Complex Fluid Dynamical Computations via the Finite Volume Method

Faniry Nadia Zazaravaka Rahantamialisoa (zaza@aims.ac.za)
Department of Mathematics and Applied Mathematics
University of Cape Town

Supervised by: Dr Tiri Chinyoka
Center for Research in Computational and Applied Mechanics
.and.
Department of Mathematics and Applied Mathematics
University of Cape Town

15 02 2018

Thesis submitted in partial fulfilment of the requirements for the award of the degree of Master in Science

Abstract

Numerical simulations of the complex flows of viscoelastic fluids are investigated. The viscoelastic fluids are modelled, primarily, via the Johnson-Segalman constitutive model. Our Numerical approach is based on finite volume method, based on the Johnson-Segalman constitutive model and implemented on the OpenFOAM® platform. The Johnson-Segalman model also easily reduces to the Oldroyd-B model under certain conditions of the material parameters. Since computations using the Oldroyd-B model have been extensively documented in the literature, we take advantage of the mathematical modelling connection between the Johnson-Segalman and Oldroyd-B models to validate the accuracy of our Johnson-Segalman solver via reduction to the Oldroyd-B model. Numerical validation of our results is conducted via the most commonly used benchmark problems. The final aim of our work is to assess the viability and efficiency of our numerical solver via an investigation into the complex fluid dynamical processes associated with shear banding.

Keywords: Viscoelastic fluids; Johnson-Segalman model; finite volume method; OpenFOAM; shear banding.

Declaration

I, the undersigned, hereby declare that the work contained in this research project is my original work, and that any work done by others or by myself previously has been acknowledged and referenced accordingly.

Signed by candidate

Faniry Rahantamialisoa, February 15, 2018

Contents

List of Figures

1. Introduction

Viscoelastic materials play a widely important role in many industrial processes and products such as food processing, polymer processing, pharmaceutical products etc. Viscoelastic fluids are a class of non-Newtonian fluids whose complex behaviour exhibit both solid-like (elastic) and liquid-like (viscous) characteristics. Indeed, viscoelastic fluids when subjected to stress and deformation do not return immediately to their original configuration after the stress is removed. Due to their elastic property, they have partial memory which makes them different from simple Newtonian fluids. This complexity makes viscoelastic fluids more susceptible to instabilities under similar flow conditions as Newtonian fluids. A better understanding of these types of fluids is therefore fundamental to improving related industrial processes. The increased interest in understanding viscoelastic fluids and their flow properties is intricately connected to their mathematical modelling and analysis. Being costly to produce in general, experimental methods of viscoelastic fluid flow analysis would not be the most cost effective avenues of recourse. Since they are also modelled via coupled and highly nonlinear partial differential equations, computational approaches have tended to be the most efficient and cost effective avenues of analysis. Due to the increased computational capabilities, numerous studies and simulations of viscoelastic fluid flows have vastly increased over the past decades.

The governing equations for the flow of a general viscoelastic fluid under isothermal conditions are:

- The continuity equation, also known as the mass conservation equation;

$$\frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho + \rho \nabla \cdot \mathbf{u} = 0, \tag{1.0.1}$$

  which, under incompressible (constant density) conditions, reduces to,

$$\nabla \cdot \mathbf{u} = 0. \tag{1.0.2}$$

- The momentum equations;

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \nabla \cdot \underline{\underline{\boldsymbol{\sigma}}}, \tag{1.0.3}$$

  or, with the material derivative on the left hand side expanded,

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) \right) = -\nabla p + \nabla \cdot \underline{\underline{\boldsymbol{\sigma}}}. \tag{1.0.4}$$

  A stress constitutive model must also be specified to describe the viscoelastic behaviour of the extra stress tensor. The total stress tensor, $\underline{\underline{\boldsymbol{\sigma}}}$, consists of two contributions $\underline{\underline{\boldsymbol{\sigma}}} = \underline{\underline{\boldsymbol{\sigma}}}_S + \underline{\underline{\boldsymbol{\sigma}}}_p$, where,

  - $\underline{\underline{\boldsymbol{\sigma}}}_S = 2\eta_S \underline{\underline{\boldsymbol{D}}}$ is the solvent contribution and,
  - $\underline{\underline{\boldsymbol{\sigma}}}_p$ is the polymer contribution which is described by a constitutive equation for the respective viscoelastic fluid model.

Several stress constitutive equations have been developed to model various viscoelastic fluids and their respective behaviour. This work will only focus on two models, the Oldroyd-B model and the Johnson-Segalman model whose constitutive equations are respectively given by,

$$\underline{\underline{\boldsymbol{\sigma}}}_p + \lambda \overset{\triangledown}{\underline{\underline{\boldsymbol{\sigma}}}}_p = 2\eta_p \underline{\underline{\boldsymbol{D}}}, \tag{1.0.5}$$

$$\underline{\underline{\boldsymbol{\sigma}}}_p + \lambda \overset{\square}{\underline{\underline{\boldsymbol{\sigma}}}}_p = 2\eta_p \underline{\underline{\boldsymbol{D}}}. \tag{1.0.6}$$

where $\overset{\square}{\underline{\underline{\boldsymbol{\sigma}}}}_p$ is called the mixed convected derivative or the Gordon-Schowalter and defined as follows,

$$\overset{\square}{\underline{\underline{\boldsymbol{\sigma}}}}_p = (1 - \frac{\xi}{2}) \overset{\triangledown}{\underline{\underline{\boldsymbol{\sigma}}}}_p + \frac{\xi}{2} \overset{\triangle}{\underline{\underline{\boldsymbol{\sigma}}}}_p \tag{1.0.7}$$

with,

- $\overset{\triangledown}{\underline{\underline{\boldsymbol{\sigma}}}}_p$ referring to the upper convected derivative which is given by,

$$\overset{\triangledown}{\underline{\underline{\boldsymbol{\sigma}}}}_p = \frac{D}{Dt}\underline{\underline{\boldsymbol{\sigma}}}_p - \nabla \boldsymbol{u} \cdot \underline{\underline{\boldsymbol{\sigma}}}_p - \underline{\underline{\boldsymbol{\sigma}}}_p \cdot (\nabla \boldsymbol{u})^T,$$

- and $\overset{\triangle}{\underline{\underline{\boldsymbol{\sigma}}}}_p$ referring to the lower convective derivative which is given by,

$$\overset{\triangle}{\underline{\underline{\boldsymbol{\sigma}}}}_p = \frac{D}{Dt}\underline{\underline{\boldsymbol{\sigma}}}_p + \underline{\underline{\boldsymbol{\sigma}}}_p \cdot \nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T \cdot \underline{\underline{\boldsymbol{\sigma}}}_p.$$

The Johnson-Segalman model can therefore be expressed as,

$$\underline{\underline{\boldsymbol{\sigma}}}_p + \lambda \overset{\triangledown}{\underline{\underline{\boldsymbol{\sigma}}}}_p + \lambda\xi(\underline{\underline{\boldsymbol{\sigma}}}_p \cdot \underline{\underline{\boldsymbol{D}}} + \underline{\underline{\boldsymbol{D}}} \cdot \underline{\underline{\boldsymbol{\sigma}}}_p) = 2\eta_P \underline{\underline{\boldsymbol{D}}}. \tag{1.0.8}$$

Taking $\xi = 0$, reduces the Johnson-Segalman model to the Oldroyd-B model.

In Equations (1.0.1 – 1.0.8) $\mathbf{u}$ is the fluid velocity; ρ the fluid density; p the fluid pressure, t the time; η_S the solvent viscosity; η_p the polymer viscosity; λ is a relaxation time; $\underline{\underline{\boldsymbol{D}}} = \nabla \mathbf{u} + (\nabla \mathbf{u})^T$ the deformation rate tensor and $0 \leq \xi \leq 2$ is a material parameter.

The stress constitutive equation expresses the relation between the stress and the deformation rate. Combined with the governing equations of continuity and momentum, they collectively form a highly non linear system of partial differential equations. The numerical solutions of such systems of equations invariably leads to convergence and stability issues. For viscoelastic flows, these issues are especially relevant beyond some critical values of the Weissenberg number, leading to the well known High Weissenberg Number Problem (HWNP). The Weissenberg number is a non dimensional parameter that gives an indication of the ratio of the viscous to the elastic contributions.

The HWNP appears physically as a numerical breakdown and is independent of the choice of numerical method used, whether it be the finite difference method, the finite element method or the finite volume method. In this work we will adopt the finite volume method due to its versatility in dealing with complex geometries as well as due to its natural conservativeness properties. The HWNP is still an

open problem despite the effort of many researchers to address the issue [1], [4],[6], [7]. In so as far as finite volume method is concerned, many approaches have been developed to alleviate the HWNP but none of them seem to be able to tackle with the problem completely (see [3]). In fact, in [3] Chen at al. give a good comparison of some of the most popular methodologies in FVM that address the stability problem caused by the high Weissenberg number. The methods were assessed against their accuracy, their efficiency, their stability and the complexity of their implementation using three different geometries.

For the present work, we focus on only two of these methods which are the both side diffusion (BSD) and the log conformation reformulation by Fattal and Kuperfman (LCR) [4]. The BSD is the easiest to implement among the approaches mentioned in [3] and the LCR guarantees a higher convergence rate and offers a good stability with higher Wiessenberg numbers. These methods are implemented in our solver which is built on to the open-source platform OpenFOAM® (more precisely foam-extend: an extended version of OpenFOAM®) as developed by Jasak [10]. In fact, Favero et. al. developed a solver, on the OpenFOAM® platform, for viscoelastic flows based on the DEVSS method called *viscoelasticFluidFoam*. Many constitutive models are already built into this solver whose viability was extensively validated using benchmark 4:1 contraction flow problems. The Johnson-Segalman model is however not one of the models that has been built into the solver as yet. Details of the steps followed for the solver validation are given in [6] (mesh convergence, comparison of interpolations schemes and comparison with experimental data). Since its development, this solver has been used by many researchers for the analysis of viscoelastic flows, for instance [7] for non-isothermal flow, [8] for three dimensional simulation.

The solver based on the LCR method has just been recently released as an open-source package for OpenFOAM® even though, this approach had been used many times before in the literature. It was developed by Pimenta at al [15] and is called *rheoFoam*. In fact, when developed by Fattal and Kupperfman this method was first implemented through the finite difference method for the lid-driven cavity problem [5]. Other researchers tested the LCR stabilization method by using their in-house solvers, for example in [3] while others still used OpenFOAM but did not make the solver an open-source package as for example in [17]. Consequently, only limited data is available regarding the implementation of the LCR stabilization technique in open source solvers.

The main purposes of this work are therefore;

- to investigate the differences of the two stabilization methods aforementioned by using two of the most popular benchmarks problems. Indeed, Pimenta et. al. already provided a good comparison of the two methods and very useful data in [15] but by only using the planar 4:1 contraction flow. The lid-driven cavity flow problem will be also addressed in this work. This problem is indeed extremely suited for validation purposes. In spite of a simple geometry, the flow can present important instabilities.

- to implement the Johnson-Segalman model as it is not among the several models currently built into OpenFOAM®. Nevertheless, this model presents an interesting characteristic as it allows a non-monotonic relationship between the shear stress and the strain rate which is believed to be one of the causes of the shear banding phenomena.

