Numerical Simulations of Viscoelastic Flows Using the Discontinuous Galerkin Method

John Taylor Burleson

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Mathematics

Pengtao Yue, Chair

Traian Iliescu

Peter Wapperom

July 21, 2021

Blacksburg, Virginia

Keywords: viscoelastic fluids, Oldroyd-B model, finite element method, discontinuous Galerkin method, cavity flow

Numerical Simulations of Viscoelastic Flows Using the Discontinuous Galerkin Method

John Taylor Burleson

(ABSTRACT)

In this work, we develop a method for solving viscoelastic fluid flows using the Navier-Stokes equations coupled with the Oldroyd-B model. We solve the Navier-Stokes equations in skew-symmetric form using the mixed finite element method, and we solve the Oldroyd-B model using the discontinuous Galerkin method. The Crank-Nicolson scheme is used for the temporal discretization of the Navier-Stokes equations in order to achieve a second-order accuracy in time, while the optimal third-order total-variation diminishing Runge-Kutta scheme is used for the temporal discretization of the Oldroyd-B equation. The overall accuracy in time is therefore limited to second-order due to the Crank-Nicolson scheme; however, a third-order Runge-Kutta scheme is implemented for greater stability over lower order Runge-Kutta schemes. We test our numerical method using the 2D plane Poiseuille benchmark problem and the 2D cavity flow benchmark problem and compare results generated with those found in literature while discussing the influence of mesh refinement on suppressing oscillations in the polymer stress.

Numerical Simulations of Viscoelastic Flows Using the Discontinuous Galerkin Method

John Taylor Burleson

(GENERAL AUDIENCE ABSTRACT)

Viscoelastic fluids are a type of non-Newtonian fluid of great importance to the study of fluid flows. Such fluids exhibit both viscous and elastic behaviors. We develop a numerical method to solve the partial differential equations governing viscoelastic fluid flows using various finite element methods. Our method is then validated using previous numerical results in literature.

Dedication

To my family and friends.

Acknowledgments

I would like to thank my advisor, Professor Pengtao Yue, for his ineffable patience and wisdom, and his guidance in developing my ability as a researcher.

I would also like to thank my thesis committee: Professor Traian Iliescu and Professor Peter Wapperom for sharing their knowledge.

Lastly I would like to thank the developers of the deal.II library, whose open source finite element library was crucial to the work of this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Fluid modeling is a field both incredibly broad and important. A considerable number of materials in industrial operations are processed in liquid form. Additionally, fluid modeling is incredibly complex, often requiring specialized models with little theoretical grounding, which makes the continuing study of fluid flows of great importance, and there are many open problems in the field. Not only are fluids comprised of a single molecule type studied, but suspensions of many particle types are of interest.

1.1 Non-Newtonian Fluids

Non-Newtonian fluids are a class of fluids where the relationship between stress and deformation rate is nonlinear.

In contrast, Newtonian fluids, also known as simple fluids, are fluids that respond linearly to shear stress forces [26]. Examples of Newtonian fluids are water, air, and alcohol.

Examples of Non-Newtonian fluids are pseudoplastics, dilatants, Bingham plastics, Bingham pseudoplastics, viscoplastics, and viscoelastics. Pseudoplastics exhibit a behavior where a larger shear rate will not necessarily induce a larger shear stress, which is exemplified by latex paint. As the application of latex paint creates a larger shear rate, the shear stress does not increase, allowing for easy application. In other words, small increases in shear

stress yield large increases in shear rate, which translates to flow. Dilatants differ from pseudoplastics in that the relationship between shear rate and shear stress is reversed. Large increases in shear stress will induce only small increases in shear rate, which can be seen in mixtures of water and corn starch. Bingham plastics and pseudoplastics are quite different from the others in that a yield stress is required in order for flow to even occur. An everyday example of a Bingham plastic is toothpaste. Toothpaste will not naturally flow out of the tube without a force being applied first.

1.2 Viscoelastic Fluids

Viscoelastic fluids are non-Newtonian fluids that exhibit both elastic and viscous responses, where the elasticity can come from the polymers suspended in the fluid or the polymer molecules in a polymer melt. Viscoelastic fluids are of special interest in the manufacturing of materials, as polymer solvent solutions are commonplace. There are many different models governing the viscoelasticity behavior of these fluids including the Maxwell model and the Oldroyd-B model, the latter being an extension of the former with an additional viscous solvent.

There are many interesting phenomena observed in viscoelastic fluids such as the Weissenberg effect [6], the Uebler effect [21], the Barus effect [20], and the Kaye effect [18]. The Weissenberg effect, also known as rod-climbing, is an effect detailed in [6] where a cylindrical rod spun in a pool of viscoelastic fluid causes that fluid to climb up the rod. This is due to the normal stress induced by shearing of the fluid. The normal stress wrapping around the rod, also known as the hoop stress, squeezes the fluid to move upward along the rod.

Models for viscoelastic fluids invariably involve models for the polymer stress, a tensor that arises from the action of polymer molecules to maintain a state of highest-entropy i.e.

lowest energy [26]. Different models have different approaches to describing the behavior of these polymeric molecules. An early model for polymer stress was the Oldroyd-B model, proposed by Oldroyd in 1950 [22]. This model assumes that the polymer behavior can be modeled by beads connected by elastic springs. This model has an unphysical singularity in steady extensional flow wherein the spring may get infinitely stretched, which has led to the development of the finitely extensible nonlinear elastic (FENE) model [26]. However, it is often the case that this singularity remains integrable, so a steady state solution can be found, which is illustrated by the flow through cross-slot geometry done in [25].

Other models include the Giesekus model [14] and the Phan-Thien-Tanner models [29] for viscoelastic fluids. The Giesekus model introduces a nonlinear quadratic polymer stress term, while the Phan-Thien-Tanner model introduces either a linear or an exponential relaxation function.

1.3 Weissenberg Number Problem

The Weissenberg number is a dimensionless quantity determined by $Wi = \lambda\dot{\gamma}$ where λ is the relaxation time and $\dot{\gamma}$ is the strain rate [5]. Thus the Weissenberg number can be viewed as the ratio of the relaxation time to the flow time. In steady simple shear flow, the Weissenberg number also represents the ratio of elastic to viscous stress, where the dominant elastic stress difference is characterized by the first normal stress difference $\tau_{xx} - \tau_{yy}$ and viscous stress is characterized by shear stress τ_{xy} [24]. The Maxwell model then yields $\tau_{xx} - \tau_{yy} = 2\lambda\eta\dot{\gamma}^2$ and $\tau_{xy} = \eta\dot{\gamma}$, where η is the polymer viscosity. Thus

$$Wi = \lambda\dot{\gamma} = \frac{1}{2}\frac{\tau_{xx} - \tau_{yy}}{\tau_{xy}} \tag{1.1}$$

The Weissenberg number is of interest for solving fluid flow problems using the Oldroyd-B model given that numerical methods generally fail for $Wi = O(1)$ [9]. This failure arises in any method that uses polynomial basis functions to interpolate polymer stress, given the exponential growth of the stress [12].

In order to resolve this exponential nature of stress, two basic solutions arise: use exponential basis functions for interpolation or take a matrix logarithm of stress. The issue with these solutions would be the necessity that stress be strictly positive, which is not always the case. Instead, the conformation tensor, which is symmetric positive-definite, is used. Since the conformation tensor is SPD, the matrix-logarithm is well defined. Thus, the exponential growth of the stress can be transformed into linear growth, making it ideal for interpolation using polynomial functions. To do that, the constitutive equations must be written in terms of the conformation tensor.

This is done using the relation between the dimensionless polymer stress and conformation tensor as given below:

$$\boldsymbol{\tau} = \frac{g(\boldsymbol{\sigma})}{Wi}(\boldsymbol{\sigma} - \boldsymbol{I}), \tag{1.2}$$

where $g(\boldsymbol{\sigma})$ is a scalar-valued function of the conformation tensor $\boldsymbol{\sigma}$.

Taking the Oldroyd-B model as an example, we then have

$$\frac{\partial \boldsymbol{\sigma}}{\partial t} + (\boldsymbol{u} \cdot \boldsymbol{\nabla})\boldsymbol{\sigma} - \boldsymbol{\sigma} \cdot \nabla \boldsymbol{u} - (\nabla \boldsymbol{u})^T \cdot \boldsymbol{\sigma} = \frac{1}{Wi}(\boldsymbol{\sigma} - \boldsymbol{I}) \tag{1.3}$$

From [11], we get that $\nabla \boldsymbol{u}$ can be locally decomposed into

$$\nabla \boldsymbol{u} = \boldsymbol{\Omega} + \boldsymbol{B} + \boldsymbol{N}\boldsymbol{\sigma}^{-1} \tag{1.4}$$

where $\boldsymbol{\Omega}$ and $\boldsymbol{N}$ are anti-symmetric i.e. pure rotations, and $\boldsymbol{B}$ is symmetric, traceless, and commutes with $\boldsymbol{\sigma}$.

Letting $\boldsymbol{\Psi} = \log(\boldsymbol{\sigma})$ it can be shown that $\boldsymbol{\Psi}$ satisfies the following equation [11]:

$$\frac{\partial \boldsymbol{\Psi}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{\Psi} - (\boldsymbol{\Omega}\boldsymbol{\Psi} + \boldsymbol{\Psi}\boldsymbol{\Omega}) - 2\boldsymbol{B} = \frac{1}{Wi} e^{-\boldsymbol{\Psi}}(e^{\boldsymbol{\Psi}} - \boldsymbol{I}) \tag{1.5}$$

Solving (1.3) numerically will allow for solutions for higher Weissenberg numbers. However, this does not allow for problems with arbitrary Weissenberg numbers to become solvable. As observed in [17], for the flow past a cylinder problem, solutions to the Oldroyd-B model, while stable for high Weissenberg numbers, are inaccurate for Weissenberg numbers of order 1 even with the use of the log-conformation tensor. This is due to a fundamental nature of the Oldroyd-B model which allows unlimited growth of polymer stress.

1.4 Numerical Methods for Viscoelastic Fluids

There have been several methods presented for solving viscoelastic fluids. Of note, the Oldroyd-B model is convection dominated: there is no diffusion term to balance $\boldsymbol{u} \cdot \nabla \boldsymbol{\tau_p}$, the convection term. This means that the unmodified Galerkin method for solving partial differential equations will not work since the problem becomes highly hyperbolic. Methods such as streamline-upwind Petrov Galerkin (SUPG) and discontinuous Galerkin have been

implemented by various authors to introduce upwinding to the finite element methods. With the SUPG method, a stabilization term is added to the test function in the direction of the streamline.

Another popular method for handling the hyperbolic term present in viscoelastic models is the discontinuous Galerkin (DG) method wherein the elements used are discontinuous with an applied flux as a correction to retain continuous approximate solutions. This method has several advantages, which have led to its rise in popularity. The convergence order scales with the order of basis polynomials[19]. Another major advantage of using DG is that due to the nature of each element being solved individually, the computation of solutions is highly parallelizable, therefore multithreading can be used to aid in greatly reducing computation times.

However, it is not clear which method is superior to the other, and it can be problem dependent. To compare SUPG and DG, the work of [31] found that in the benchmark problem for axisymmetric steady flows of the modified Chilcott and Rallison fluid through a contraction, the DG method outperformed the SUPG method with more accurate solutions and higher limit for Weissenberg number, and that DG generally outperforms SUPG in cases with singular stress in the computational domain, which given the nature of the Oldroyd-B model is common. In some other problems, SUPG seems to be preferred. For example, for the stability analysis of the planar Couette flow of the upper-convected Maxwell fluid, DG runs into an unphysical instability while SUPG remains stable [3].

1.5 Numerical Methods for Navier-Stokes Flow

Numerical solutions to the Navier-Stokes equations are of great interest given the famously unsolved nature of the equations in general. Two main methods for obtaining numerical

solutions to the incompressible Navier-Stokes equations are the projection and mixed finite element methods.

