-surface viscoelastic fluid flows. *Advanced Modeling and Simulation in Engineering Sciences*, **3**, 2. 18
- PATANKAR, S.V. (1980). *Numerical Heat Transfer and Fluid Flow*. Hemisphere Publishing Corporation. 23
- PHAN-THIEN, N., DUDEK, J., BOGER, D. & TIRTAATMADJA, V. (1985). Squeeze film flow of ideal elastic liquids. *Journal of Non-Newtonian Fluid Mechanics*, **18**, 227–254. 60
- PHAN-THIEN, N. (1978). A nonlinear network viscoelastic model. *Journal of Rheology*, **22**, 259–283. 12

- PILLAPAKKAM, S. & SINGH, P. (2001). A level-set method for computing solutions to viscoelastic two-phase flow. *Journal of Computational Physics*, **174**, 552–578. 3
- PIMENTA, F. & ALVES, M. (2017). Stabilization of an open-source finite-volume solver for viscoelastic fluid flows. *Journal of Non-Newtonian Fluid Mechanics*, **239**, 85–104. 1
- POOLE, R.J. (2012). The Deborah and Weissenberg numbers. In *Rheology bulletin*, vol. 53, 32 – 39, British Society of Rheology. 16
- RASMUSSEN, H. & HASSAGER, O. (1995). Simulation of transient viscoelastic flow with second order time integration. *Journal of Non-Newtonian Fluid Mechanics*, **56**, 65 – 84. 2, 19
- REINER, M. (1964). The Deborah number. *Physics*, **17**, 62. 16
- SCHLICHTING, H. & GERSTEN, K. (2000). *Boundary Layer Theory*. Springer-Verlag Berlin Heidelberg, 8th edn. 8, 9, 15
- SPANJAARDS, M., HULSEN, M. & ANDERSON, P. (2019). Transient 3d finite element method for predicting extrudate swell of domains containing sharp edges. *Journal of Non-Newtonian Fluid Mechanics*, **270**, 79 – 95. 2, 46
- STEWART, P.A., LAY, N., SUSSMAN, M. & OHTA, M. (2008). An improved sharp interface method for viscoelastic and viscous two-phase flows. *Journal of Scientific Computing*, **35**, 43–61. 3, 18
- SUNDIALS (2020). Suite of nonlinear and differential/algebraic equation solvers cvode, <http://computation.llnl.gov/projects/sundials/cvode>. 35
- SVENNING, E., MARK, A. & EDELVIK, F. (2014). Simulation of a highly elastic structure interacting with a two-phase flow. *Journal of Mathematics in Industry*, **4**, 7. 6
- SVENSSON, M., MARK, A., EDELVIK, F., KRESSIN, J., BOHLIN, R., SEGERDAHL, D., CARLSON, J.S., WAHLBORG, P.J. & SUNDBÄCK, M. (2016). Process simulation and automatic path planning of adhesive joining. *Procedia CIRP*, **44**, 298 – 303, 6th CIRP Conference on Assembly Technologies and Systems (CATS). 6

REFERENCES

- TAHIR-KHELI, R. (2018). *Ordinary Differential Equations. [electronic resource] : Mathematical Tools for Physicists..* Springer International Publishing. 34
- TANNER, R.I. (1970). A theory of die-swell. *Journal of Polymer Science Part A-2: Polymer Physics*, **8**, 2067–2078. 47
- TANNER, R.I. (2005). A theory of die-swell revisited. *Journal of Non-Newtonian Fluid Mechanics*, **129**, 85 – 87. 47
- THIEN, N.P. & TANNER, R.I. (1977). A new constitutive equation derived from network theory. *Journal of Non-Newtonian Fluid Mechanics*, **2**, 353–365. 12
- TOMÉ, M., MANGIAVACCHI, N., CUMINATO, J., CASTELO, A. & MCKEE, S. (2002). A finite difference technique for simulating unsteady viscoelastic free surface flows. *Journal of Non-Newtonian Fluid Mechanics*, **106**, 61 – 106. 2, 46, 49
- TOMÉ, M., CASTELO, A., FERREIRA, V. & MCKEE, S. (2008). A finite difference technique for solving the Oldroyd-B model for 3d-unsteady free surface flows. *Journal of Non-Newtonian Fluid Mechanics*, **154**, 179 – 206. 2, 46, 49
- TOMÉ, M., PAULO, G., PINHO, F. & ALVES, M. (2010). Numerical solution of the PTT constitutive equation for unsteady three-dimensional free surface flows. *Journal of Non-Newtonian Fluid Mechanics*, **165**, 247 – 262. 2, 46, 49
- TOMÉ, M., CASTELO, A., AFONSO, A., ALVES, M. & PINHO, F. (2012). Application of the log-conformation tensor to three-dimensional time-dependent free surface flows. *Journal of Non-Newtonian Fluid Mechanics*, **175–176**, 44 – 54. 2, 46, 49
- TRUESDELL, C. & RAJAGOPAL, K. (1999). *An Introduction to the Mechanics of Fluids.* Birkhäuser Boston. 9
- TRYGGVASON, G., SCARDOVELLI, R. & ZALESKI, S. (2011). *Direct Numerical Simulations of Gas-Liquid Multiphase Flows..* Cambridge University Press. 23
- UBBINK, O. & ISSA, R. (1999). A method for capturing sharp fluid interfaces on arbitrary meshes. *Journal of Computational Physics*, **153**, 26 – 50. 23
- IPS IBOFLOW (2022). IPS IBOFlow, <http://ipsiboflow.com>. 6

- WAPPEROM, P., KEUNINGS, R. & LEGAT, V. (2000). The backward-tracking lagrangian particle method for transient viscoelastic flows. *Journal of Non-Newtonian Fluid Mechanics*, **91**, 273 – 295. 2, 19, 30
- WHITE, J.L. (1964). Dynamics of viscoelastic fluids, melt fracture, and the rheology of fiber spinning. *Journal of Applied Polymer Science*, **8**, 2339–2357. 16
- XUE, S.C., TANNER, R. & PHAN-THIEN, N. (2004). Numerical modelling of transient viscoelastic flows. *Journal of Non-Newtonian Fluid Mechanics*, **123**, 33 – 58. 17
- ZOGRAFOS, K., AFONSO, A.M., POOLE, R.J. & OLIVEIRA, M.S. (2020). A viscoelastic two-phase solver using a phase-field approach. *Journal of Non-Newtonian Fluid Mechanics*, **284**, 104364. 3