- to investigate the shear banding phenomena, more precisely to validate the results obtained with the finite difference method in our previous work. In fact, in a previous research project (see [16]), we worked on a simple planar shear flow and observed the presence of shear bands near the walls for the Johnson-Segalman model.

This works is organised as follows. Chapter 2 recalls the basic principles of the finite volume method (FVM) as the numerical method on which the OpenFOAM® software is built. The chapter also introduces the two main stabilization techniques used for the numerical simulations of viscoelastic fluid flows. Chapter 3 presents the results and discussions for the benchmark flow problems. Our solver is developed for the Johnson-Segalman model and hence we validate its accuracy by reducing the model to the Oldroyd-B equations and testing the results against the Oldroyd-B results in the literature. We also then present the benchmark results for the full Johnson-Segalman model in planar contraction geometry and the lid-driven cavity. Chapter 4 deals with the shear-banding phenomena. Chapter 5 presents a summary discussion and shares some insights into possible future work.

2. Numerical Methodology

This section gives a general background of the numerical method used in this work, the finite volume method, as well as the software platform on which the numerical method will be implemented, OpenFOAM®.

2.1 The finite volume method

As with the finite difference and finite element methods, the finite volume method is a numerical method that is routinely used to approximate the solutions of a partial differential equation (PDE) or a system of partial differential equation (PDEs). The discretization of the computational domain in the finite volume method (FVM) consists of control volumes. The FVM is based on the integration of the governing equations over the control volumes.

2.1.1 Computational domain. The solution grid for the finite volume discretization is obtained by subdividing the domain into non overlapping control volumes (CV) as shown below. A couple of flat cells (or faces) delimit each CV and each pair of CVs shares only one face. Note that a CV can have any shape, generally a polyhedron.

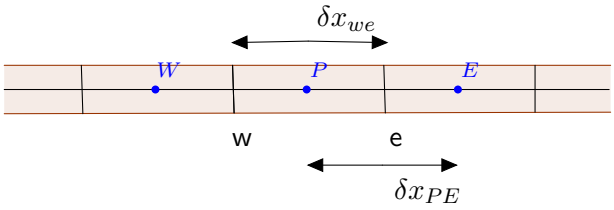


Figure 2.1: Example of a 1D computational grid

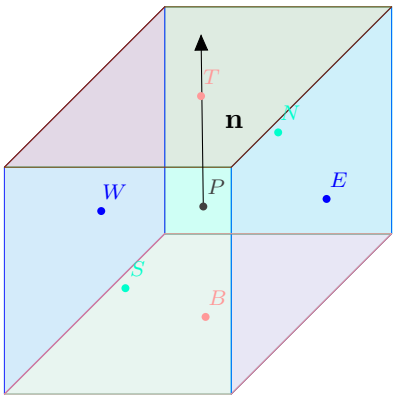


Figure 2.2: Example of a typical 3D CV: a hexahedral

The computational node is situated at the centroid of the CV and is denoted P. Its neighbours are

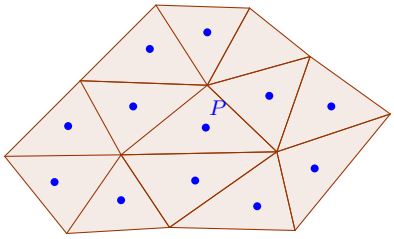


Figure 2.3: Example of 2D unstructured mesh.

referred as W, E, S, N, T, B depending on their position with respect to P (west, east, south, north, top or bottom). The step in space in a given direction is then given by the width of CV in that direction.

2.1.2 Principle of conservation for a general transport formula. To illustrate the principle of the FVM, let us consider the general transport equation for an arbitrary fluid property ϕ, [18].

$$\frac{\partial \rho\phi}{\partial t} + \nabla \cdot (\rho\phi\mathbf{u}) = \nabla \cdot (\Gamma \nabla\phi) + S_\phi, \tag{2.1.1}$$

where on the left hand side, we have respectively the transient term and the convective term, and on the right hand side we have the diffusion term, with diffusion coefficient Γ, and the source term S_ϕ.

The finite volume discretization for Equation (2.1.1) proceeds as follows:

- Integrate the equation over a CV;

$$\int_{\text{CV}} \frac{\partial \rho\phi}{\partial t}\, dV + \int_{\text{CV}} \nabla \cdot (\rho\phi\mathbf{u})\, dV = \int_{\text{CV}} \nabla \cdot (\Gamma\nabla\phi)\, dV + \int_{\text{CV}} S_\phi\, dV, \tag{2.1.2}$$

- Apply the divergence theorem to the convective and the diffusive terms in Equation (2.1.2);

$$\frac{\partial}{\partial t}\int_{\text{CV}} \rho\phi\, dV + \int_{\text{A}} \mathbf{n} \cdot (\rho\phi\mathbf{u})\, dA = \int_{\text{A}} \mathbf{n} \cdot (\Gamma\nabla\phi)\, dA + \int_{\text{CV}} S_\phi\, dV. \tag{2.1.3}$$

In Equation (2.1.3), A denotes the surface bounding the control volume CV.

2.1.3 Discretization in space. There are various ways of spatial discretization with respect to the FVM. In order to focus only on spatial discretization, let us consider the steady transport equation that is derived from Equation (2.1.3) by omitting the temporal terms,

$$\int_{\text{A}} \mathbf{n} \cdot (\rho\phi\mathbf{u})\, dA = \int_{\text{A}} \mathbf{n} \cdot (\Gamma\nabla\phi)\, dA + \int_{\text{CV}} S_\phi\, dV. \tag{2.1.4}$$

Equation (2.1.4) represents a steady convection-diffusion equation. The diffusion terms will almost always be treated stably via second order central difference schemes. Such schemes generally, however, cause spurious oscillations if applied to the convection terms. The difference schemes for the convection terms will therefore require deeper exploration including the possibility of using lower order schemes. Since the bounding surface (A) of each CV is composed of several (flat) faces, the surface integrals over

the bounding surface (A) is carried out (for any flow quantity $\mathbf{a}$) as a sum over the collective faces,

$$\begin{aligned}\int_{CV} \nabla \cdot \mathbf{a}\, dV &= \int_A \mathbf{n} \cdot \mathbf{a}\, dA \\ &= \sum_f \int_f \mathbf{n} \cdot \mathbf{a}\, dA,\end{aligned} \tag{2.1.5}$$

where f refers to the collective of bounding faces of A. The surface integrals are calculated as,

$$\int_f \mathbf{n} \cdot \mathbf{a}\, dA = \mathbf{A} \cdot \mathbf{a}_f, \tag{2.1.6}$$

where $\mathbf{A}$ is a vectorial representation of the bounding surfaces and $\mathbf{a}_f$ represents the value of the quantity $\mathbf{a}$ on face f. This leads to the final result,

$$\int_{CV} \nabla \cdot \mathbf{a}\, dV = \sum_f \mathbf{A} \cdot \mathbf{a}_f. \tag{2.1.7}$$

In evaluation of the sum in Equation (2.1.7) it is important to note the relationship between the direction of the normal vector, $\mathbf{n}_f$ (associated with the face area vector $\mathbf{A}_f$) and the position of the point P. If P belongs to a cell having the face (f), then this direction is outwards from P and conversely if the face (f) is among the neighbouring cells of P.

- **Central differencing scheme**
  The diffusive flux vector $\Gamma\nabla\phi)_f$ for the control surface element $\mathbf{A}_f$ in the diffusion terms,

  $$\int_{CV} \nabla \cdot (\Gamma \nabla \phi)\, dV = \sum_f \mathbf{A} \cdot (\Gamma \nabla \phi)_f\,, \tag{2.1.8}$$

  is approximated using second order central difference methods in the direction parallel to $\mathbf{n}_f$. As an example, let us consider a 1D problem (with a corresponding 1D computational grid) as illustrated in Figure 2.1. The central difference (CD) scheme is based on determining the required value at each face through linear interpolation using the cell center values. This mathematically translates to,

  $$(\nabla\phi)_w = \frac{\phi_P - \phi_W}{\delta x_{PW}}, \tag{2.1.9}$$

  and

  $$(\nabla\phi)_e = \frac{\phi_E - \phi_P}{\delta x_{EP}}. \tag{2.1.10}$$

  In the case of an uniform grid the diffusion coefficient at each face is obtained as an average value, as follows,

  $$\Gamma_f = \frac{\Gamma_P + \Gamma_{Neigh}}{2}, \tag{2.1.11}$$

  where the subscript $Neigh$ refers to the neighbouring nodal points.

If the CD scheme is employed to discretize the convective term, this leads to,

$$\int_{\text{CV}} \nabla \cdot (\rho\phi\mathbf{u})\, dV = \sum_f A \cdot (\rho\,\mathbf{u})_f \phi_f, \\ = \sum_f F_f\, \phi_f \tag{2.1.12}$$

where $F_f = (A \cdot \rho\,\mathbf{u})_f$ refers to the convective mass flux at each face, f. For an uniform mesh, similarly to the diffusion coefficient in Equation (2.1.11), this mass flux is obtained by using interpolation of the values of ρ and $\mathbf{u}$ at each face,

$$\phi_f = \frac{\phi_P + \phi_{Neigh}}{2}, \tag{2.1.13}$$

or in general,

$$\phi_f = d\phi_P + (1-d)\phi_{Neigh}, \tag{2.1.14}$$

where d here represents the distance ratio of f_{Neigh} and P_{Neigh}, $d = \dfrac{f_{Neigh}}{P_{Neigh}}$.

Even though the CD is second is second order accurate, it is generally not ideal in discretizing the convective terms. Lower order schemes may, in particular, be more appropriate. This results from the fact that there are other essential properties that characterize a discrete scheme, in particular conservativeness, boundedness and transportiveness, see for example [18]. The CD scheme cannot identify flow direction, which (flow direction) is crucially important with regards to convective processes. Diffusion processes on the other hand happen in every direction and are not limited to a particular flow direction and hence CD schemes are well suited to the discretization of diffusion terms. Spurious oscillations would be observed in the solution when the CD schemes are inappropriately employed to discretize flow-direction sensitive convective terms. In particular, the CD schemes would in such cases not satisfy, say, the transportiveness property.

- **Upwind differencing scheme**
  The flow direction is taken into account in the upwind differencing (UD) scheme. In this case, the value of any flow quantity (ϕ) at each face is evaluated as follows,

$$\phi_f = \begin{cases} \phi_P, \text{ if } F_f \geq 0 \\ \phi_{Neigh}, \text{ for other values of } F_f. \end{cases} \tag{2.1.15}$$

  The cell face value of ϕ is then considered to be equal to the upstream cell center value. The UD scheme therefore satisfies the transportiveness property. It is, however, only first order accurate.

- **Hybrid differencing scheme**
  The hybrid differencing scheme is also known as the blended differencing scheme. As its name, this scheme is based on the combination of the two schemes given above, the CD and UD schemes, giving rise to a scheme that has both better accuracy and satisfies transportiveness. The hybrid scheme is described mathematically as, see for example [10],

$$\phi_f = [(1-\gamma)\max(sign(F),0) + \gamma d]\phi_P + [(1-\gamma)\min(sign(F),0) + \gamma(1-d)]\phi_{Neigh}. \tag{2.1.16}$$

  where γ is the weighting parameter and $0 \leq \gamma \leq 1$. Taking $\gamma = 1$ gives the CD scheme whereas $\gamma = 0$ leads to the UD scheme.