The first major hurdle in solving the Navier-Stokes equations numerically using finite element methods is the Ladyzhenskaya-Babuška-Brezzi (LBB) condition [7], which necessitates that the polynomial degree of interpolating functions for velocity and pressure be different, often done by choosing polynomial spaces P_n and P_{n-1} for velocity and pressure respectively. The LBB condition arises from the necessity of unique solutions to saddle point problems that generate from solving the momentum equation using finite element methods as is introduced in section 2.5. An alternative method is the so-called projection method [16], which effectively splits the Navier-Stokes equations into two separate steps: first solving for velocity and then solving for pressure. In each step, once velocity is known, the Navier-Stokes equations become the weak form of the Poisson equation for pressure [16]. This effectively sidesteps the need for the LBB condition, i.e. the order of the shape functions for velocity and pressure can be equivalent, which leads to much easier implementation [13]. But it comes with some downsides: the method introduces an artificial Neumann boundary condition not satisfied by the exact solution for pressure [16]. Besides, the projection method does not work for the Stokes equations, a special case of the NS equations with zero inertia.

Coupled with viscoelastic stresses, the Navier-Stokes equations often require additional techniques beyond those listed above. With the addition of elastic stresses due to polymer molecules, traditional methods of solving the Navier-Stokes equations may become unstable. This can occur if the solvent viscosity is too small, leading to the development of the elastic–viscous-split-stress (EVSS) family of methods [3].

Chapter 2

Mathematical Formulation

2.1 Governing Equations

This work aims to model viscoelastic fluids using the Oldroyd-B model coupled with the Navier-Stokes equations for incompressible fluids. Assuming isothermal fluid flow, we have

$$\rho(\boldsymbol{u}_t + (\boldsymbol{u} \cdot \nabla \boldsymbol{u})) = -\nabla P + \nabla \cdot (\boldsymbol{\tau_s} + \boldsymbol{\tau_p}), \tag{2.1}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{2.2}$$

$$\lambda \overset{\nabla}{\boldsymbol{\tau_p}} + \boldsymbol{\tau_p} = \eta_p(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T). \tag{2.3}$$

Here ρ is the fluid density, η_p is the polymer viscosity, λ is the relaxation time, $\boldsymbol{u}$ is the fluid velocity, and P is the pressure.

Further, $\boldsymbol{\tau_s}$ represents the solvent stress, which is given by $\boldsymbol{\tau_s} = \eta_s(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)$ with η_s being the solvent viscosity, and $\boldsymbol{\tau_p}$ is the polymer stress. The $\overset{\nabla}{\boldsymbol{\tau_p}}$ term represents the upper-convected time derivative, given by

$$\overset{\nabla}{\boldsymbol{\tau_p}} = \frac{\partial \boldsymbol{\tau_p}}{\partial t} + (\boldsymbol{u} \cdot \boldsymbol{\nabla})\boldsymbol{\tau_p} - \boldsymbol{\tau_p} \cdot \nabla \boldsymbol{u} - (\nabla \boldsymbol{u})^T \cdot \boldsymbol{\tau_p}$$

2.2 Solving Navier-Stokes Equations

We solve the Navier-Stokes equations decoupled from the Oldroyd-B equation, and pair the solutions at each time step in a manner which will be outlined in Section 2.5.

As outlined in [30], the weak form of the Navier-Stokes equations paired with the incompressibility condition (2.1)-(2.2) can be found by first applying the Galerkin method and multiplying by test functions, which we will call $\hat{\boldsymbol{u}}$ and $\hat{P}$, for velocity and pressure respectively.

$$\left(\rho\left(\frac{\partial \boldsymbol{u}}{\partial t}+\boldsymbol{u}\cdot\nabla\boldsymbol{u}\right),\hat{\boldsymbol{u}}\right)_{\Omega}=\left(\boldsymbol{n}\cdot(-p\boldsymbol{I}+\boldsymbol{\tau}),\hat{\boldsymbol{u}}\right)_{\partial\Omega}-\left(-p\boldsymbol{I}+\boldsymbol{\tau},\nabla\hat{\boldsymbol{u}}\right)_{\Omega}, \tag{2.4}$$

$$-\left(\nabla\cdot\boldsymbol{u},\hat{P}\right)_{\Omega}=0, \tag{2.5}$$

where $\boldsymbol{\tau}=\boldsymbol{\tau_s}+\boldsymbol{\tau_p}$ is the total stress, $(\cdot\,,\cdot)_{\Omega}$ is the inner product on the space L^2 with computational domain Ω and $(\cdot\,,\cdot)_{\partial\Omega}$ is the L^2 inner product on the boundary domain $\partial\Omega$.

We will use the skew-symmetric form of the convection term

$$\boldsymbol{u}\cdot\nabla\boldsymbol{u}=\frac{1}{2}\boldsymbol{u}\cdot\nabla\boldsymbol{u}+\frac{1}{2}\nabla\cdot(\boldsymbol{u}\boldsymbol{u}), \tag{2.6}$$

which holds since

$$\frac{1}{2}\nabla\cdot(\boldsymbol{u}\boldsymbol{u})=\frac{1}{2}\boldsymbol{u}\cdot\nabla\boldsymbol{u}+\frac{1}{2}(\nabla\cdot\boldsymbol{u})\boldsymbol{u} \tag{2.7}$$

and $\nabla\cdot\boldsymbol{u}=0$. We then make the following approximation:

$$\begin{aligned}\boldsymbol{u}\cdot\nabla\boldsymbol{u}\approx\bar{\boldsymbol{u}}\cdot\nabla\boldsymbol{u}&=\frac{1}{2}\bar{\boldsymbol{u}}\cdot\nabla\boldsymbol{u}+\frac{1}{2}\bar{\boldsymbol{u}}\cdot\nabla\boldsymbol{u}+\frac{1}{2}(\nabla\cdot\bar{\boldsymbol{u}})\boldsymbol{u}\\&=\bar{\boldsymbol{u}}\cdot\nabla\boldsymbol{u}+\frac{1}{2}(\nabla\cdot\bar{\boldsymbol{u}})\boldsymbol{u}\end{aligned}\tag{2.8}$$

where $\bar{\boldsymbol{u}}$ is an extrapolated approximation of velocity. Normally, the divergence term would disappear due to the incompressibility condition, but since $\bar{\boldsymbol{u}}$ is only an approximation of $\boldsymbol{u}$, $\nabla\cdot\bar{\boldsymbol{u}}\neq 0$ unless $\bar{\boldsymbol{u}}=\boldsymbol{u}$ exactly. Thus, we have $\boldsymbol{u}\cdot\nabla\boldsymbol{u}\approx\bar{\boldsymbol{u}}\cdot\nabla\boldsymbol{u}+\frac{1}{2}(\nabla\cdot\bar{\boldsymbol{u}})\boldsymbol{u}$ as our skew-symmetric convection term. This skew-symmetric form is known to conserve energy at the discrete level for flows with constant density. Meanwhile, it has the benefit of being linear in $\boldsymbol{u}$ since $\bar{\boldsymbol{u}}$ will be known at every time step.

Applying the following temporal discretization, we seek solutions $\boldsymbol{u}^{n+1}\in H^1(\Omega)^2$ and $P^{n+1/2}\in L^2(\Omega)$ satisfying

$$\begin{aligned}\left(\rho\left(\frac{\boldsymbol{u}^{n+1/2}-\boldsymbol{u}^n}{\Delta t}+\bar{\boldsymbol{u}}^{n+1/2}\cdot\nabla\boldsymbol{u}^{n+1/2}+\frac{1}{2}(\nabla\cdot\bar{\boldsymbol{u}}^{n+1/2})\boldsymbol{u}^{n+1/2}\right),\hat{\boldsymbol{u}}\right)_\Omega&\\=\left(p^{n+1/2}\boldsymbol{I}-2\eta_s\boldsymbol{D}(\boldsymbol{u}^{n+1/2})-\boldsymbol{\tau}_{\boldsymbol{p}}^{n+1/2},\nabla\hat{\boldsymbol{u}}\right)_\Omega,&\qquad(2.9)\\-\left(\nabla\cdot\boldsymbol{u}^{n+1},\hat{P}\right)_\Omega=0.&\qquad(2.10)\end{aligned}$$

for all admissible test functions $\hat{\boldsymbol{u}},\hat{P}$. Here, $\Delta t=t^{n+1}-t^n$ is the variable time step, $\boldsymbol{u}^{n+1/2}=(\boldsymbol{u}^{n+1}+\boldsymbol{u}^n)/2$ is a second-order approximation of $\boldsymbol{u}$ at the half-time level $t^{n+1/2}$, $\bar{\boldsymbol{u}}^{n+1/2}=(3\boldsymbol{u}^n-\boldsymbol{u}^{n-1})/2$ is an extrapolated approximation at the half-time level, and $\boldsymbol{D}(\boldsymbol{u})=\frac{1}{2}\left(\nabla\boldsymbol{u}+(\nabla\boldsymbol{u})^T\right)$ is the strain rate tensor.

2.3 Solving Oldroyd-B Using DG

The Oldroyd-B model (2.11), being convection-dominated, requires some modification in order to apply the finite element method. Assuming $\Omega \subset \mathbb{R}^2$ is a bounded region with Lipschitz continuous boundary Γ, the weak formulation is derived as follows. The Oldroyd-B equation can be written as

$$\lambda \left(\frac{\partial \boldsymbol{\tau_p}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{\tau_p} \right) - \boldsymbol{\tau_p} \cdot \nabla \boldsymbol{u} - (\nabla \boldsymbol{u})^T \cdot \boldsymbol{\tau_p} + \boldsymbol{\tau_p} = \eta_p \boldsymbol{D}. \tag{2.11}$$

We then take (2.11) and take the double-contraction between it and a test tensor $\hat{\boldsymbol{\tau}}$ and then integrate over each individual cell Ω_e in order to satisfy the weak problem locally.

Note that the convection term $\boldsymbol{u} \cdot \nabla \boldsymbol{\tau_p}$ must be handled with upwinding. We handle this term by recognizing that

$$\int_{\Omega_e} \left(\boldsymbol{u} \cdot \nabla \boldsymbol{\tau_p} \right) : \hat{\boldsymbol{\tau}} d\Omega = \int_{\Omega_e} \nabla \cdot \left(\boldsymbol{u} \boldsymbol{\tau_p} \right) : \hat{\boldsymbol{\tau}} d\Omega \tag{2.12}$$

due to the divergence free velocity field. By applying integration by parts and the divergence theorem, we get

$$\int_{\Omega_e} \nabla \cdot \left(\boldsymbol{u} \nabla \boldsymbol{\tau_p} \right) : \hat{\boldsymbol{\tau}} d\Omega = \int_{\partial \Omega_e} (\boldsymbol{n} \cdot \boldsymbol{u})(\boldsymbol{\tau_p} : \hat{\boldsymbol{\tau}})) ds - \int_{\Omega_e} \boldsymbol{u} \boldsymbol{\tau_p} \vdots \nabla \hat{\boldsymbol{\tau}} d\Omega. \tag{2.13}$$

The $(\boldsymbol{n} \cdot \boldsymbol{u}) \boldsymbol{\tau_p}$ term is then approximated using numerical flux. Performing integration by parts again on the last term of (2.13), we get

$$\int_{\Omega_e} \left(\boldsymbol{u}\cdot\nabla\boldsymbol{\tau_p}\right) : \hat{\boldsymbol{\tau}} d\Omega = \int_{\partial\Omega_e} [(\boldsymbol{n}\cdot\boldsymbol{u})\boldsymbol{\tau_p}]^* : \hat{\boldsymbol{\tau}} ds - \int_{\Omega_e} [(\boldsymbol{n}\cdot\boldsymbol{u})\boldsymbol{\tau_p}]^- : \hat{\boldsymbol{\tau}}\Omega + \int_{\Omega_e} \nabla\cdot(\boldsymbol{u}\boldsymbol{\tau_p}) : \hat{\boldsymbol{\tau}} d\Omega \tag{2.14}$$

The weak form is thus given by

$$\left(\lambda\overset{\nabla}{\boldsymbol{\tau_p}} + \boldsymbol{\tau_p} - 2\eta_p\boldsymbol{D}, \hat{\boldsymbol{\tau}}\right)_{\Omega_e} + \left([(\boldsymbol{n}\cdot\boldsymbol{u})\boldsymbol{\tau_p}]^* - [(\boldsymbol{n}\cdot\boldsymbol{u})\boldsymbol{\tau_p}]^-, \hat{\boldsymbol{\tau}}\right)_{\partial\Omega_e} = 0 \tag{2.15}$$

The building block of our time discretization is the Euler's method, where we seek a solution $\boldsymbol{\tau_p}^{n+1} \in H^1(\Omega)^1$ satisfying

$$\left(\lambda\left(\frac{\boldsymbol{\tau_p}^{n+1}-\boldsymbol{\tau_p}^n}{\Delta t} + (\boldsymbol{u}^n\cdot\nabla\boldsymbol{\tau_p}^n) - \boldsymbol{\tau_p}^n\cdot\nabla\boldsymbol{u}^n - (\nabla\boldsymbol{u}^n)^T\cdot\boldsymbol{\tau_p}^n\right) + \boldsymbol{\tau_p}^n - 2\eta_p\boldsymbol{D}^n, \hat{\boldsymbol{\tau}}\right)_{\Omega_e} + \left([(\boldsymbol{n}\cdot\boldsymbol{u}^n)\boldsymbol{\tau_p}^n]^* - [(\boldsymbol{n}\cdot\boldsymbol{u}^n)\boldsymbol{\tau_p}^n]^-, \hat{\boldsymbol{\tau}}\right)_{\partial\Omega_e} = 0. \tag{2.16}$$

There are several options for numerical flux. Two commonly used choices for flux are the Roe flux and the Lax-Friedrichs flux. Both fluxes use information from neighboring cells as shown in figure 2.1.