- **Remarks** The source term S_ϕ groups all the remaining terms in the transport equation, i.e terms which do not fall into the diffusion, convection or transient categories. The goal is to estimate the volume integral associated with these source terms,

$$\int_{CV} S_\phi \, dV. \tag{2.1.17}$$

The simplest approximation is the product of the volume V_P, of the CV centered at P, with the mean value of S_ϕ (i.e. approximated, say, by the value at the center of the control volume $\bar{S}_\phi = S_{\phi_P}$),

$$\int_{CV} S_\phi \, dV = S_{\phi_P} \, V_P. \tag{2.1.18}$$

The approximation is exact if S_ϕ is constant or linear within CV. The approximation is second order accurate otherwise. Higher order approximation require more locations than just the center.

There are other high-order scheme based on the finite volume discretization such as those based on quadratic upwind (QUICK) or those with total variation diminishing (TVD) properties. Despite their higher order accuracy, such schemes are however susceptible to numerical stability issues. Nevertheless, Alves at al developed a high order scheme which offers a good iterative properties and better stability called CUBISTA (convergent and Universally Bounded Interpolation Scheme for the Treatment of Advection). This is based on a third order accurate QUICK scheme and the TVD constraints. Full details are given in [2].

2.1.4 Temporal discretization. For a transient problem, in addition to the integration over CV, we also need to integrate Equation (2.1.2) over a finite time step Δt,

$$\int_t^{t+\Delta t} \int_{\mathsf{CV}} \frac{\partial \rho\phi}{\partial t} \, dV dt + \int_t^{t+\Delta t} \int_{\mathsf{CV}} \nabla \cdot (\rho\phi \mathbf{u}) \, dV dt = \int_t^{t+\Delta t} \int_{\mathsf{CV}} \nabla.(\Gamma \nabla\phi) \, dV dt + \int_t^{t+\Delta t} \int_{\mathsf{CV}} S_\phi \, dV dt. \tag{2.1.19}$$

Substituting Equations (2.1.8), (2.1.12) and (2.1.18) into the above Equation (2.1.19) gives,

$$\int_t^{t+\Delta t} \int_{\mathsf{CV}} \frac{\partial \rho\phi}{\partial t} \, dV dt + \int_t^{t+\Delta t} \sum_f F_f \phi_f \, dt = \int_t^{t+\Delta t} \sum_f \mathbf{A} \cdot (\Gamma \nabla\phi)_f \, dt + \int_t^{t+\Delta t} S_{\phi_P} V_P \, dt. \tag{2.1.20}$$

Here, we have treated the CV as invariable with time. We will not consider the case of moving meshes, in which the mesh grid also varies in time. Using similar averaging volume integration ideas for the transient terms as already demonstrated with the source terms, Equation (2.1.20) becomes,

$$\int_t^{t+\Delta t} \left(\frac{\partial \rho\phi}{\partial t} \right)_P V_P + \int_t^{t+\Delta t} \sum_f F_f \phi_f dt = \int_t^{t+\Delta t} \sum_f \mathbf{A} \cdot (\Gamma \nabla\phi)_f \, dt + \int_t^{t+\Delta t} S_{\phi_P} V_P \, dt. \tag{2.1.21}$$

Taking into account the prescribed variation of ϕ in time, the transient integral is evaluated in a general way using the weighting parameter θ, where $0 \leq \theta \leq 1$, as follows,

$$\int_t^{t+\Delta t} \phi_f = [\theta\phi_f + (1-\theta)\phi_f^0]\Delta_t. \tag{2.1.22}$$

The transient derivative can be approximated as,

$$\frac{\partial \rho\phi}{\partial t} = \frac{\rho_P\phi_P - \rho_P\phi_P^0}{\Delta t}; \tag{2.1.23}$$

where the superscript "0" refers to the current time level t and the absence of such superscript refers to the future time level $t+\Delta t$. We remark that both the density and the diffusion coefficient are considered to be constant in time.

Equation 2.1.23 represents a backward difference discretization scheme used to approximate the temporal derivative. It is obtained using Taylor series expansion and is first order accurate.

Substituting Equations (2.1.22) and (2.1.23) into Equation (2.1.21) it follows that,

$$\begin{aligned}\frac{\rho_P\phi_P - \rho_P\phi_P^0}{\Delta t}V_P + \sum_f F_f\theta\phi_f\Delta_t + \sum_f F_f(1-\theta)\phi_f^0\Delta_t \\ = \sum_f \theta\Gamma\mathbf{A}\cdot(\nabla\phi)_f\Delta t + (1-\theta)\Gamma\mathbf{A}\cdot(\nabla\phi)_f^0\Delta t \\ + S_uV_P + \theta S_P\phi_PV_P\Delta t + (1-\theta)S_P\phi_P^0V_P\Delta t.\end{aligned} \tag{2.1.24}$$

The goal is to calculate the value of ϕ (at the nodal point P) at the next time level $(t+\Delta t)$. The following time discretization methods, which depend the on the value of θ, can be employed for this purpose, see for example [18].

- **Fully explicit scheme**
  Taking $\theta = 0$ leads to the explicit time discretization scheme. In this case, the flow variables at the next time level $(t+\Delta t)$ are calculated explicitly from values at the current time level, t. In particular, the discretized form of the transport equation given in Equation (2.1.24) then reads,

$$\rho_P\phi_P = \phi_P^0 + \frac{\Delta_t}{\rho_pV_p}\left(\sum_f F_f\phi_f^0\Delta_t + \Gamma\mathbf{A}\cdot(\nabla\phi)_f^0\Delta t + S_uV_P + S_P\phi_P^0V_P\Delta t\right) \tag{2.1.25}$$

  This method is first order accurate in time has very low computational costs as it allows direct computation of the quantities. This method, however, is not widely used due to its susceptibility to numerical instability as defined by the Courant condition. The Courant number is defined as,

$$\text{Co} = \frac{u_f\Delta_t}{\delta x_{P_{Neigh}}}.$$

  To ensure numerical stability, the Courant number must be less than 1.

- **Fully implicit scheme**
  Taking $\theta = 1$ leads to the fully implicit scheme. In this case, the flow variables at the next time level $(t + \Delta t)$ depend implicitly on the values of these quantities at the same time level, $(t + \Delta t)$. The time marching naturally depends on the initial values ϕ_P^0. At each stage of the computation, the values at the current time level will only arise from the discretization of the time derivative as all other values will be specified at the next time level, $(t + \Delta t)$. The fully implicit scheme is unconditionally stable but is still first order accurate in time. For this method, Equation (2.1.24) reduces to,

$$\rho_P \phi_P = \phi_P^0 + \frac{\Delta_t}{\rho_p V_p} \left(\sum_f F_f \phi_f \Delta_t + \Gamma \mathbf{A} \cdot (\nabla \phi)_f \Delta t + S_u V_P + S_P \phi_P V_P \Delta t \right). \tag{2.1.26}$$

- **Semi-implicit schemes**
  Values of θ in the range $0 < \theta < 1$ lead to a range of semi-implicit schemes. In such cases, the flow variables at the next time level $(t + \Delta t)$ are calculated from a weighted average of values at the current time level, t, as well as vales at the next time level, $(t + \Delta t)$.

  A popular semi-implicit scheme is the Crank-Nicholson method in which $\theta = 1/2$. This is second order accurate in time and is also unconditionally stable.

The fully implicit and semi-implicit methods require solving a (usually large) system of equations and hence generally result in high computational costs.

2.1.5 Pressure-velocity coupling. Most flows of practical interest involve variable (as opposed to constant) pressure. In such cases, the flow variables that would need to be solved for from the 3D governing equations are the three velocity components, $\mathbf{u} = [u_i]$ with $i \in \{1, 2, 3\}$; the pressure, p; and the nine extra-stress components, $\underline{\underline{\boldsymbol{\sigma}}} = [\sigma_{ij}]$ with $i, j \in \{1, 2, 3\}$. these 13 quantities must be calculated from the corresponding thirteen equations; the continuity equation; the three momentum equations; and the nine constitutive equations. Since the continuity equation is never solved directly, various methods have been developed to couple this equation with the momentum equations in order to resolve the pressure-velocity coupling.

- **The SIMPLE algorithm:** The Semi-Implicit Method for Pressure Linked Equations (SIMPLE) is the result of Patankar and Spalding's research, see ([14]). This is an iterative method to determine the pressure and the velocity fields. The method starts with an initial guess of the pressure and of the velocity field, say p^* and u^*. Pressure and velocity corrections p' and u' are introduced so that the correct pressure and the velocity fields can be written as,

$$p = p^* + p', \tag{2.1.27}$$
$$u = u^* + u'. \tag{2.1.28}$$

  To illustrate the SIMPLE algorithm, consider the semi-discretized form of the momentum equations in the form,

$$a_p \mathbf{u}_P + \sum_{Neigh} a_{Neigh} \mathbf{u}_{Neigh} = -(\nabla p)_P + b_p, \tag{2.1.29}$$

  where the left hand side of Equation (2.1.29) represents the discretization of the convective terms, $\nabla \cdot (\mathbf{uu})$, and b_P represents all the other remaining terms.

The intermediate velocity and pressure fields are required to satisfy Equation (2.1.29),

$$a_p \mathbf{u}^*{}_P + \sum_{Neigh} a_{Neigh} \mathbf{u}^*_{Neigh} = -(\nabla p^*)_P + b_p. \tag{2.1.30}$$

The velocity and pressure corrections, on the other hand, satisfy the reduced equation,

$$a_P \mathbf{u}'{}_P + \sum_{Neigh} a_{Neigh} \mathbf{u}'{}_{Neigh} = -(\nabla p')_P. \tag{2.1.31}$$

It therefore follows from Equations (2.1.30) and (2.1.31) that the intermediate velocity, $\mathbf{u}^*{}_P$, and the velocity correction, $\mathbf{u}'{}_P$, at the point P are respectively given by,

$$\mathbf{u}^*{}_P = \frac{1}{a^*{}_p} \left(- \sum a_{Neigh} \mathbf{u}^*_{Neigh} - (\nabla p^*)_P + b_p \right), \tag{2.1.32}$$

$$\mathbf{u}'{}_P = \frac{1}{a_P} \left(- \sum a_{Neigh} \mathbf{u}'_{Neigh} - (\nabla p')_P \right). \tag{2.1.33}$$

For the SIMPLE procedure, an approximation is introduced in a way that the general velocity correction is evaluated as:

$$\mathbf{u}' \approx \frac{-(\nabla p')_P}{a_P}. \tag{2.1.34}$$

Equation (2.1.34) therefore offers a way to compute the pressure correction, p' at each point P, via the continuity equation. Combined with Equations (2.1.32), the therefore also leads to the corresponding update to the velocity correction at P.

In summary, the SIMPLE algorithm consists of the following steps,

1) The predictor step: solving the momentum equation for the velocity field using the initial guesses.

2) The correction step: with the results from the predictor step, solving the pressure correction equation that has been derived from the continuity equation. Substituting Equations (2.1.32) and (2.1.34) in the discretized continuity equation gives an equation for p'.