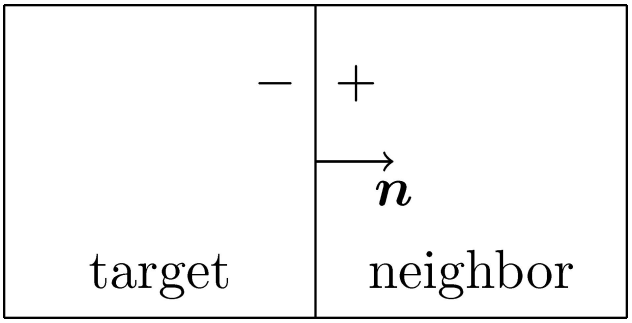


Figure 2.1: Target cell and neighboring cell

Roe flux is given by

$$[(\boldsymbol{n}\cdot\boldsymbol{u})\boldsymbol{\tau_p}]^* = \begin{cases} (\boldsymbol{n}\cdot\boldsymbol{u})\boldsymbol{\tau_p}^- & \boldsymbol{n}\cdot\boldsymbol{u} \geq 0 \\ (\boldsymbol{n}\cdot\boldsymbol{u})\boldsymbol{\tau_p}^+ & \text{otherwise} \end{cases} \tag{2.17}$$

which must be calculated on every quadrature point on the cell boundary. Roe flux is prone to oscillations in the solution when normal velocity is close to zero and switches signs due to numerical errors. This downside poses enough of a problem for us to favor the Lax-Friedrichs flux instead.

The Lax-Friedrichs flux is given by

$$[(\boldsymbol{n}\cdot\boldsymbol{u})\boldsymbol{\tau_p}]^* = (\boldsymbol{n}\cdot\boldsymbol{u})\frac{\boldsymbol{\tau_p}^+ + \boldsymbol{\tau_p}^-}{2} - \frac{\alpha}{2}(\boldsymbol{\tau_p}^+ - \boldsymbol{\tau_p}^-) \tag{2.18}$$

where $\alpha = \max|\boldsymbol{n}\cdot\boldsymbol{u}|$ is the maximum normal velocity calculated over each quadrature point on the relevant cell edge and is the penalty term associated with the Lax-Friedrichs flux. If the cell edge is on a boundary with no inflow or outflow, then the numerical flux is set to zero. If the cell edge is on a boundary that does have inflow or outflow, we simply use the Roe flux. That is, the numerical flux is determined by the prescribed boundary condition on the inflow boundary and by the target cell on the outflow boundary. The global Lax-Friedrichs flux, where α is evaluated throughout the whole domain, yields a monotone scheme, which is very stable and converges to the physical entropy solution [8], but this global Lax-Friedrichs flux may cause too much dissipation to the solution. Therefore we chose the local Lax-Friedrichs where α is evaluated over the relevant edge. In order to get a convergence order higher than 1 in time, Runge-Kutta methods are required [8], which we will outline our use of in Section 2.4.

2.4 Runge-Kutta DG

As mentioned in the previous section, use of the DG requires a Runge-Kutta method in order to obtain convergence order higher than 1 in time [8]. Runge-Kutta methods describe a general class of iterative methods including both implicit and explicit schemes used in approximating time-derivatives.

The time-stepping in our work is handled by using a third-order total variation diminishing Runge-Kutta scheme described in [15], which is essentially a weighted average of multiple forward-Euler steps.

In order to implement a third-order total variation diminishing Runge-Kutta scheme, first define the forward Euler discretization of (2.16) as

$$\frac{\boldsymbol{\tau}_{\boldsymbol{p}}^{n+1} - \boldsymbol{\tau}_{\boldsymbol{p}}^{n}}{\Delta t} = \boldsymbol{L}(\boldsymbol{\tau}_{\boldsymbol{p}}^{n}, \boldsymbol{u}^{n}). \tag{2.19}$$

Then, we perform the following computations to obtain $\boldsymbol{\tau}_{\boldsymbol{p}}^{n+1}$:

$$\begin{aligned}
\boldsymbol{\tau}_{\boldsymbol{p}}^{(1)} &= \boldsymbol{\tau}_{\boldsymbol{p}}^{n} + \Delta t \boldsymbol{L}(\boldsymbol{\tau}_{\boldsymbol{p}}^{n}, \boldsymbol{u}^{n}) \\
\boldsymbol{\tau}_{\boldsymbol{p}}^{(2)} &= \frac{3}{4}\boldsymbol{\tau}_{\boldsymbol{p}}^{n} + \frac{1}{4}\boldsymbol{\tau}_{\boldsymbol{p}}^{(1)} + \frac{1}{4}\Delta t \boldsymbol{L}(\boldsymbol{\tau}_{\boldsymbol{p}}^{(1)}, \boldsymbol{u}^{n}) \\
\boldsymbol{\tau}_{\boldsymbol{p}}^{n+1} &= \frac{1}{3}\boldsymbol{\tau}_{\boldsymbol{p}}^{n} + \frac{2}{3}\boldsymbol{\tau}_{\boldsymbol{p}}^{(2)} + \frac{2}{3}\Delta t \boldsymbol{L}(\boldsymbol{\tau}_{\boldsymbol{p}}^{(2)}, \boldsymbol{u}^{n}).
\end{aligned} \tag{2.20}$$

The work of Gottlieb and Shu [15] showed that (2.20) is the optimal third-order total variation diminishing scheme. Since this third order Runge-Kutta method is only implemented for the polymer stress solutions and the scheme for the Navier-Stokes equations is 2nd order accurate in time, the overall scheme is limited to 2nd order accuracy in time. It should then

be noted that we choose 3rd order Runge-Kutta method because of its better stability than lower-order Runge-Kutta methods.

The benefit to writing the scheme in the form used in (2.20) is that the three Runge-Kutta stages can be completed using a single for-loop and only needs one additional vector to store the intermediate solutions from different Runge-Kutta stages.

2.5 Numerical Discretization and Time-Stepping

We aim to solve (2.9-2.10), decoupled at each step from (2.15) using the mixed finite element method. Equation (2.15) is solved separately using the discontinuous Galerkin method. The code is developed solved using the deal.II library [1, 4].

The system described by (2.9-2.10) is solved using mixed finite element method, and the resulting linear system is the saddle point problem:

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} U \\ P \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix} \tag{2.21}$$

where U, P are the solution vectors for $\boldsymbol{u}^{n+1}$ and $P^{n+1/2}$, respectively, A is an asymmetric block matrix that arises from (2.9), and B is the block matrix that arises from (2.10). The linear system (2.21) is then solved using the direct sparse linear solver UMFPACK [10], since the number of degrees of freedom is small enough in 2D.

Since we can compute the solution to equation (2.15) on each cell individually, this problem is highly parallelizable. The resulting systems are then small enough that we can compute the solutions on each cell directly using the deal.II function for matrix inverses [4], which is done in parallel. The local solutions are then mapped to the global solution. However, in

our current method, most of the computational cost is spent on solving the Navier-Stokes equations. This computational cost can be reduced, but in this work, we focus on making the solver for the Oldroyd-B model efficient.

The procedure for time stepping is described as follows:

(i) Set up the initial mesh.

(ii) Set up the initial data for $\boldsymbol{\tau}_{\boldsymbol{p}}^0$ and $\boldsymbol{u}^0$.

(iii) For each time step $n \geq 0$, choose a suitable Δt and perform the following steps until the stopping stopping criterion is met.

  (a) Use $\boldsymbol{u}^n$ and $\boldsymbol{\tau}_{\boldsymbol{p}}^n$ to compute $\boldsymbol{\tau}_{\boldsymbol{p}}^{n+1}$ using the TVD-RK method.

  (b) Use $\boldsymbol{u}^n$ and $\boldsymbol{\tau}_{\boldsymbol{p}}^{n+1}$ to compute $\boldsymbol{u}^{n+1}$ and $P^{n+1/2}$.

Since we aim to show steady state solutions, we loop through many time steps until the solution no longer varies in time.

Chapter 3

Plane Poiseuille Flow

3.1 Introduction

An important benchmark for testing the validity of viscoelastic fluid flow-solvers is the plane Poiseuille flow problem. Plane Poiseuille flow arises when a fluid is flowing through a channel of uniform width driven by a pressure with constant gradient and no-slip conditions on the channel boundary. Due to plane Poiseuille flow having a closed-form, steady state solution, numerical tests can be directly compared with the exact solution to verify accuracy and confirm mesh convergence.

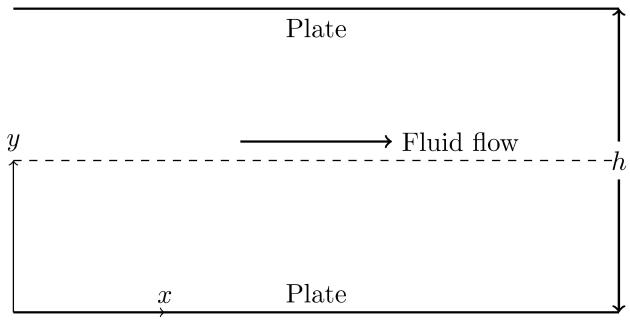


Figure 3.1: Diagram of plane Poiseuille flow

3.2 Initial Conditions and Boundary Conditions

The assumptions for the plane Poiseuille flow is that the fluid is flowing in an infinitely long channel with a total width of h. As such, the fluid is assumed to be in a steady state so $\frac{\partial \boldsymbol{u}}{\partial t} = 0$. Further, it is assumed that $v = 0$ since the flow will only occur in the x direction. Thus, by the incompressibility condition $\nabla \cdot \boldsymbol{u} = 0$, we have

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \frac{\partial u}{\partial x} = 0.$$

This is enough to reduce the complexity of the Navier-Stokes equations considerably, such that the solution is readily found.

From the polymer stress in steady shear flow (3.5-3.7), and further considering $\frac{\partial \tau_{p,xx}}{\partial x} = 0$ in the fully developed Poiseuille flow, we can see that only the $\tau_{p,xy}$ component affects flow. Thus the behavior of the Oldroyd-B fluid in the Poiseuille flow is identical to that of a Newtonian fluid with viscosity $\eta = \eta_s + \eta_p$.

In vector form, we have that (2.1) becomes

$$\rho \begin{pmatrix} \frac{\partial u}{\partial t} \\ \frac{\partial v}{\partial t} \end{pmatrix} + \rho \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = - \begin{pmatrix} \frac{\partial P}{\partial x} \\ \frac{\partial P}{\partial y} \end{pmatrix} + \eta \begin{pmatrix} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \\ \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \end{pmatrix}. \tag{3.1}$$

Solving for the first component of system (3.1) will yield

$$\rho \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right) = - \frac{\partial P}{\partial x} + \eta \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \tag{3.2}$$

Canceling terms using the problem assumptions, (3.2) becomes $-\frac{\partial P}{\partial x} + \eta \frac{\partial^2 u}{\partial y^2} = 0$.

Imparting the assumption that the gradient of pressure is constant, we let $-\frac{\partial P}{\partial x} = 2C$ for some constant C. Then, we have the solution

$$u = \frac{C}{\eta} y(h-y); v = 0. \tag{3.3}$$

Now that we have the velocity initial condition solved for, we can solve for the polymer stress. Similar to the velocity profile, the polymer stress is in a steady state. Therefore $\frac{\partial \boldsymbol{\tau_p}}{\partial t} = 0$. Also, note that

$$\nabla \boldsymbol{u} = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} \\ \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ \frac{\partial u}{\partial y} & 0 \end{pmatrix},$$

and $\boldsymbol{\tau_p}$ is a function of y only so $\frac{\partial \boldsymbol{\tau_p}}{\partial x} = 0$.

Taking (2.11) and plugging in the velocity found above, we get

$$\begin{aligned} -\begin{pmatrix} \tau_{p,xx} & \tau_{p,xy} \\ \tau_{p,xy} & \tau_{p,yy} \end{pmatrix} \begin{pmatrix} 0 & 0 \\ \frac{\partial u}{\partial y} & 0 \end{pmatrix} - \begin{pmatrix} 0 & \frac{\partial u}{\partial y} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \tau_{p,xx} & \tau_{p,xy} \\ \tau_{p,xy} & \tau_{p,yy} \end{pmatrix} \\ + \frac{1}{\lambda} \begin{pmatrix} \tau_{p,xx} & \tau_{p,xy} \\ \tau_{p,xy} & \tau_{p,yy} \end{pmatrix} = \eta_p \begin{pmatrix} 0 & \frac{\partial u}{\partial y} \\ \frac{\partial u}{\partial y} & 0 \end{pmatrix}. \end{aligned} \tag{3.4}$$

Focusing on the bottom-right component, we immediately get that $\tau_{p,yy} = 0$. Then focusing on the bottom-left, we get $\tau_{p,xy} = \eta_p \frac{\partial u}{\partial y}$. Lastly, the first component will yield $\tau_{p,xx} = 2\lambda\eta_p \left(\frac{\partial u}{\partial y}\right)^2$.

Therefore, the components of the polymer stress are

$$\tau_{p,xx} = 2\lambda\eta_p \left(\frac{\partial u}{\partial y}\right)^2 \tag{3.5}$$

$$\tau_{p,xy} = \eta_p \frac{\partial u}{\partial y} \tag{3.6}$$

$$\tau_{p,yy} = 0. \tag{3.7}$$

The characteristic length of this problem corresponds to the height between the two plates, and the characteristic flow rate is the average value of the inflow velocity. Therefore, in order to get $Wi = \lambda$, we assume $h = 1$, so $u = \frac{C}{\eta_p} y(1-y)$. We then require

$$\int_0^1 u(y)dy = 1,$$

and we get

$$u = 6y(1-y). \tag{3.8}$$

Thus $\frac{\partial P}{\partial x} = -\frac{12}{\eta}$, $P = -\frac{12}{\eta}x + D$ for some integration constant D. Plugging the velocity profile into (3.5-3.7), we get the boundary condition for polymer stress at the inflow boundary:

$$\tau_{p,xx} = 36\lambda\eta\,(1-2y)^2 \tag{3.9}$$

$$\tau_{p,xy} = 6\eta\,(1-2y) \tag{3.10}$$

$$\tau_{p,yy} = 0. \tag{3.11}$$

For incompressible fluid flows, pressure interacts with the flow field by its gradient. There-

fore, without extra conditions placed on pressure, the solution is not unique or unique only up to an additive constant. Often assigning a value for pressure at a single point in the computational domain is enough, but to further add stability to the code, the mean value constraint for pressure was implemented on the inflow boundary:

$$\int_{\{(x,y):x=0,0\leq y\leq 1\}} P ds = 0.$$

The second component of (3.1) yields $\frac{\partial P}{\partial y} = 0$. Thus the exact solution of pressure is $P = -\frac{12}{\eta}x$.

Besides the inflow boundary, we assume that velocity and polymer stress is initially zero everywhere in the computational domain. Lastly, we impose the no-slip conditions on the upper and lower walls, and impose the known solutions as conditions for polymer stress and velocity on the inflow and outflow boundaries.

3.3 Results

The results in this section display that the numerical solutions were accurate to the exact solutions for polymer stress and velocity. We found that the solution became unstable around $Wi \approx 2.0$ due to initial transient, so only results below that value are shown. All results in this section are steady state, which was determined to be when the numerical solution did not change from time-step to time-step. Additionally, we set $\eta_s = \eta_p = 0.5, \rho = 0$, and we used biquadratic Lagrange polynomials for the basis functions for velocity, bilinear Lagrange polynomials for pressure, and bilinear Lagrange polynomials for polymer stress. These polynomial basis functions are tensor product polynomials of the specified degree with Gauss-Lobatto node points in each coordinate direction. Although pressure and polymer

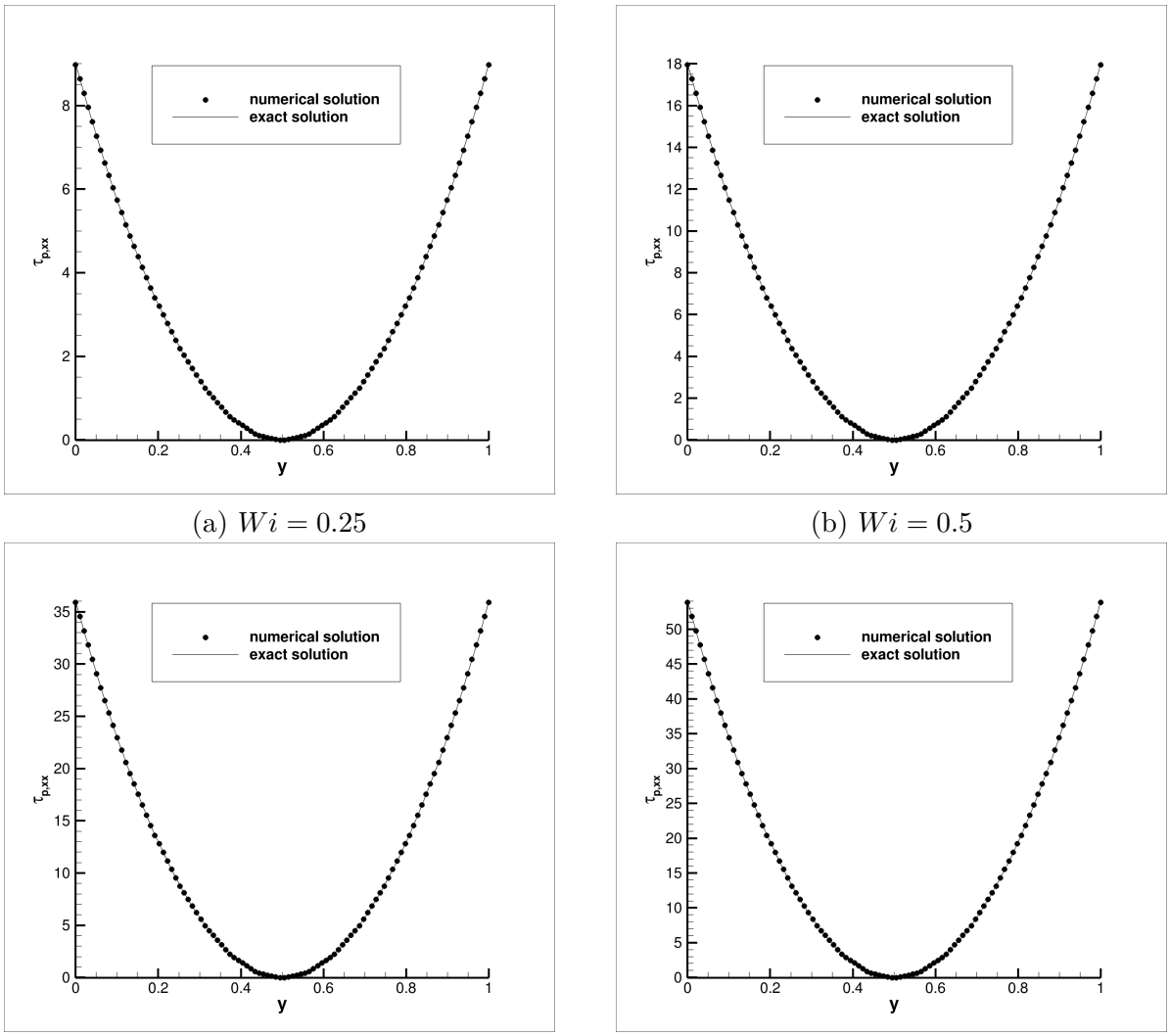
h	u error	u order	P error	P order	$\boldsymbol{\tau_p}$ error	$\boldsymbol{\tau_p}$ order
1/4	4.57e-4	—	6.56e-01	—	1.26e0	—
1/8	2.92e-06	7.29	4.65e-02	3.82	2.02e-1	2.64
1/16	3.86e-08	6.24	4.90e-03	3.24	4.59e-2	2.14
1/32	3.90e-11	9.95	3.07e-05	7.32	3.19e-3	3.85
1/64	1.02e-11	1.93	5.99e-07	5.68	1.03e-3	1.63

Table 3.1: L^2 Errors between numerical solution and exact solution for u, P, and $\boldsymbol{\tau_p}$ at various mesh sizes and the corresponding convergence orders. We use the Frobeinius norm for the pointwise error of the polymer stress.

stress have the same polynomial degree, the former is in a continuous finite-element space while the latter is discontinuous. The mesh used was a uniform mesh with computation domain $[0, 2] \times [0, 1]$.

First, we show that the polymer stress and velocity profiles match the exact solutions. We let $h = 1/32$ and compute the solutions for $Wi = \lambda = 0.25, 0.5, 1.0,$ and 1.5, which can be seen in Figures 3.2, 3.3, and 3.4. These results show a nice match between the computed and exact solutions.

Additionally, we tested the mesh convergence of our code by fixing $Wi = 1$ and varying the mesh size. Then, we calculated the L^2 errors between the calculated solutions and the exact solutions for the u-component of velocity and the pressure. For the polymer stress, we calculated the Frobenius norm between the calculated solutions and the exact solutions. The calculated errors are shown in Table 3.1. Our results show that the mesh convergence order for polymer stress is approximately 2.65 and the mesh convergence order for velocity is approximately 6.7. The large convergence order for velocity is likely due to the fact that we used degree 2 polynomials for our velocity solution space, and the exact solution for u is a degree 2 polynomial. The exact solution for the pressure is $P = -12x$ since $\eta = 1.0$ for these results, and the convergence order for pressure was determined to be approximately 5.1.