3) Determining the correction for the velocity via the, now known, corrected pressure.

4) Solving the remaining discretized governing equations, say the stress constitutive equations.

These steps have to be repeated until convergence is reached and hence, the results obtained from the previous time-step will be used as the initial guesses for the next time-step.

• **Remark:** The discretized form of the continuity equation is,

$$\nabla \cdot \mathbf{u} = \sum_f S_f \mathbf{u}_f = 0. \tag{2.1.35}$$

With the correction step, this procedure may be subject to divergence. An under-relaxation factor α_p has to be used to ensure convergence and stability. This yields the new corrected pressure (from step 3) as,

$$p^{new} = p^* + \alpha_p p', \tag{2.1.36}$$

with $\alpha_p \in [0, 1]$. The choice of α is dependent on the flow characteristic. Very large or very small values are generally not recommended as they may lead to instability and/or slow convergence.

- **The SIMPLEC algorithm:** The Semi-Implicit Method for Pressure Linked Equations Consistent (SIMPLEC) is a modified version of the SIMPLE algorithm which addresses some of the shortcomings of the later method, especially with regards to stability. The SIMPLEC algorithm uses the following approximation for the velocity correction, see for example [15],

$$\mathbf{u'}_P \approx \frac{\sum_{Neigh} a_{Neigh} \mathbf{u'}_{Neigh}}{\sum_{Neigh} a_{Neigh}}. \tag{2.1.37}$$

The pressure correction is calculated from the continuity equation and the modified approximation,

$$\mathbf{u'}_P = -\frac{(\nabla p')_p}{a_P + \sum_{Neigh} a_{Neigh}}. \tag{2.1.38}$$

- **The PISO algorithm:** The Pressure Implicit with splitting of operators (PISO) algorithm, see for example [9, 18] is an extended version of SIMPLE algorithm. Another corrector step is added in the procedure. PISO starts with the 3 first steps of SIMPLE which give p' together with the first corrected velocity and pressure fields denoted respectively $\mathbf{u}^{**}$ and p^{**}. Introducing the second correction step leads to p'' as well as the twice-corrected pressure and velocity fields p^{***} and u^{***} that satisfying the momentum equations,

$$a_p \mathbf{u}_P^{***} + \sum_{Neigh} a_{Neigh} \mathbf{u}_{Neigh}^{**} = -(\nabla p^{***})_P + b_p, \tag{2.1.39}$$

where,

$$p^{***} = p^{**} + p''. \tag{2.1.40}$$

Since $\mathbf{u}^{**}$ is also required to satisfy the momentum equations,

$$a_p \mathbf{u}_P^{**} + \sum_{Neigh} a_{Neigh} \mathbf{u}_{Neigh}^{*} = -(\nabla p^{**})_P + b_p, \tag{2.1.41}$$

the second corrected velocity can be written as,

$$\mathbf{u}_P^{***} = \mathbf{u}^{**}{}_P + \frac{1}{a_P} \left(\sum_{Neigh} a_{Neigh} (\mathbf{u}_{Neigh}^{*} - \mathbf{u}_{Neigh}^{**}) - (\nabla p^{''})_P \right), \tag{2.1.42}$$

which after applying the continuity equation leads to the second pressure correction equation. Once this equation is solved, the second correct pressure can be obtained from (2.1.40) and hence the second corrected velocity follows.

2.2 OpenFOAM® and the viscoelastic fluid solvers

The Open-source Field Operation and Manipulation or OpenFOAM® is a software which was developed originally as proprietary FOAM software by Henry Weller and Hrvoje Jasak at Imperial College in 1989. They were motivated by the aim to develop a customizable and stronger simulation platform for computational fluid dynamics (CFD) than FORTRAN (widely used at that time). In 2004 the FOAM software project was turned into an open-source platform. After its first release in 1996 ([10]), FOAM was still indeed a commercial software application. OpenFOAM® is a toolbox consisting of a collection of applications created with C^{++} libraries. It is used to solve problems the are modelled via systems of partial differential equations. OpenFOAM® therefore find very wide application in science, engineering and commerce (say in financial mathematics). Since it is based on the finite volume method (FVM), OpenFOAM® find extensive application in continuum mechanics, mainly fluid flows problems.

The applications in OpenFOAM® can be classified as follows as either solvers or utilities.

- **Solvers:** Each solver is intended to do the actual resolution of a specific problem. OpenFOAM® offers a wide range of solvers, say related to fluid dynamics. Indeed there are solvers for incompressible flow (icoFOAM, simpleFOAM, etc.); for compressible flow; for heat transfer etc. Solvers have also been developed for non-Newtonian flows, say the flow of viscoelastic fluids (for example viscoelasticFluidFOAM, rheoFoam, etc.) This work indeed aims to contribute in the later direction, in developing a viscoelastic solver for the simulation of fluids modelled by the Johnson-Segalman constitutive equation.
- **Utilities:** These are the platforms on which the data manipulations are performed. They consist of the pre- and post-processing aspects of OpenFOAM®.

2.2.1 General organisation of a case in OpenFOAM®. Three basics folders are necessary to run a case in OpenFOAM®, [12],

- A **constant** directory which contains,
  - a subdirectory called *polyMesh* gathering all the information about the case geometry and the mesh,
  - files with all the physical properties of, say, the fluid and flow, for example *viscoelastic properties*.
- All the simulation settings are included in a **system** directory where at least the following three files must be present,
  - *controlDict* for all the time parameters,
  - *fvSchemes* where the discretization schemes are specified and,
  - *fvSolutions* for the interpolations.
- A **0** directory is a time directory and contains all the specifications for the initial and boundary conditions for all the flow quantities (pressure, velocity, extra stresses, etc.) After running the application for a particular case, other time directories are created for each time step.

2.2.2 The ViscoelasticFluidFoam solver. Developed by [6], this solver is designed, as its name suggests, for viscoelastic flow computations. As already mentioned, the High Weissenberg Number Problem (HWBP) presents challenges of numerical stability with regards to viscoelastic fluid flow computations. The *ViscoelasticFluidFoam* solver uses the Discrete Elastic Viscous Stress Split (DEVSS) stabilization

technique to mitigate against the HWNP. The DEVSS technique, also called the both-side diffusion technique, consists on adding a diffusion term on each side of the momentum equations. The momentum equation are therefore recast as,

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) - \kappa \nabla \cdot (\nabla \mathbf{u}) = -\nabla p + \nabla \cdot \underline{\underline{\boldsymbol{\sigma}}} - \kappa \nabla \cdot (\nabla \mathbf{u}), \tag{2.2.1}$$

where κ is a positive constant. Taking $\kappa = \eta_p$, i.e. the polymer viscosity coefficient, is generally considered to be a convenient choice. The terms on the left hand side of Equation (2.2.1) are treated implicitly and those on the right hand side are treated as explicit terms in the numerical discretization.

In summary, the resolution process used in the *ViscoelasticFluidFoam* solver is based on four steps summarized in the following algorithm,

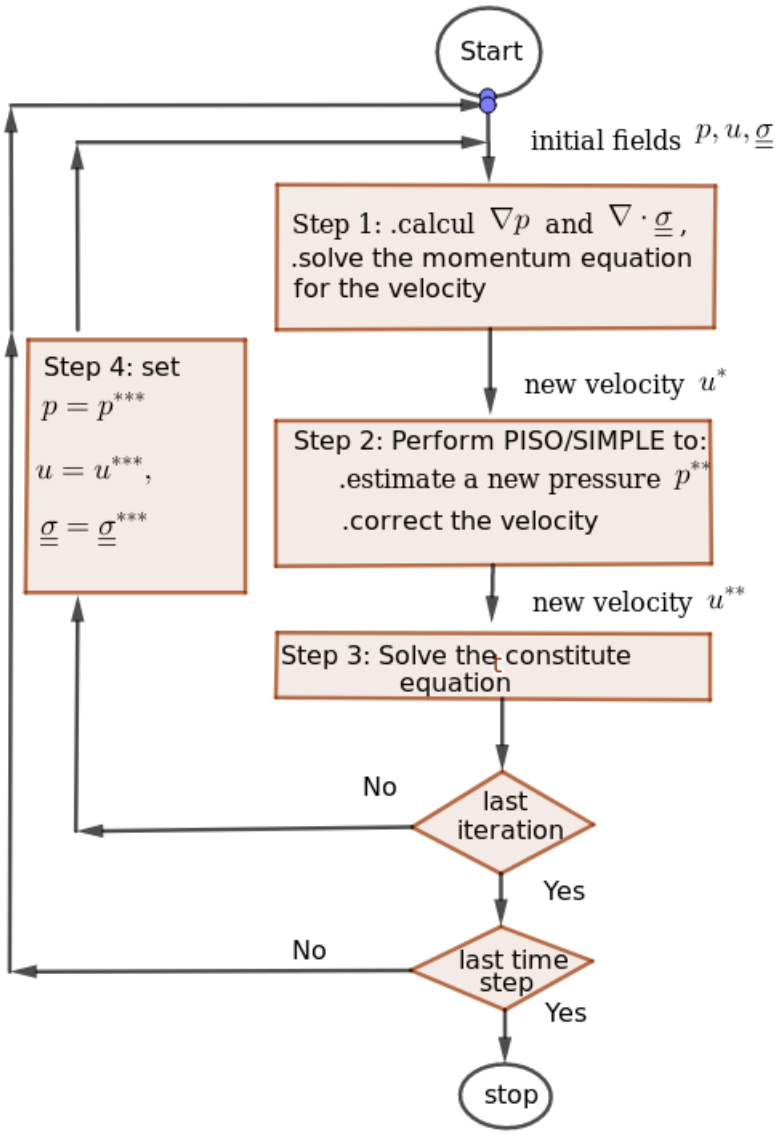


Figure 2.4: Computational steps of the *viscoelasticFluidFoam* solver

2.2.3 The RheoFoam solver. The DEVSS method is not entirely efficient in dealing with the HWNP, see also [3]. The development of other stabilization techniques, such as the Log Conformation Reformulation technique therefore remains important for viscoelastic flow computations.