(a) $Wi = 0.25$

(b) $Wi = 0.5$

(c) $Wi = 1.0$

(d) $Wi = 1.5$

Figure 3.2: Comparisons between exact and numerical solutions for $\boldsymbol{\tau_p}_{,xx}$ at $x = 1$ and various Wi

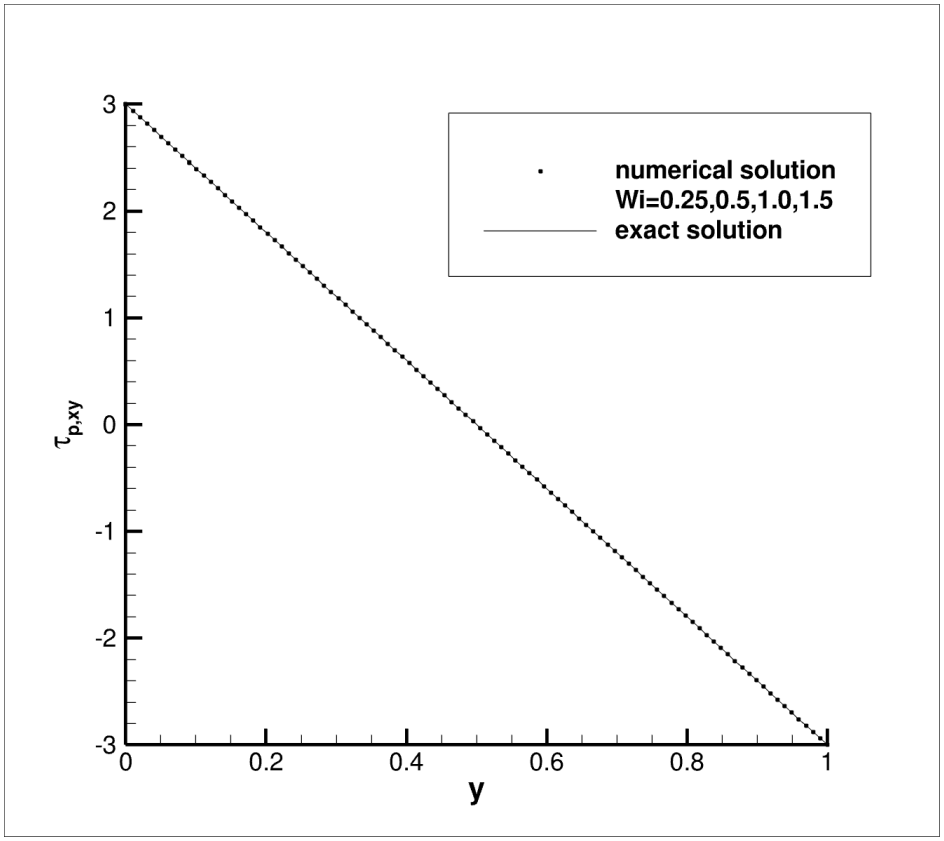


Figure 3.3: Comparison between exact and numerical solutions for $\boldsymbol{\tau}_{p,xy}$ with $Wi = 0.25, 0.5, 1.0, 1.5$ at $x = 1$ compiled onto a single graph

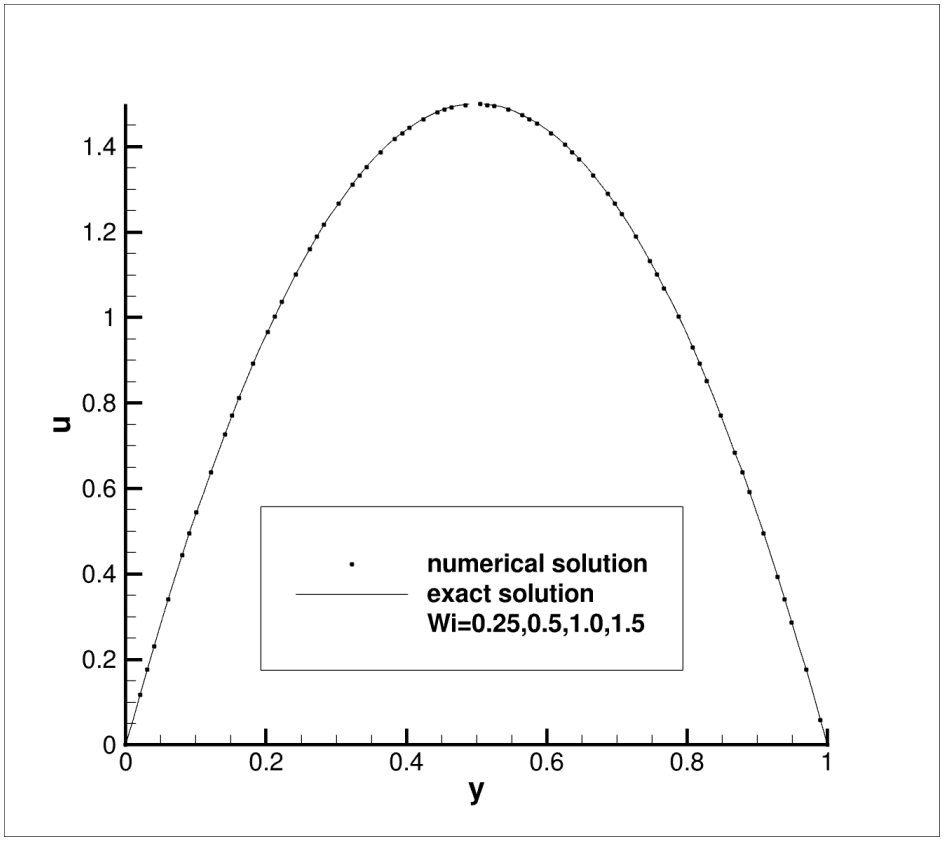
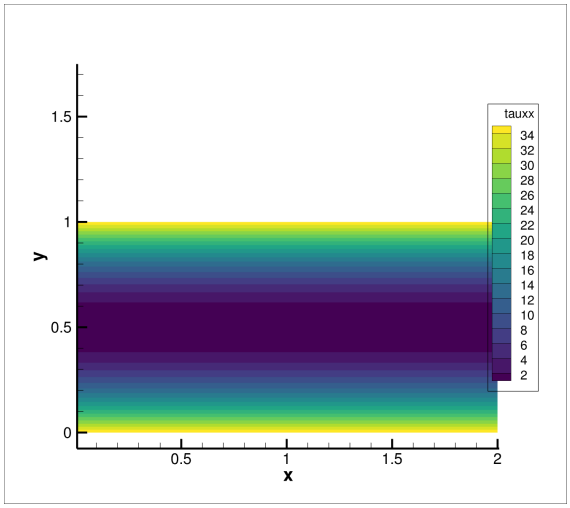
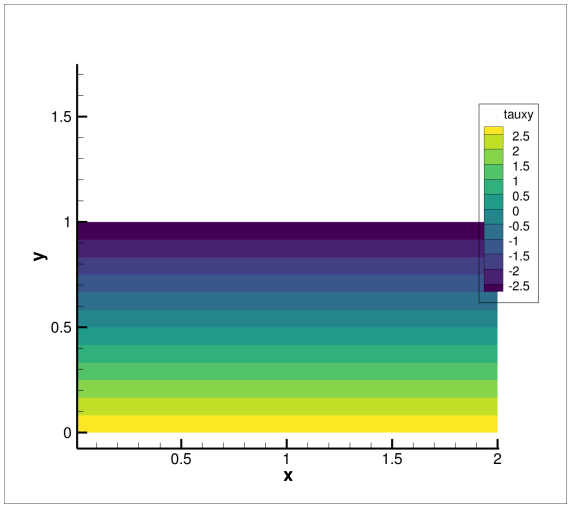


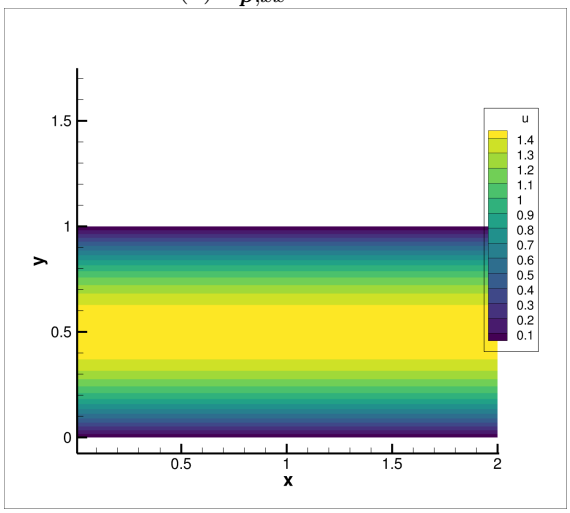
Figure 3.4: Comparison between exact and numerical solutions for $u(y)$ with $Wi = 0.25, 0.5, 1.0, 1.5$ at $x = 1$ compiled onto a single graph



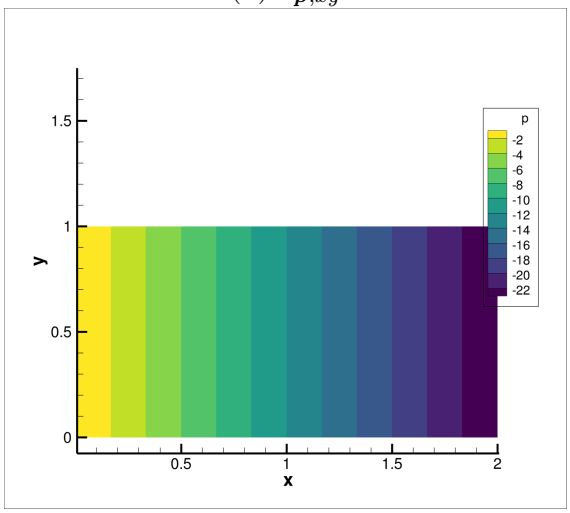
(a) $\boldsymbol{\tau_p}_{,xx}$ contour



(b) $\boldsymbol{\tau_p}_{,xy}$



(c) u contour



(d) P contour

Figure 3.5: Contours of various solution for $Wi = 1.0$

Chapter 4

Lid-driven Cavity Flow Problem

4.1 Introduction

Despite not having immediate application, the cavity flow problem is an important benchmark for flow-solvers trying to solve problems involving viscous incompressible fluid flows. A particular feature that arises in this flow problem is the large vortex that quickly develops near the center of the computational domain. The location of the center of this vortex is of particular interest in benchmarking the numerical solutions. In this chapter, we introduce initial and boundary conditions as well as discuss results in detail, including modifications made to the code to aid in solutions to this particular problem.

4.2 Initial Conditions and Boundary Conditions

In this problem, the characteristic strain rate is $\dot{\gamma} = 1$. This is a result of the characteristic length of the square cavity being 1, with the characteristic flow velocity having a magnitude of 1. Therefore, the Weissenberg number for this problem is given by $Wi = \lambda$. The conventional setup for the upper boundary for the cavity flow problem is to have the horizontal component of velocity be 1 on the upper boundary with the other components set to 0. This is to mimic a flow across the entrance to the channel causing vortices to develop. However, the standard boundary condition causes a discontinuity on the boundary, which poses numerical issues at

the two upper corners, especially when viscoelasticity is considered. Thus, in our work, the upper boundary set to the time-dependent function

$$u(x,t) = 8\big(1+\tanh(4t-8)\big)x^2(1-x)^2 \tag{4.1}$$

to aid in stabilizing the solution as was applied in [13]. The new time-dependent boundary condition creates a smooth transition from the zero boundary condition on the left and right boundaries to the upper boundary. As time increases, (4.1) approaches the steady state of $x^2(1-x)^2$.

The velocity boundary conditions are shown in Figure 4.1.

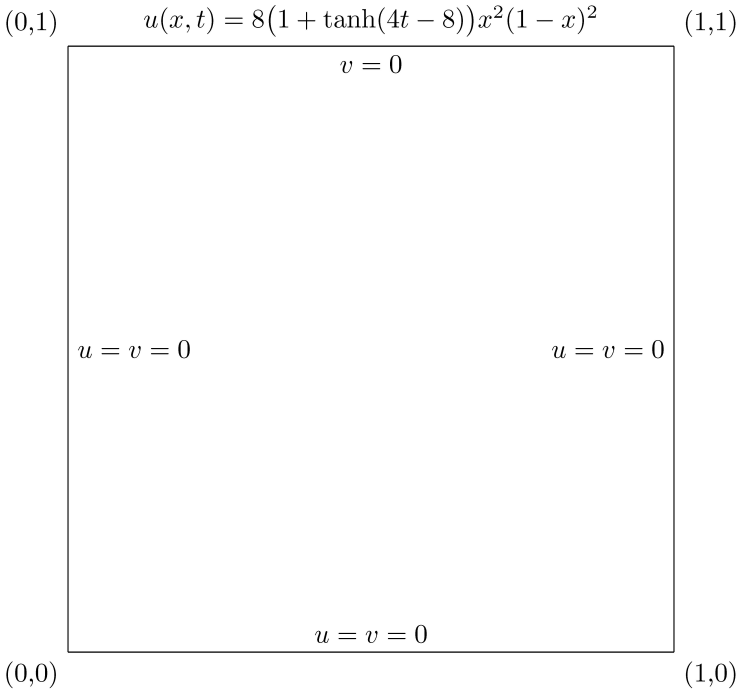


Figure 4.1: The computational domain of the cavity flow problem with boundary conditions for velocity

To impose a condition on pressure, we applied the mean value constraint for pressure on the

upper boundary:

$$\int_{\{(x,y):y=1,0\leq x\leq 1\}} Pds = 0.$$

The initial condition for polymer stress was zero everywhere. Since there is no flow across the boundaries, we only need to impose zero flux in DG and no other boundary conditions are required for the polymer stress.

The mesh used was the 2D Cartesian mesh with no localized mesh refinement.

4.3 DEVSS-Type Stabilization

For the case where $Wi = 0$, we found that there was an instability in the code since the polymer stress is zero everywhere at $t = 0$ but reaches a finite value at $t > 0$. This sharp increase in polymer stress causes numerical oscillations due to the explicit treatment of polymer stress in the momentum equation.

We thus implemented a DEVSS-type of stabilization by modifying the right-hand side of (2.9) in the following way:

$$\begin{aligned}\rho\left(\frac{\boldsymbol{u}^{n+1/2}-\boldsymbol{u}^{n}}{\Delta t}+\bar{\boldsymbol{u}}^{n+1/2}\cdot\nabla\boldsymbol{u}^{n+1/2}+\frac{1}{2}(\nabla\cdot\bar{\boldsymbol{u}}^{n+1/2})\boldsymbol{u}^{n+1/2}\right)\\ =\nabla\cdot\left[2(\eta_s+\delta)\boldsymbol{D}(\boldsymbol{u}^{n+1/2})-2\delta\boldsymbol{D}(\bar{\boldsymbol{u}}^{n+1/2})+\boldsymbol{\tau_p}^{n+1/2}\right]\end{aligned} \tag{4.2}$$

where δ is a stabilization parameter. A positive δ adds more ellipticity to the momentum equation, which is similar to the DEVSS method [2]. Note that this equation reduces to the standard weak formulation if $\boldsymbol{u}^{n+1/2} = \bar{\boldsymbol{u}}^{n+1/2}$. We choose $\delta = \eta_p$. In the case where $Wi = 0$,

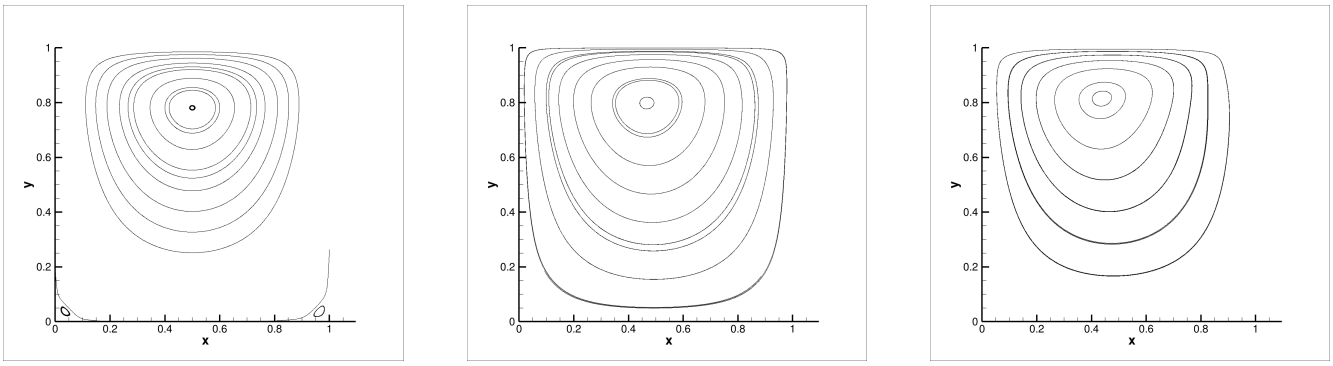
$\boldsymbol{\tau_p}^{n+1/2} \approx 2\delta \boldsymbol{D}(\bar{\boldsymbol{u}}^{n+1/2})$ and (4.2) reduces to the momentum equation for a Newtonian fluid with viscosity $\eta_s + \eta_p$.

4.4 Results

The results in this section display the difficulty in solving viscoelastic fluid flows. For one, the problem exhibits the typical numerical instability present with large values of Weissenberg number, as discussed in section 1.3. We found that the solution became unstable around $Wi \approx 1.5$, so only results below that value are shown. We run the simulations to $t_{\text{total}} \approx 30$, which is sufficient for the solution to reach the steady state. All results in this section are steady state. The mesh size used was 128×128. Additionally, we set $\eta_s = \eta_p = 0.5, \rho = 0$, and similar to Chapter 3, we used biquadratic Lagrange polynomials of for the basis functions for velocity, bilinear Lagrange polynomials for pressure, and bilinear Lagrange polynomials for polymer stress. These polynomial basis functions are tensor product polynomials of the specified degree with Gauss-Lobatto node points in each coordinate direction. Although pressure and polymer stress have the same polynomial degree, the former is in a continuous finite-element space while the latter is discontinuous.

The results found are consistent with those found in literature, as is evident in the locations of vortex centers in Table 4.1. The expectation that the vortex center moves leftward as Weissenberg number increases was observed. Further, the velocity profiles shown in Figure 4.3 are also consistent with those found in [13].

However, the polymer stress needed additional work to properly match the results found in literature. Of note in Figure 4.4, the solutions for polymer stress on the upper boundary exhibit oscillations. This is likely due to the sharp increase in polymer stress near the upper boundary. The coarse mesh could not adequately resolve this sharp increase and thus leads



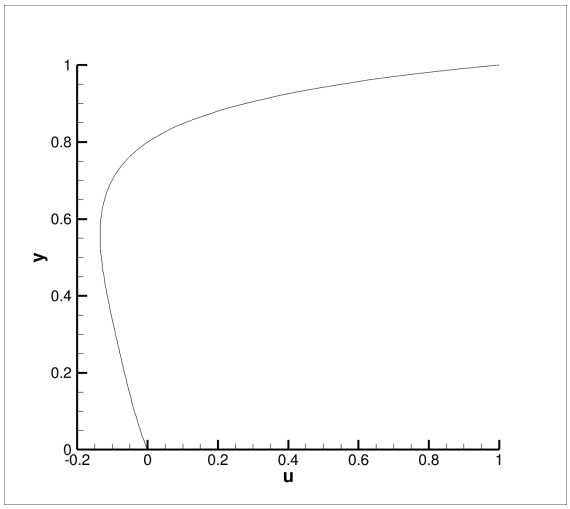
(a) $Wi = 0.0$ (b) $Wi = 0.5$ (c) $Wi = 1.0$

Figure 4.2: Streamlines for various Weissenberg numbers

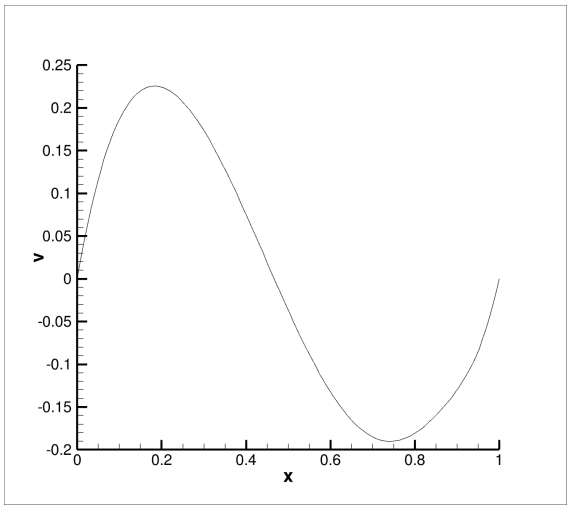
to oscillations.

When the mesh was refined to 512×512, the oscillations greatly diminished, which can be seen in the polymer stress profiles in Figure 4.5 and polymer stress contours in Figure 4.6.

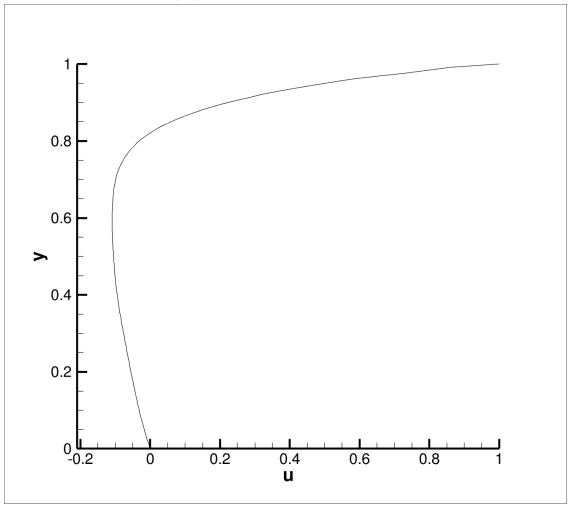
It should be noted that for cavity flow, there is no consensus on the maximum polymer stress on the upper boundary. For example, [23] had 5.4×10^3 for max $\tau_{p,xx}$ using the log-conformation formulation, [27] had 1.4×10^3 also using the log-conformation formulation, and [13] had around 900 using the regular formulation. Our results show a maximum of about 600, which keep increasing with mesh refinement.



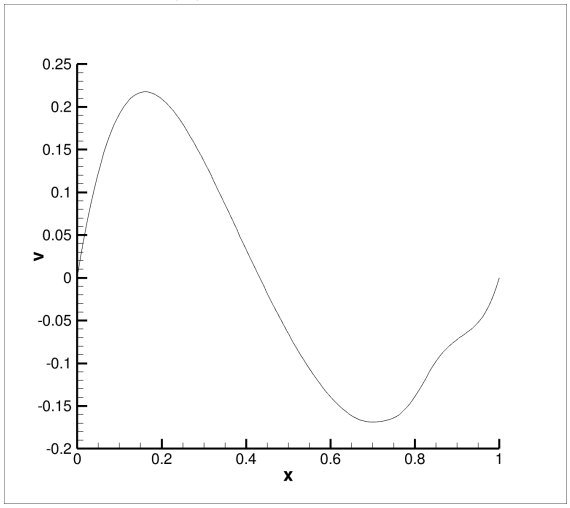
(a) u for $Wi = 0.5$



(b) v for $Wi = 0.5$

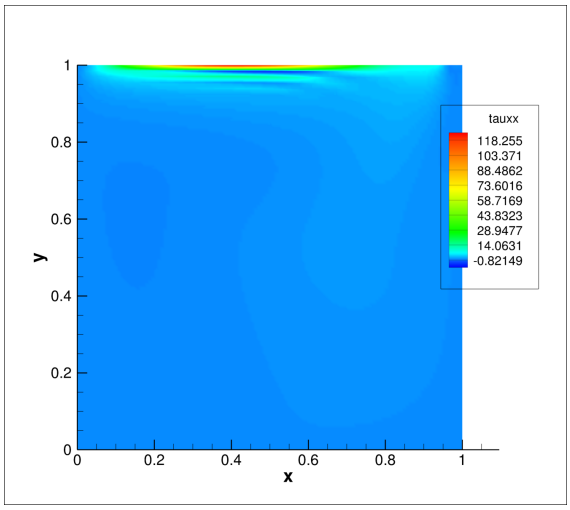


(c) u for $Wi = 1.0$

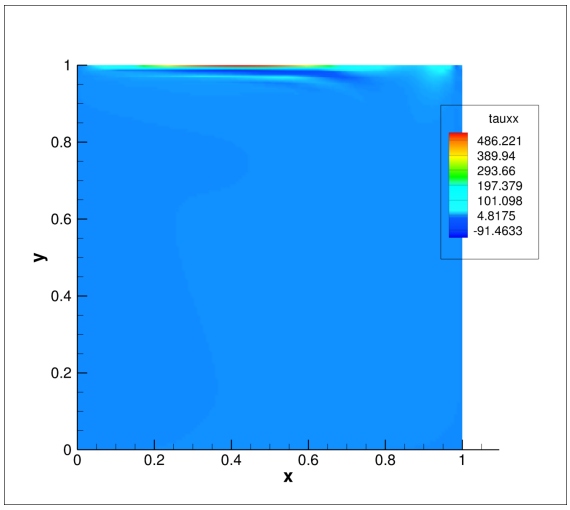


(d) v for $Wi = 1.0$

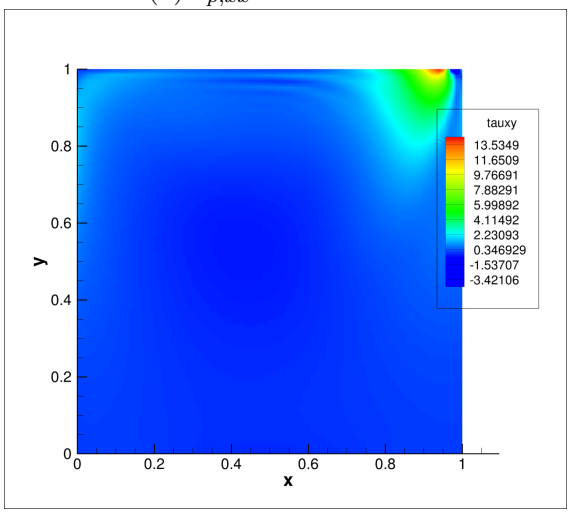
Figure 4.3: Left column: u-component velocity profile curves across $x = 0.5$. Right column: v-component velocity profile across $y = 0.5$