The *RheoFoam* solver is found in *RheoTool*, a toolbox which was recently developed by [15], based on the OpenFOAM® software. Unlike the viscoelastic solver in OpenFOAM® which uses the both-side diffusion, the *RheoFoam* solver adopts the log-conformation formulation approach to deal with the HWNP. This is based on the work of [4] who reformulated the constitutive equations in terms of the logarithm of the conformation tensor instead of the stress tensor itself. It has been demonstrated that the numerical stability issues associated with the high Weissenberg number is related to the failure to represent large stress gradients which leads to the loss of positive definiteness. The log-conformation approach has been shown to preserve the positivity of the conformation tensor and hence enhance numerical stability. To illustrate this approach, let us consider the two constitutive equations of the two viscoelastic fluids that we will be using in this work. For two models the relation between the conformation tensor and the polymer extra stress tensor is given by,

$$\underline{\underline{\boldsymbol{\sigma}}}_p = \frac{\eta_p}{\lambda}(\underline{\underline{\boldsymbol{A}}} - \underline{\underline{\boldsymbol{I}}}). \tag{2.2.2}$$

The diagonalization of $\underline{\underline{\boldsymbol{A}}}$ yields,

$$\underline{\underline{\boldsymbol{A}}} = \underline{\underline{\boldsymbol{R}}} \cdot \underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\boldsymbol{R}}}^T, \tag{2.2.3}$$

where $\underline{\underline{\boldsymbol{R}}}$ is an orthogonal matrix and $\underline{\underline{\boldsymbol{\Lambda}}}$ is a diagonal matrix. These matrices are respectively composed of the eigenvectors and the eigenvalues of $\underline{\underline{\boldsymbol{A}}}$. For the reformulation purpose of the constitutive law, a new tensor $\underline{\underline{\boldsymbol{\Theta}}}$ is then introduced using the logarithm of $\underline{\underline{\boldsymbol{A}}}$,

$$\underline{\underline{\boldsymbol{\Theta}}} = \underline{\underline{\boldsymbol{R}}} \cdot \log \underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\boldsymbol{R}}}^T. \tag{2.2.4}$$

Let us also define the following tensors,

$$\begin{aligned} \underline{\underline{\boldsymbol{\Omega}}} = -\underline{\underline{\boldsymbol{R}}} \cdot \frac{D}{Dt}\underline{\underline{\boldsymbol{R}}}^T,\ \underline{\underline{\boldsymbol{L}}} = \nabla \boldsymbol{u}, \\ \underline{\underline{\tilde{\boldsymbol{\Omega}}}} = \underline{\underline{\boldsymbol{R}}}^T \cdot \underline{\underline{\boldsymbol{\Omega}}} \cdot \underline{\underline{\boldsymbol{R}}},\ \underline{\underline{\tilde{\boldsymbol{L}}}} = \underline{\underline{\boldsymbol{R}}}^T \cdot \underline{\underline{\boldsymbol{L}}} \cdot \underline{\underline{\boldsymbol{R}}}, \end{aligned} \tag{2.2.5}$$

where $\underline{\underline{\boldsymbol{\Omega}}}$ is an antisymmetric matrix. The material derivative of $\underline{\underline{\boldsymbol{\Theta}}}$ can be obtained via the chain rule,

$$\begin{aligned} \frac{D}{Dt}\underline{\underline{\boldsymbol{\Theta}}} &= \frac{D}{Dt}\underline{\underline{\boldsymbol{R}}} \cdot \log \underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\boldsymbol{R}}}^T + \underline{\underline{\boldsymbol{R}}} \cdot \frac{D}{Dt} \log \underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\boldsymbol{R}}}^T + \underline{\underline{\boldsymbol{R}}} \cdot \log \underline{\underline{\boldsymbol{\Lambda}}} \cdot \frac{D}{Dt}\underline{\underline{\boldsymbol{R}}}^T, \\ &= \underline{\underline{\boldsymbol{R}}} \cdot \underline{\underline{\tilde{\boldsymbol{\Omega}}}} \cdot \log \underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\boldsymbol{R}}}^T + \underline{\underline{\boldsymbol{R}}} \cdot \frac{D}{Dt}\underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\boldsymbol{\Lambda}}}^{-1} \cdot \underline{\underline{\boldsymbol{R}}}^T + \underline{\underline{\boldsymbol{R}}} \cdot \log \underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\tilde{\boldsymbol{\Omega}}}}^T \cdot \underline{\underline{\boldsymbol{R}}}^T, \\ \frac{D}{Dt}\underline{\underline{\boldsymbol{\Theta}}} &= \underline{\underline{\boldsymbol{R}}} \cdot \left(\underline{\underline{\tilde{\boldsymbol{\Omega}}}} \cdot \log \underline{\underline{\boldsymbol{\Lambda}}} + \frac{D}{Dt}\underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\boldsymbol{\Lambda}}}^{-1} + \log \underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\tilde{\boldsymbol{\Omega}}}}^T \right) \cdot \underline{\underline{\boldsymbol{R}}}^T. \end{aligned} \tag{2.2.6}$$

- **Oldroyd-B model**
  Substitution of Equation (2.2.2) and $\underline{\underline{\boldsymbol{L}}}$ in the constitutive equation of the Oldroyd-B gives,

$$\frac{\partial}{\partial t}(\underline{\underline{\boldsymbol{A}}} - \underline{\underline{\boldsymbol{I}}}) + \boldsymbol{u} \cdot \nabla(\underline{\underline{\boldsymbol{A}}} - \underline{\underline{\boldsymbol{I}}}) = (\underline{\underline{\boldsymbol{A}}} - \underline{\underline{\boldsymbol{I}}}) \cdot \underline{\underline{\boldsymbol{L}}} + \underline{\underline{\boldsymbol{L}}}^T(\underline{\underline{\boldsymbol{A}}} - \underline{\underline{\boldsymbol{I}}}) + (\underline{\underline{\boldsymbol{L}}} + \underline{\underline{\boldsymbol{L}}}^T) - \frac{1}{\lambda}(\underline{\underline{\boldsymbol{A}}} - \underline{\underline{\boldsymbol{I}}}). \tag{2.2.7}$$

Rearranging yields,

$$\frac{D}{Dt}\underline{\underline{\boldsymbol{A}}} = \underline{\underline{\boldsymbol{A}}} \cdot \underline{\underline{\boldsymbol{L}}} + \underline{\underline{\boldsymbol{L}}}^T \cdot \underline{\underline{\boldsymbol{A}}} + \frac{1}{\lambda}(\underline{\underline{\boldsymbol{I}}} - \underline{\underline{\boldsymbol{A}}}). \tag{2.2.8}$$

Substituting Equation (2.2.3)and the definitions given in Equation (2.2.5) and then multiplying the left hand side by $\underline{\underline{\boldsymbol{R}}}^T$ and the right hand side $\underline{\underline{\boldsymbol{R}}}$ yields,

$$\underline{\underline{\boldsymbol{R}}}^T \cdot \frac{D}{Dt}\underline{\underline{\boldsymbol{R}}} \cdot \underline{\underline{\boldsymbol{\Lambda}}} + \frac{D}{Dt}\underline{\underline{\boldsymbol{\Lambda}}} + \underline{\underline{\boldsymbol{\Lambda}}}\frac{D}{Dt}R^T\underline{\underline{\boldsymbol{R}}} = \underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\boldsymbol{R}}}^T \cdot \underline{\underline{\boldsymbol{L}}} \cdot \underline{\underline{\boldsymbol{R}}} + \underline{\underline{\boldsymbol{R}}}^T \cdot \underline{\underline{\boldsymbol{L}}}^T \cdot \underline{\underline{\boldsymbol{R}}} \cdot \underline{\underline{\boldsymbol{\Lambda}}} + \frac{1}{\lambda}\underline{\underline{\boldsymbol{I}}} - \frac{1}{\lambda}\underline{\underline{\boldsymbol{R}}} \cdot \underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\boldsymbol{R}}}^T, \tag{2.2.9}$$

or

$$\underline{\underline{\tilde{\boldsymbol{\Omega}}}} \cdot \underline{\underline{\boldsymbol{\Lambda}}} + \frac{D}{Dt}\underline{\underline{\boldsymbol{\Lambda}}} + \underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\tilde{\boldsymbol{\Omega}}}}^T = \underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\tilde{\boldsymbol{L}}}} + \underline{\underline{\tilde{\boldsymbol{L}}}}^T \cdot \underline{\underline{\boldsymbol{\Lambda}}} + \frac{1}{\lambda}(\underline{\underline{\boldsymbol{I}}} - \underline{\underline{\boldsymbol{\Lambda}}}). \tag{2.2.10}$$

Our aim is to obtain a reformulation of the constitutive equation in terms of $\underline{\underline{\boldsymbol{\Theta}}}$. Using Equation (2.2.10) we obtain expressions of the terms $\frac{D}{Dt}\underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\boldsymbol{\Lambda}}}^{-1}$ and $\underline{\underline{\tilde{\boldsymbol{\Omega}}}}$. Moreover, note that $\underline{\underline{\tilde{\boldsymbol{\Omega}}}}$ is also antisymmetric. In fact, we have,

$$\underline{\underline{\tilde{\boldsymbol{\Omega}}}}^T + \underline{\underline{\tilde{\boldsymbol{\Omega}}}} = \frac{D}{Dt}\underline{\underline{\boldsymbol{R}}}^T \cdot \underline{\underline{\boldsymbol{R}}} + \underline{\underline{\boldsymbol{R}}}^T \cdot \frac{D}{Dt} \cdot \underline{\underline{\boldsymbol{R}}}^T = \frac{D}{Dt}\underline{\underline{\boldsymbol{R}}}^T \cdot R = \frac{D}{Dt}\underline{\underline{\boldsymbol{I}}} = 0. \tag{2.2.11}$$

Consequently, the diagonal components of $\underline{\underline{\tilde{\boldsymbol{\Omega}}}} \cdot \underline{\underline{\boldsymbol{\Lambda}}} + \underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\tilde{\boldsymbol{\Omega}}}}^T$ will cancel out and thus from Equation (2.2.10) $\frac{D}{Dt}\underline{\underline{\boldsymbol{\Lambda}}}$ is a diagonal matrix whose entries are composed of the diagonal entries of the right hand side of Equation (2.2.10).

We obtain an expression for $\frac{D}{Dt}\underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\boldsymbol{\Lambda}}}^{-1}$ by defining another diagonal tensor $\underline{\underline{\tilde{\boldsymbol{B}}}}$ with $\underline{\underline{\tilde{\boldsymbol{B}}}}_{ii} = \underline{\underline{\tilde{\boldsymbol{L}}}}_{ii}$,

$$\frac{D}{Dt}\underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\boldsymbol{\Lambda}}}^{-1} = 2\underline{\underline{\tilde{\boldsymbol{B}}}} + \frac{1}{\lambda\underline{\underline{\boldsymbol{\Lambda}}}} - \frac{1}{\lambda}\underline{\underline{\boldsymbol{I}}}. \tag{2.2.12}$$

Since $\underline{\underline{\boldsymbol{\Lambda}}}$ is a diagonal matrix, Equation (2.2.10) leads to,

$$(\underline{\underline{\tilde{\boldsymbol{\Omega}}}} \cdot \underline{\underline{\boldsymbol{\Lambda}}} + \underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\tilde{\boldsymbol{\Omega}}}}^T)_{ij} = (\underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\tilde{\boldsymbol{L}}}} + \underline{\underline{\tilde{\boldsymbol{L}}}}^T \cdot \underline{\underline{\boldsymbol{\Lambda}}})_{ij}, \tag{2.2.13}$$

and hence,

$$\begin{aligned}(\underline{\underline{\tilde{\boldsymbol{\Omega}}}} \cdot \log\underline{\underline{\boldsymbol{\Lambda}}} + \log\underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\tilde{\boldsymbol{\Omega}}}})_{ij} &= \underline{\underline{\tilde{\boldsymbol{\Omega}}}}_{ij}(\log\underline{\underline{\boldsymbol{\Lambda}}}_{jj} - \log\underline{\underline{\boldsymbol{\Lambda}}}_{ii}) \\ &= \frac{\underline{\underline{\boldsymbol{\Lambda}}}_{ii}\underline{\underline{\tilde{\boldsymbol{L}}}}_{ij} + \underline{\underline{\tilde{\boldsymbol{L}}}}_{ji}\underline{\underline{\boldsymbol{\Lambda}}}_{jj}}{\underline{\underline{\boldsymbol{\Lambda}}}_{jj} - \underline{\underline{\boldsymbol{\Lambda}}}_{ii}}(\log\underline{\underline{\boldsymbol{\Lambda}}}_{jj} - \log\underline{\underline{\boldsymbol{\Lambda}}}_{ii}).\end{aligned} \tag{2.2.14}$$

Finally, substituting Equations (2.2.12) and (2.2.14) into Equation (2.2.6) gives the log-conformation reformulation of the constitutive equation for the Oldroyd-B model,

$$\frac{\partial}{\partial t}\underline{\underline{\boldsymbol{\Theta}}} + \boldsymbol{u} \cdot \nabla\underline{\underline{\boldsymbol{\Theta}}} = \underline{\underline{\boldsymbol{\Omega}}} \cdot \underline{\underline{\boldsymbol{\Theta}}} - \underline{\underline{\boldsymbol{\Theta}}} \cdot \underline{\underline{\boldsymbol{\Omega}}} + 2\underline{\underline{\boldsymbol{B}}} + \frac{1}{\lambda}(e^{-\underline{\underline{\boldsymbol{\Theta}}}} - \underline{\underline{\boldsymbol{I}}}), \tag{2.2.15}$$

where $\underline{\underline{\boldsymbol{B}}} = \underline{\underline{\boldsymbol{R}}} \cdot \underline{\underline{\tilde{\boldsymbol{B}}}} \cdot \underline{\underline{\boldsymbol{R}}}^T$.