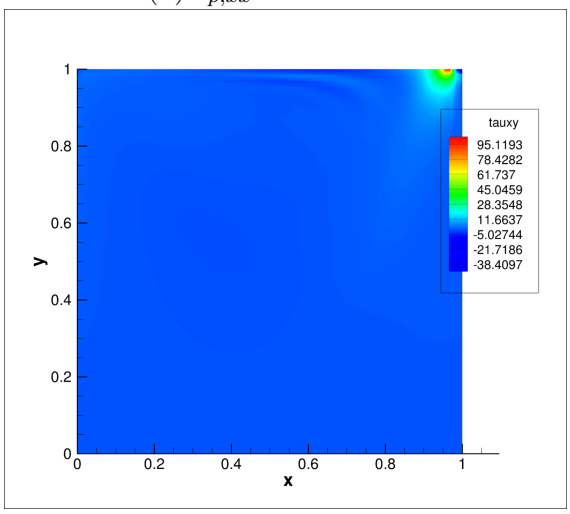
(a) $\tau_{p,xx}$ for $Wi = 0.5$



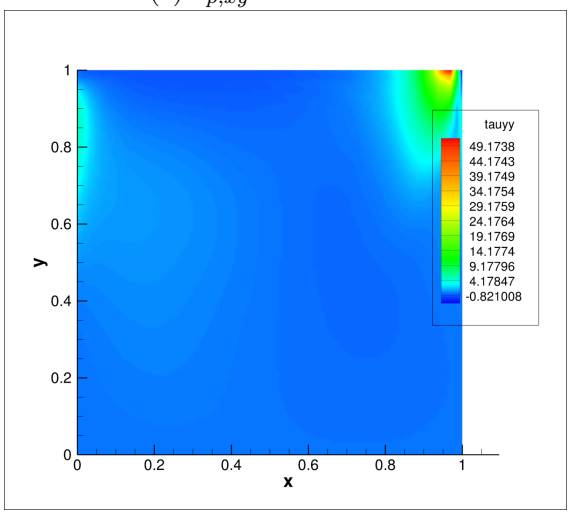
(b) $\tau_{p,xx}$ for $Wi = 1.0$



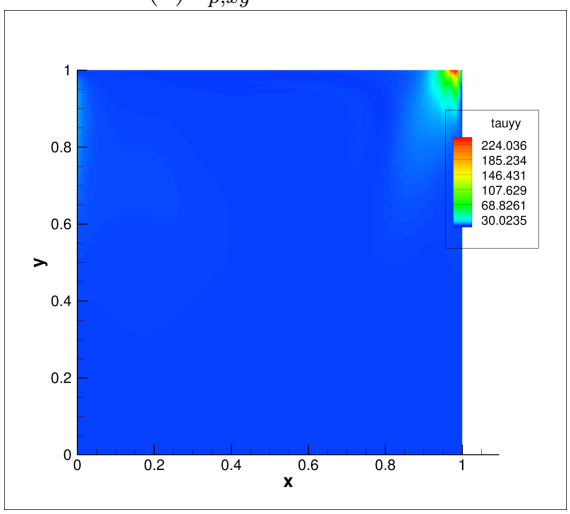
(c) $\tau_{p,xy}$ for $Wi = 0.5$



(d) $\tau_{p,xy}$ for $Wi = 1.0$

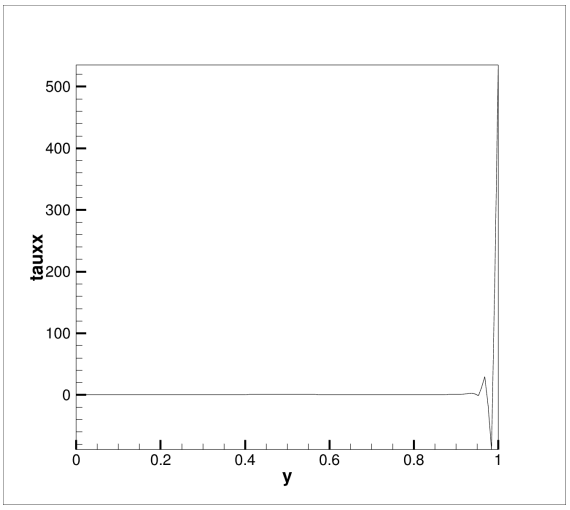


(e) $\tau_{p,yy}$ for $Wi = 0.5$

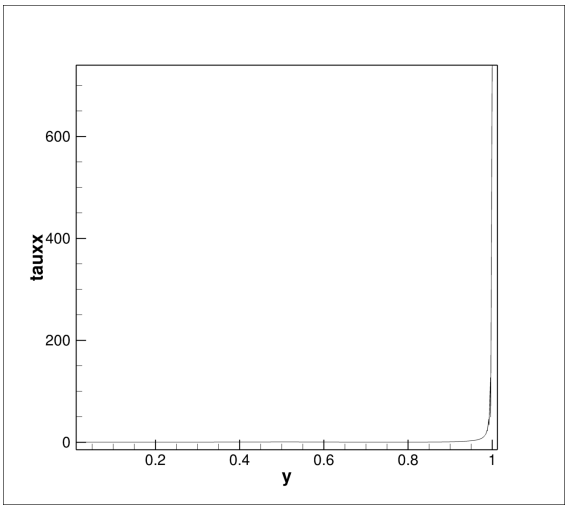


(f) $\tau_{p,yy}$ for $Wi = 1.0$

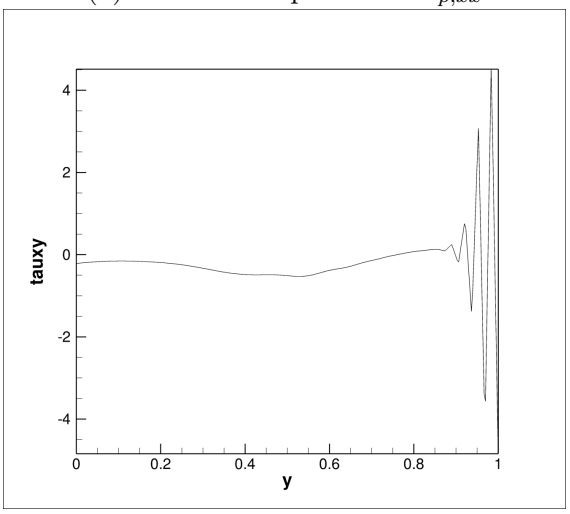
Figure 4.4: Contours of each component of polymer stress for $Wi = 0.5$ and $Wi = 1.0$



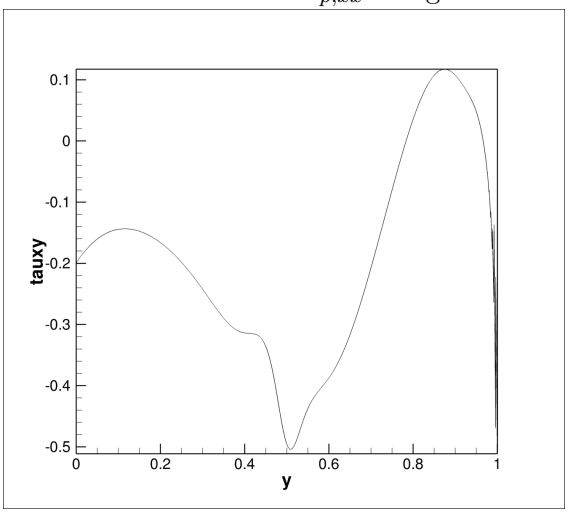
(a) Oscillations present in $\tau_{p,xx}$



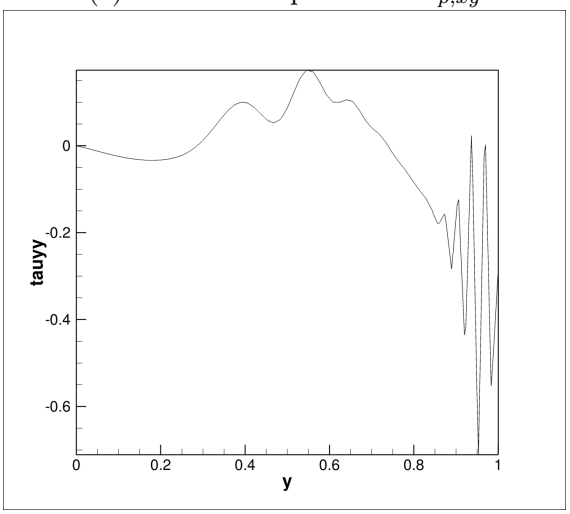
(b) Reduced oscillations in $\tau_{p,xx}$ using refined mesh



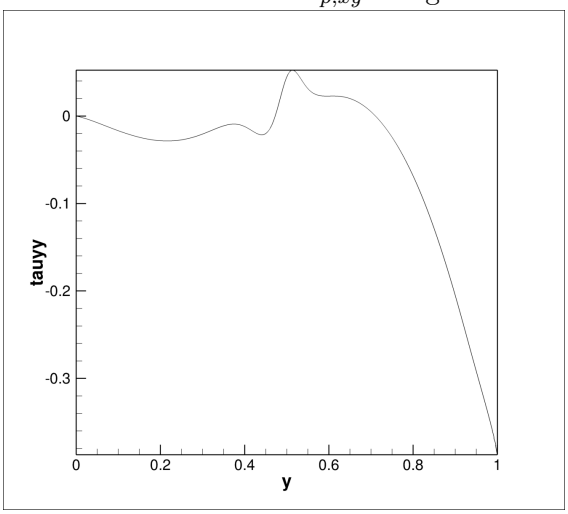
(c) Oscillations present in $\tau_{p,xy}$



(d) Reduced oscillations in $\tau_{p,xy}$ using refined mesh

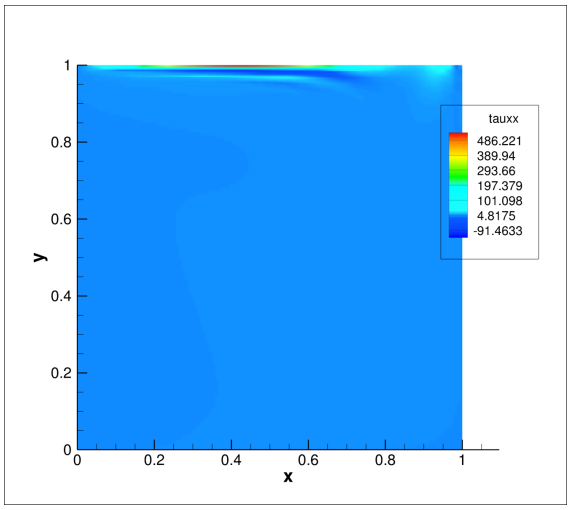


(e) Oscillations present in $\tau_{p,yy}$

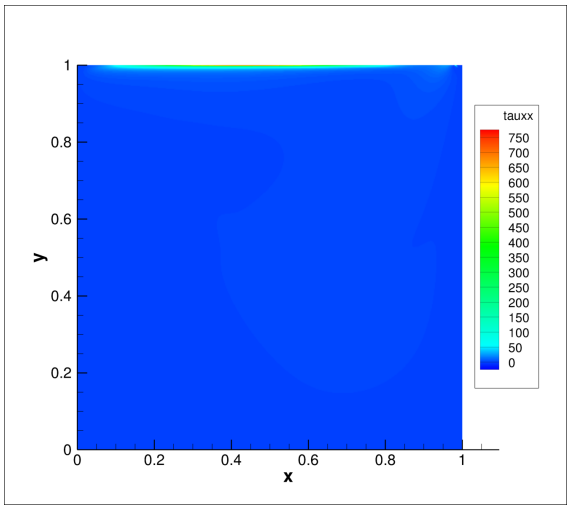


(f) Reduced oscillations in $\tau_{p,yy}$ using refined mesh

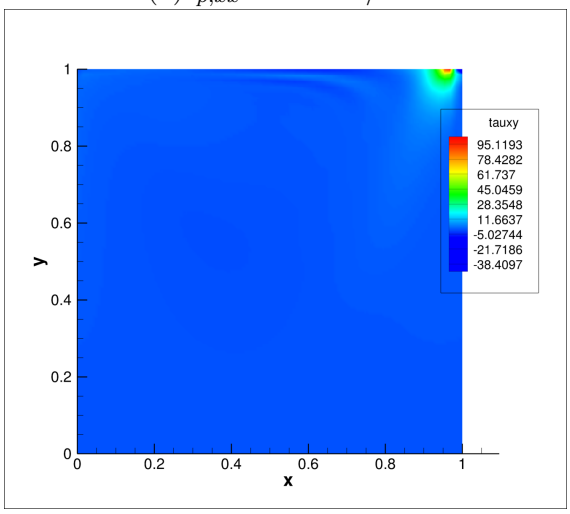
Figure 4.5: Oscillations present in $\boldsymbol{\tau_p}$ profiles across $x = 0.5$ and with $Wi = 1.0$



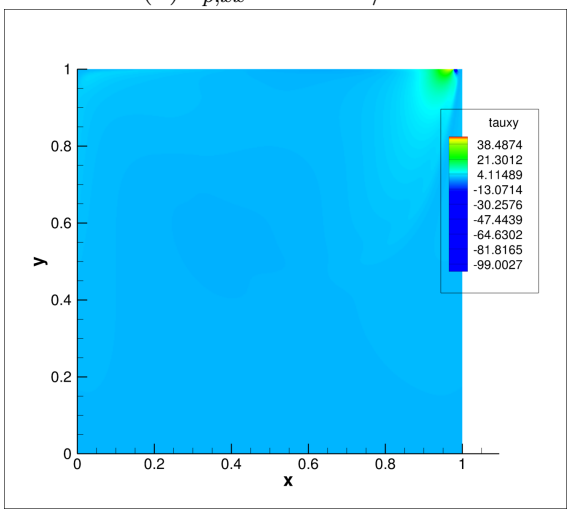
(a)$\tau_{p,xx}$ for $h = 1/128$



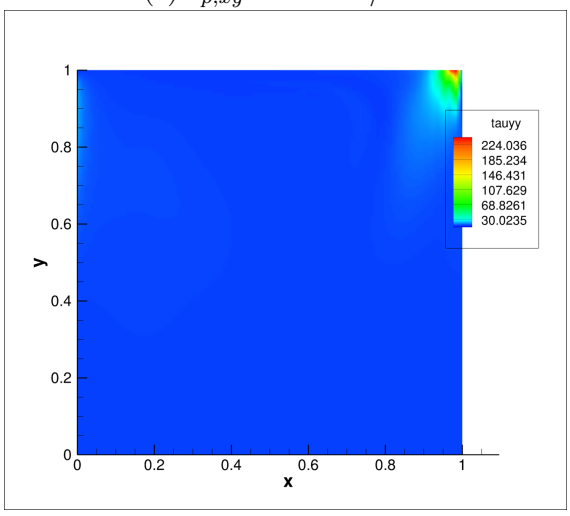
(b) $\tau_{p,xx}$ for $h = 1/512$



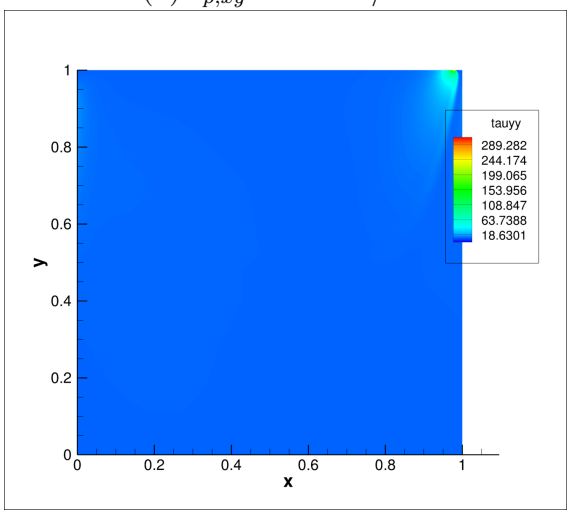
(c) $\tau_{p,xy}$ for $h = 1/128$



(d) $\tau_{p,xy}$ for $h = 1/512$



(e) $\tau_{p,yy}$ for $h = 1/128$



(f) $\tau_{p,yy}$ for $h = 1/512$

Figure 4.6: Influence of mesh refinement on polymer stress contours with $Wi = 1.0$

Wi	**Reference**	x	y
0.0	Su et al [28]	0.500245	0.763323
	Zhou et al [32]	0.500003	0.781014
	Gao et al [13]	0.5001	0.7765
	Current result	0.499	0.773
0.5	Pan et al [23]	0.4692	0.7982
	Sousa et al [27]	0.467	0.801
	Gao et al [13]	0.4702	0.7957
	Current result	0.467	0.799
1.0	Pan et al [23]	0.439	0.816
	Sousa et al [27]	0.434	0.814
	Gao et al [13]	0.4351	0.8132
	Current result	0.438	0.816

Table 4.1: Coordinates of vortex centers at different Wi number

Chapter 5

Summary

The major contribution of this work is to develop a discontinuous Galerkin solver for Oldroyd-B viscoelastic fluids and couple it with a mixed finite element solver for the Navier-Stokes equations. We implemented a DEVSS-type stabilization to introduce more ellipticity to the momentum equation. We validated our code against the analytic solution for the plane Poiseuille problem and we show numerical results from the lid-driven cavity problem.

The major conclusions of this work are as follows.

(i) The discontinuous Galerkin method with the Lax-Friedrichs flux performs well for the convection-dominated Oldroyd-B equation.

(ii) The DEVSS-type stabilization can effectively suppress numerical oscillations caused by abrupt changes in polymer stress, which may occur upon startup of the cavity flow as the Weissenberg number approaches zero.

(iii) Our numerical results on the plane Poiseuille problem match the analytic solution with high order mesh convergence

(iv) Our numerical results on the lid-driven cavity problem are consistent with those in the literature.

(v) With mesh refinement, our method can effectively capture the large polymer stress as well as its sharp transition layer at the top wall of the cavity.

Future plans for research is to involve the Cahn-Hilliard model in order to study interfacial dynamics in viscoelastic fluids.

Bibliography

[1] G. Alzetta, D. Arndt, W. Bangerth, V. Boddu, B. Brands, D. Davydov, R. Gassmöller, T. Heister, L. Heltai, K. Kormann, M. Kronbichler, M. Maier, J.-P. Pelteret, B. Turcksin, and D. Wells. The deal.ii library, version 9.0. *Journal of Numerical Mathematics*, 26(4):173–183, 2018.

[2] F. P. T. Baaijens. Mixed finite element methods for viscoelastic flow analysis: a review. *Journal of Non-Newtonian Fluid Mechanics*, 79:361–385, 1998.

[3] F. P. T. Baaijens, M. A. Hulsen, and P. D. Anderson. *The Use of Mixed Finite Element Methods for Viscoelastic Fluid Flow Analysis*, pages 1–18. American Cancer Society, 2017.

[4] W. Bangerth, R. Hartmann, and G. Kanschat. Deal.ii—a general-purpose object-oriented finite element library. *ACM Transactions on Mathematical Software*, 33(4): 24–es, 2007.

[5] R. B. Bird, R. C. Armstrong, and O. Hassager. *Dynamics of polymeric liquids, Volume 1: Fluid mechanics, 2nd Edition*. Wiley, 2nd edition, 1987.

[6] D.V. Boger and K. Walters. *Rheological Phenomena in Focus*, volume 4, pages 11–19. 1993.

[7] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods*. Springer Science Business Media, New York, 1991.

[8] B. Cockburn. Discontinuous Galerkin methods. *ZAMM - Journal of Applied Mathe-*

matics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik, 83(11): 731–754, 2003.

[9] P. Daripa and G Paşa. On the high weissenberg number problem. *Institutul de Matematica "Simion Stoilow" al Academiei Romane*, pages 1–21, 2017.

[10] T. A. Davis. Algorithm 832: Umfpack v4.3—an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software*, 30(2):196–199, 2004.

[11] R. Fattal and R. Kupferman. Constitutive laws for the matrix-logarithm of the conformation tensor. *Journal Non-Newtonian Fluid Mechanics*, 123:281–285, 2004.

[12] R. Fattal and R. Kupferman. Time-dependent simulation of viscoelastic flows at high weissenberg number using the log-conformation representation. *J. Non-Newtonian Fluid Mech.*, 126:23–37, 2005.

[13] P. Gao, J. Ouyang, and W. Zhou. Coupling of finite element method and discontinuous Galerkin method to simulate viscoelastic flows. *International Journal for Numerical Methods in Fluids*, 86(6):414–432, 2017.

[14] H. Giesekus. A simple constitutive equation for polymer fluids based on the concept of deformation-dependent tensorial mobility. *Journal of Non-Newtonian Fluid Mechanics*, 11(1):69–109, 1982. ISSN 0377-0257.

[15] S. Gottlieb and C.-W. Shu. Total variation diminishing Runge-Kutta schemes. 67: 73–85, 1998.

[16] J. L. Guermond, P. Minev, and Jie Shen. An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 195(44): 6011–6045, 2006.

[17] M. Hulsen, R. Fattal, and R. Kupferman. Flow of viscoelastic fluids past a cylinder at high weissenberg number: Stabilized simulations using matrix logarithms. *J. Non-Newtonian Fluid Mech.*, 127:27–39, 2005.

[18] A. Kaye. A bouncing liquid stream. *Nature*, 197:1001–1002, 1963.

[19] A. Kikker, F. Kummer, and M. Oberlack. A fully coupled high-order discontinuous Galerkin solver for viscoelastic fluid flow. *International Journal for Numerical Methods in Fluids*, 93, 2020.

[20] N. Kishi and H. Iizuka. The barus effect of polymer melts. *Journal of Polymer Science Part B: Polymer Letters*, 2(4):399–402, 1964.

[21] A.B. Metzner, E.A.Uebler, and C.F. Chan Man Fong. Converging flow of viscoelastic materials. *AIChE Journa*, 15:750, 1969.

[22] J. G. Oldroyd and Alan Herries Wilson. On the formulation of rheological equations of state. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 200(1063):523–541, 1950.

[23] T.-W. Pan, J. Hao, and R. Glowinski. On the simulation of a time-dependent cavity flow of an Oldroyd-B fluid. *International Journal for Numerical Methods in Fluids*, 60 (7):791–808, 2009.

[24] R. Poole. The Deborah and Weissenberg numbers. *The British Society of Rheology - Rheology Bulletin*, 53:32–39, 2012.

[25] R. J. Poole, M. A. Alves, and P. J. Oliveira. Purely elastic flow asymmetries. *Physical Review Letters*, 99:164503, 2007.

[26] M. E. Rosti, F. Picano, and L. Brandt. *Numerical Approaches to Complex Fluids*, pages 1–34. Springer International Publishing, Cham, 2019.

[27] R. G. Sousa, R.J. Poole, A.M. Afonso, F.T. Pinho, P.J. Oliveira, A. Morozov, and M.A. Alves. Lid-driven cavity flow of viscoelastic liquids. *Journal of Non-Newtonian Fluid Mechanics*, 234:129–138, 2016.

[28] J. Su, J. Ouyang, X. Wang, B. Yang, and W. Zhou. Lattice Boltzmann method for the simulation of viscoelastic fluid flows over a large range of weissenberg numbers. *Journal of Non-Newtonian Fluid Mechanics*, 194:42–59, 2013.

[29] N. P. Thien and R. I. Tanner. A new constitutive equation derived from network theory. *Journal of Non-Newtonian Fluid Mechanics*, 2(4):353–365, 1977.

[30] P. Yue. A thermodynamically consistent phase-field method for contact angle hysteresis. *Journal of Fluid Mechanics*, 899:A15, 2020.

[31] F. Yurun. A comparative study of the discontinuous Galerkin and continuous supg finite element methods for computation of viscoelastic flows. *Computer Methods in Applied Mechanics and Engineering*, 141(1):47–65, 1997.

[32] W. Zhou, J. Ouyang, X. Wang, J. Su, and B. Yang. Numerical simulation of viscoelastic fluid flows using a robust FVM framework on triangular grid. *Journal of Non-Newtonian Fluid Mechanics*, 236:18–34, 2016.