- **Johnson-Segalman model**
  The Johnson-Segalman constitutive equation differs from the Oldroyd-B model by the term $-\lambda\xi(\underline{\underline{\boldsymbol{\sigma}}}_p \cdot \underline{\underline{\boldsymbol{D}}} + \underline{\underline{\boldsymbol{D}}} \cdot \underline{\underline{\boldsymbol{\sigma}}}_p)$. We therefore only need to obtain the corresponding log-conformation expression for that term.

  Substituting Equation (2.2.2) and $\underline{\underline{\boldsymbol{L}}}$ yields,

$$-\lambda\xi(\underline{\underline{\boldsymbol{\sigma}}}_p \cdot \underline{\underline{\boldsymbol{D}}} + \underline{\underline{\boldsymbol{D}}} \cdot \underline{\underline{\boldsymbol{\sigma}}}_p) = -\eta_p\xi(\underline{\underline{\boldsymbol{A}}} \cdot \underline{\underline{\boldsymbol{L}}} + \underline{\underline{\boldsymbol{A}}} \cdot \underline{\underline{\boldsymbol{L}}}^T + \underline{\underline{\boldsymbol{L}}} \cdot \underline{\underline{\boldsymbol{A}}} + \underline{\underline{\boldsymbol{L}}}^T \cdot \underline{\underline{\boldsymbol{A}}} - 2(\underline{\underline{\boldsymbol{L}}} + \underline{\underline{\boldsymbol{L}}}^T)). \tag{2.2.16}$$

  Substituting Equation (2.2.3) and multiplying on the left hand side by $\underline{\underline{\boldsymbol{R}}}^T$ and on the right hand side by $\underline{\underline{\boldsymbol{R}}}$ leads to,

$$-\lambda\xi(\underline{\underline{\boldsymbol{\sigma}}}_p \cdot \underline{\underline{\boldsymbol{D}}} + \underline{\underline{\boldsymbol{D}}} \cdot \underline{\underline{\boldsymbol{\sigma}}}_p) = -\eta_p\xi(\underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\tilde{\boldsymbol{L}}}}^T + \underline{\underline{\tilde{\boldsymbol{L}}}} \cdot \underline{\underline{\boldsymbol{\Lambda}}} + \underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\tilde{\boldsymbol{L}}}} + \underline{\underline{\tilde{\boldsymbol{L}}}}^T \cdot \underline{\underline{\boldsymbol{\Lambda}}} - 2(\underline{\underline{\tilde{\boldsymbol{L}}}} + \underline{\underline{\tilde{\boldsymbol{L}}}}^T)). \tag{2.2.17}$$

  Adding Equation (2.2.17) to Equation (2.2.10) yields,

$$\begin{aligned}\underline{\underline{\tilde{\boldsymbol{\Omega}}}} \cdot \underline{\underline{\boldsymbol{\Lambda}}} + \frac{D}{Dt}\underline{\underline{\boldsymbol{\Lambda}}} + \underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\tilde{\boldsymbol{\Omega}}}}^T = \underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\tilde{\boldsymbol{L}}}} + \underline{\underline{\tilde{\boldsymbol{L}}}}^T \cdot \underline{\underline{\boldsymbol{\Lambda}}} + \frac{1}{\lambda}(\underline{\underline{\boldsymbol{I}}} - \underline{\underline{\boldsymbol{\Lambda}}}) - \xi(\underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\tilde{\boldsymbol{L}}}}^T + \underline{\underline{\tilde{\boldsymbol{L}}}} \cdot \underline{\underline{\boldsymbol{\Lambda}}} \\ + \underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\tilde{\boldsymbol{L}}}} + \underline{\underline{\tilde{\boldsymbol{L}}}}^T \cdot \underline{\underline{\boldsymbol{\Lambda}}} - 2(\underline{\underline{\tilde{\boldsymbol{L}}}} + \underline{\underline{\tilde{\boldsymbol{L}}}}^T)).\end{aligned} \tag{2.2.18}$$

  Following the same principles as for the Oldroyd-B case, we need to get expressions for $\frac{D}{Dt}\underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\boldsymbol{\Lambda}}}^{-1}$ and $\underline{\underline{\tilde{\boldsymbol{\Omega}}}}$ from Equation (2.2.19).

  Since $\frac{D}{Dt}\underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\boldsymbol{\Lambda}}}^{-1}$ is a diagonal matrix, we have,

$$\left(\frac{D}{Dt}\underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\boldsymbol{\Lambda}}}^{-1}\right)_{ii} = 2\underline{\underline{\tilde{\boldsymbol{L}}}}_{ii} + \frac{1}{\lambda\underline{\underline{\boldsymbol{\Lambda}}}_{ii}} - \frac{1}{\lambda} - 4\xi\underline{\underline{\tilde{\boldsymbol{L}}}}_{ii} + \xi\frac{\underline{\underline{\tilde{\boldsymbol{L}}}}_{ii}}{\underline{\underline{\boldsymbol{\Lambda}}}_{ii}}. \tag{2.2.19}$$

  Moreover, we also have,

$$(\underline{\underline{\tilde{\boldsymbol{\Omega}}}} \cdot \log\underline{\underline{\boldsymbol{\Lambda}}} + \log\underline{\underline{\boldsymbol{\Lambda}}} \cdot \underline{\underline{\tilde{\boldsymbol{\Omega}}}})_{ij} = \frac{\underline{\underline{\boldsymbol{\Lambda}}}_{ii}\underline{\underline{\tilde{\boldsymbol{L}}}}_{ij} + \underline{\underline{\tilde{\boldsymbol{L}}}}_{ji}\underline{\underline{\boldsymbol{\Lambda}}}_{jj} - \xi((\underline{\underline{\boldsymbol{\Lambda}}}_{ii} + \underline{\underline{\boldsymbol{\Lambda}}}_{jj} - 2)(\underline{\underline{\tilde{\boldsymbol{L}}}}_{ij} + \underline{\underline{\tilde{\boldsymbol{L}}}}_{ji})}{\underline{\underline{\boldsymbol{\Lambda}}}_{jj}\underline{\underline{\boldsymbol{\Lambda}}}_{ii}}(\log\underline{\underline{\boldsymbol{\Lambda}}}_{jj} - \log\underline{\underline{\boldsymbol{\Lambda}}}_{ii}). \tag{2.2.20}$$

  The final expression of the reformulation of the Johnson-Segalman model with log-conformation is therefore given by,

$$\frac{\partial}{\partial t}\underline{\underline{\boldsymbol{\Theta}}} + \boldsymbol{u} \cdot \nabla\underline{\underline{\boldsymbol{\Theta}}} = \underline{\underline{\boldsymbol{\Omega}}} \cdot \underline{\underline{\boldsymbol{\Theta}}} - \underline{\underline{\boldsymbol{\Theta}}} \cdot \underline{\underline{\boldsymbol{\Omega}}} + 2\underline{\underline{\boldsymbol{B}}} + \frac{1}{\lambda}(e^{-\underline{\underline{\boldsymbol{\Theta}}}} - \underline{\underline{\boldsymbol{I}}}) - \xi(4\underline{\underline{\boldsymbol{B}}} - \frac{\underline{\underline{\boldsymbol{B}}}}{\underline{\underline{\boldsymbol{\Lambda}}}}). \tag{2.2.21}$$

- **Other numerical considerations for the viscoelastic fluid solver** Apart from new stabilization approaches, other numerical considerations have also implemented in the original viscoelastic fluid solver in OpenFOAM®. For example, the most recent solver uses the SIMPLEC algorithm to deal with the pressure-velocity coupling instead of the PISO algorithm. Additionally, to avoid the stress

fields from decoupling from the velocity fields, a new term is added to the momentum equations, see for example [15],

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) - (\mu_s + \eta_p)\nabla \cdot (\nabla \mathbf{u}) = -\nabla p + \nabla \cdot \underline{\overline{\boldsymbol{\sigma}}} - \overline{\nabla \cdot \eta_p(\nabla \mathbf{u})} + \mathbf{f}, \tag{2.2.22}$$

where $\overline{\nabla \cdot \eta_p(\nabla \mathbf{u})}$ is called the divergence of the velocity – a special second derivative. Due to this latter, there is a fourth-order derivatives in the momentum equation that will be canceled out with mesh refinement. To achieve this numerically in the FVM, a linear interpolation of the velocity gradient on the cell centers is used to determine the velocity gradient at the faces. The solving steps of the solver can be summarized in the following diagram,

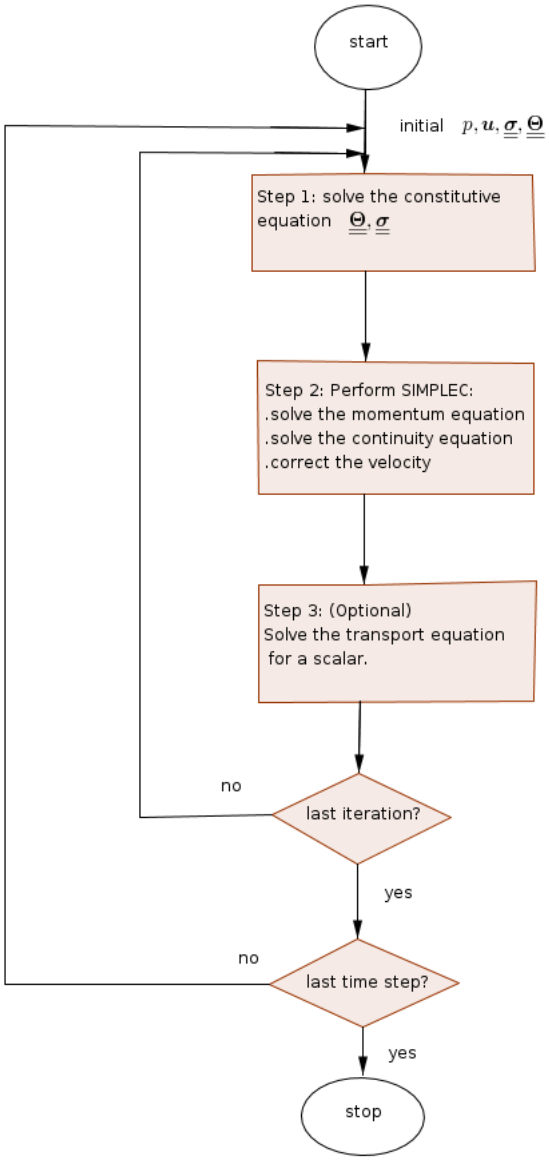


Figure 2.5: Computational steps of the *rheoFoam* solver

3. Benchmark solutions and code validation

In this chapter, we develop a solver for the numerical simulation of flows of viscoelastic fluids which are modelled by the Johnson-Segalman constitutive equations. The solver is based on the finite volume method and is implemented on the OpenFOAM® software platform. Taking the material parameter $\xi = 0$ in the Johnson-Segalman model reduces the model to the Oldroyd-B constitutive equations. Benchmark computational results for the Oldroyd-B model are widely documented in the literature. The accuracy of our solver will therefore be demonstrated under the conditions where $\xi = 0$, in which case our Johnson-Segalman solver is expected to reduce to an Oldroyd-B solver. We focus attention on two widely used and well documented benchmark problems, the 4:1 contraction flow and the lid-driven cavity flow are.

An additional aim of this chapter would then be to present our novel results for the benchmark solutions of flows of viscoelastic fluids which are governed by the full Johnson-Segalman model, i.e. for the case $\xi \neq 0$. We will also investigate the effects of numerical stabilization by implementing both the log-conformation reformulation approach and the DEVSS technique.

3.1 Benchmarking via the 4:1 contraction flow problem

For the contraction flow validation, we will only focus on the DEVSS stabilization technique and will therefore not compare with the results for a log-conformation reformulation approach. A detailed comparison of the two stabilization techniques for this benchmark flow of Oldroyd-B fluids can be found in [15].

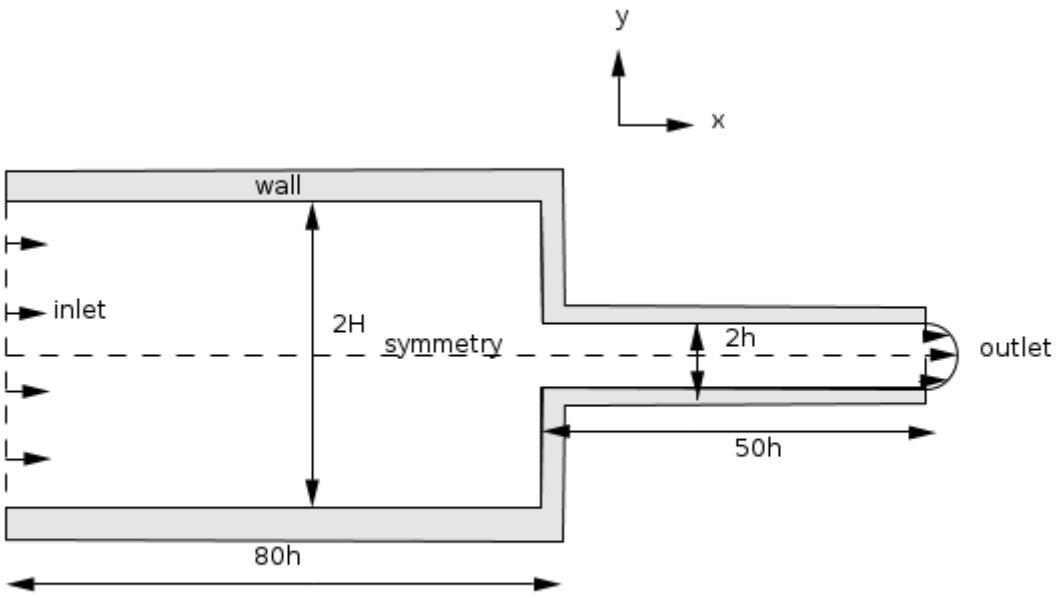


Figure 3.1: Geometry of a $4:1$ planar contraction flow

3.1.1 Schematic geometry and the flow conditions. Figure 3.1 shows a schematic of the planar contraction flow with contraction ratio 4:1 obtained from the ratio of the channel widths $H : h$. In particular, the upstream channel width is $2H$ and the downstream channel width is $2h$.

We assume zero initials conditions for all flow variables (pressure, velocity and extra stresses) and unless otherwise stated the following boundary conditions are considered,

- at the walls: the no-slip conditions for the velocity;
- at the centerline: the symmetry condition for the velocity;
- at the entrance: the velocity value is obtained by dividing the prescribed downstream average velocity by four and the stress tensor is considered null;
- at the exit: the velocity gradient along the x-direction is taken to be equal to zero as well as the value of the pressure. While not stated, the boundary condition used for the stress tensor is the zero gradient condition.

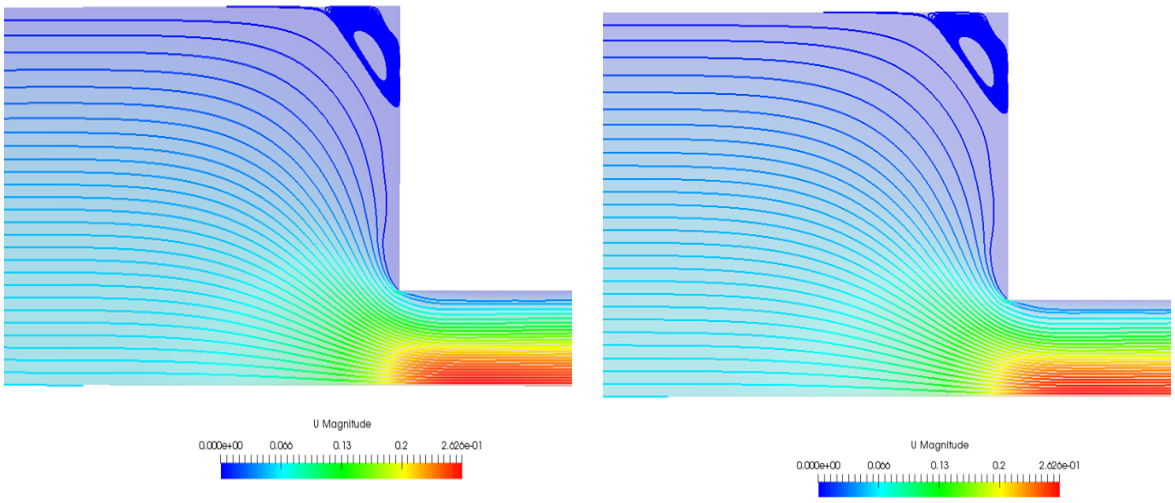
To discretize the computational domain, a hexahedral mesh is used. The computational domain consists of 20,700 control volumes or cells. In addition, a higher refinement is made near the walls and in the vicinity of the contraction. For the *viscoelasticFluidFoam* solver that we will adopt; the Crank-Nicholson scheme is employed for the temporal discretization, the upwind difference scheme for the discretization of the convective terms and finally the central difference scheme for the remaining terms.

It is worth mentioning that when using the *rheoFoam* solver, some modifications are made to the flow conditions. In particular, upstream and downstream of the contraction which both have the same width, the computational domain is composed of 11,991 cells with a higher refinement also near the walls. The CUBISTA high resolution technique is adopted for the discretization of the convective terms.

3.1.2 Results. Unless otherwise stated, the following parameter values will be assumed respectively for the Wissenberg number, the Reynolds number, the Johnson-Segalman material parameter and the relaxation time; Wi $= 2.9$, $Re = 0.56$, $\xi = 0.8$ and $\lambda = 0.03$.

The computational results were obtained with the modified *viscoelasticFluidFoam* solver which we in particular modified as a solver for flows of fluids described by the Johnson-Segalman model. For comparison purposes with the Oldroyd-B results from the literature, say for example the results of [6], we will take $\xi = 0$ in our computations. With $\xi = 0$, our Johnson-Segalman solver must expectedly reduce to an Oldroyd-B solver.

Figure 3.2 shows the results for the streamlines over the flow geometry. Our results (based on our Johnson-Segalman solver) for the case $\xi = 0$ produces similar results to those for the Oldroyd-B case as computed in [6].



Oldroyd-B results of [6] Johnson-Segalman for $\xi = 0$

Figure 3.2: Streamlines computed using Wi $= 2.9$

Figure 3.3 gives the upstream shear-stress profiles. As with the streamlines in Fig. 3.2, our results (based on our Johnson-Segalman solver) for the case $\xi = 0$ produces similar shear-stress profiles to those for the Oldroyd-B case as computed in [6].

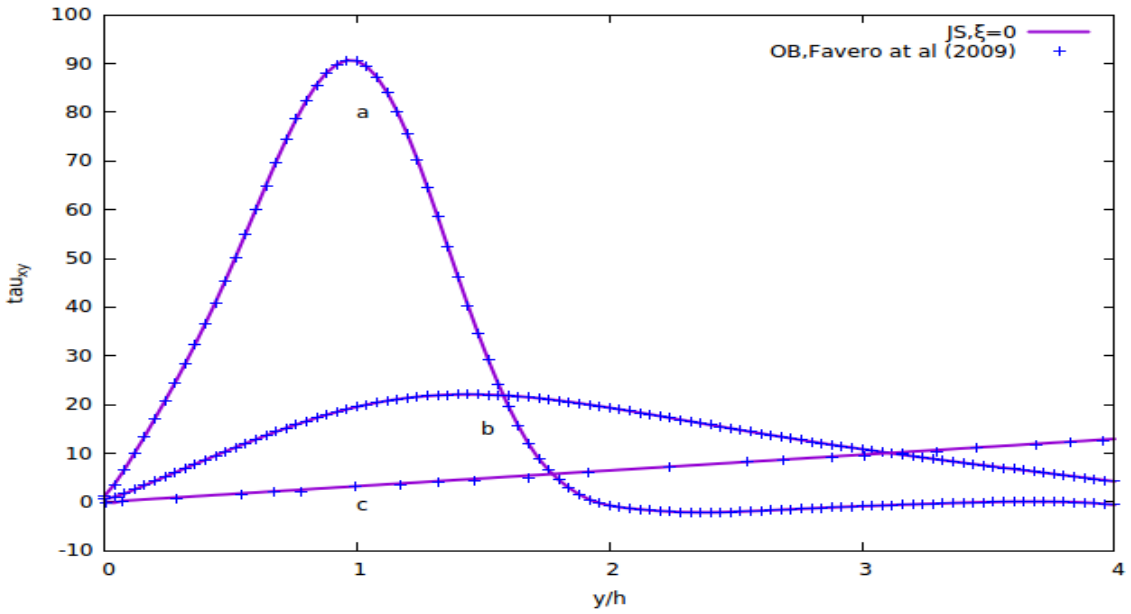


Figure 3.3: shear stress profiles along the y-axis at $a = 0.254m$, $b = 0.251m$, $c = 0.224m$.

Figures 3.2 and 3.3 provide robust validation our solver for the planar 4:1 contraction flow and for the case $\xi = 0$ based on published results in the literature. We therefore proceed to present our solver's novel prediction for non-zero values of ξ. We limit attention to the case $\xi = 0.8$. Since there is no data in the literature to validate our results (for $\xi \neq 0$) our corresponding results in this case will hopefully serve as the basis for future validation studies by other researchers. Figure 3.4 gives our streamline results for the case $\xi = 0.8$ with all other parameter values kept the same as in Fig. 3.2.

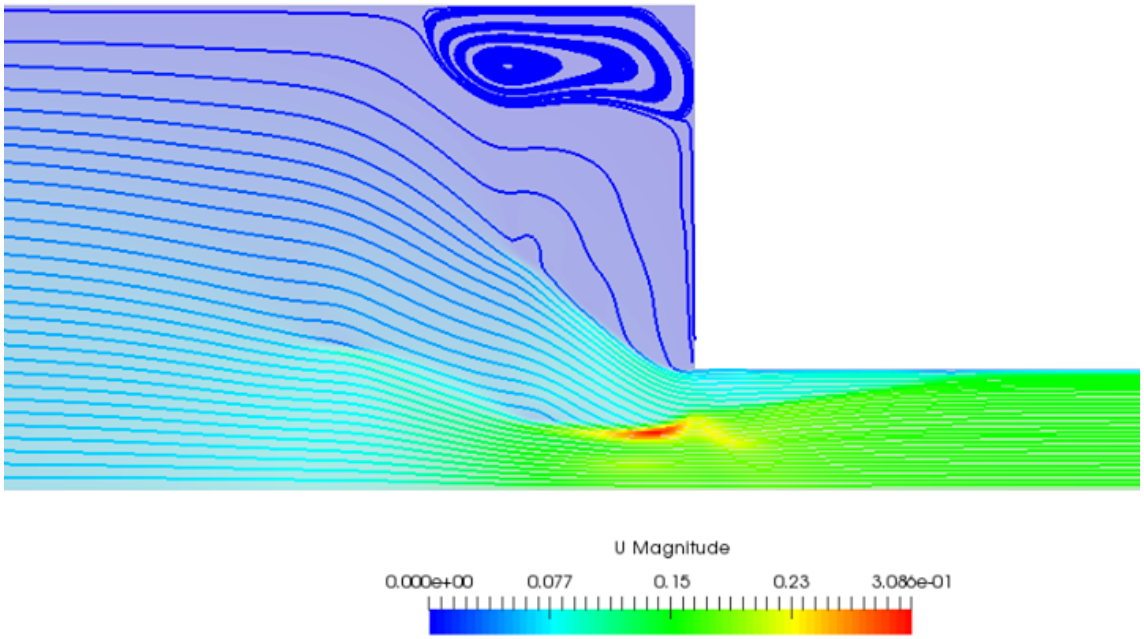


Figure 3.4: Johnson-Segalman results for Wi $= 2.9$ and $\xi = 0.8$

Compared to the streamlines for the Oldroyd-B model, we notice a drastic change in the streamlines and vortices for the Johnson-Segalman model in the vicinity of the contraction. To further assist with future cross-validation efforts by other researchers, Fig. 3.5 gives the velocity distribution characteristics near the contraction for the case $\xi = 0.8$.

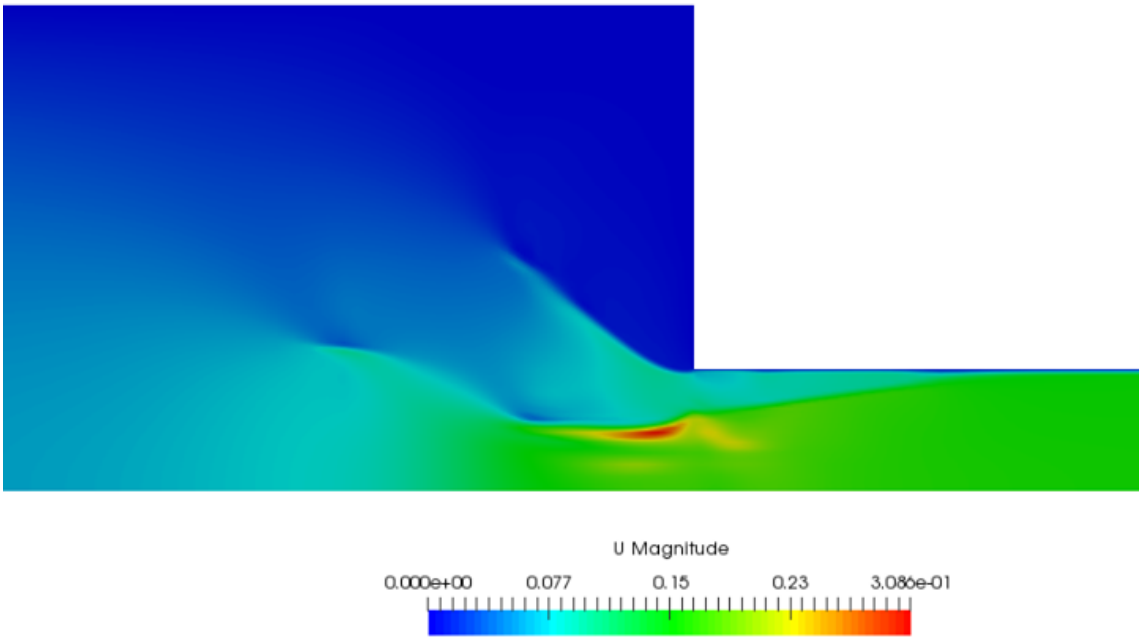


Figure 3.5: velocity distribution near the contraction for JS with $\xi = 0.8$.

3.2 The Lid-driven cavity

3.2.1 Presentation of the geometry and the flow conditions. The lid-driven cavity is characterized by a square cavity whose top wall moves in its own plane, while the bottom and side walls are kept stationary, see Fig. 3.6.

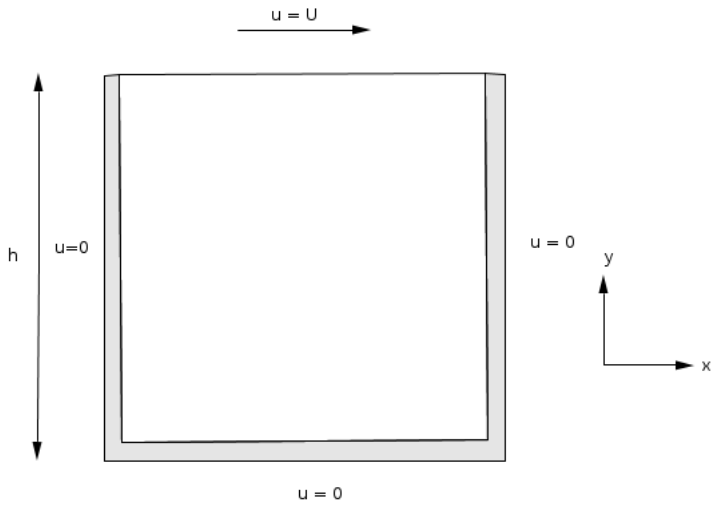


Figure 3.6: Geometry of the lid-driven cavity problem

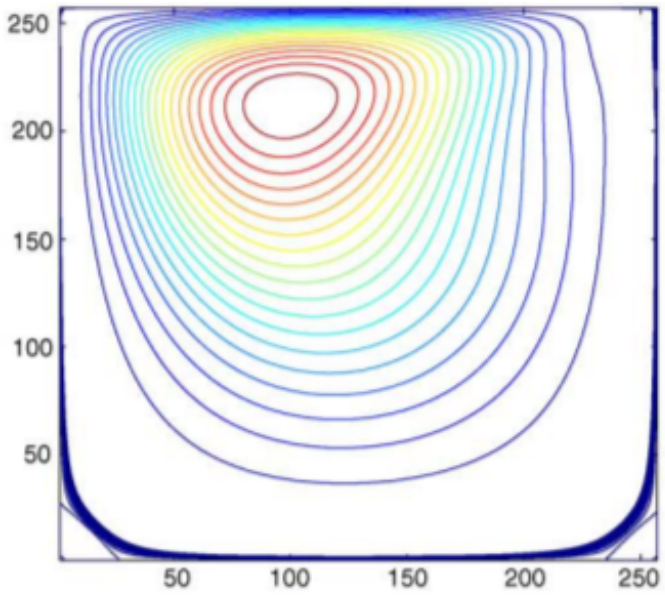
No-slip and wall impermeability conditions are imposed at the boundaries. The normal pressure is assumed to be null both at the moving and the stationary walls. In line with the stability prescriptions from the existing literature, see for example [3] and [15], instead of the constant speed U, the speed of the top wall is modified to variable speed $U_{lid}(x,t)$ which varies in space and time according to the formula,

$$U_{lid}(x,t) = 8U[1 + tanh(8(t-0.5))]x^2(1-x)^2, \tag{3.2.1}$$

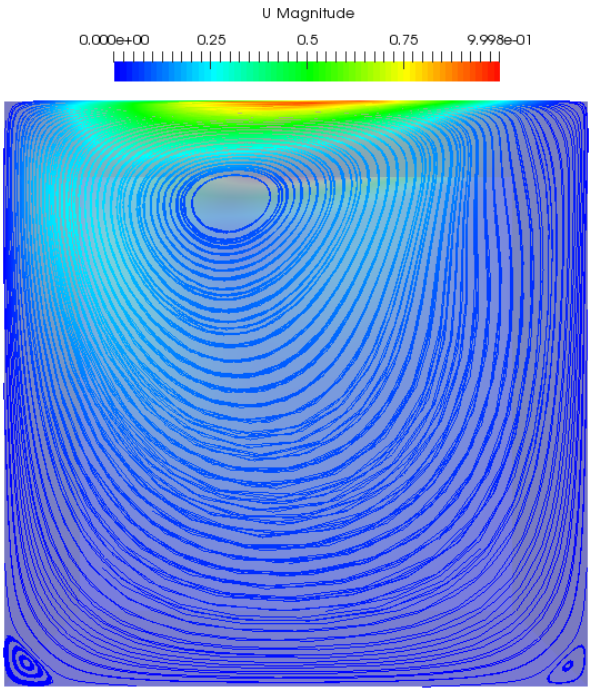
Similar boundary conditions as for the planar 4:1 contraction case are adopted for the extra-stress and the pressure for the log-conformation reformulation (LCR) based solver while the zero gradient boundary condition is assumed everywhere for the extra-stress and the pressure for the discreet elastic-viscous stress split (DEVSS) based solver. A hexahedral mesh is used for both approaches with 16129 control volumes (more precisely 127 cells uniformly distributed in both directions).

3.2.2 Results. In this section, as for the planar contraction flow we present our results for the case $\xi = 0$ for comparison with the Oldroyd-B results in the literature. Additionally, since for the lid-driven cavity flow the computations using both the LCR and DEVSS stabilization techniques are well documented in the literature, we will also therefore investigate the differences in results obtained via these two approaches.

Figure 3.7 shows that our results for Wi $= 2$ using LCR compare favourably with the results presented in [5]. Recall that the finite difference method was used in [5] and the results taken into consideration were obtained using the 256×256 point grid. We notice that with the finite volume method the corner vortices are more defined.



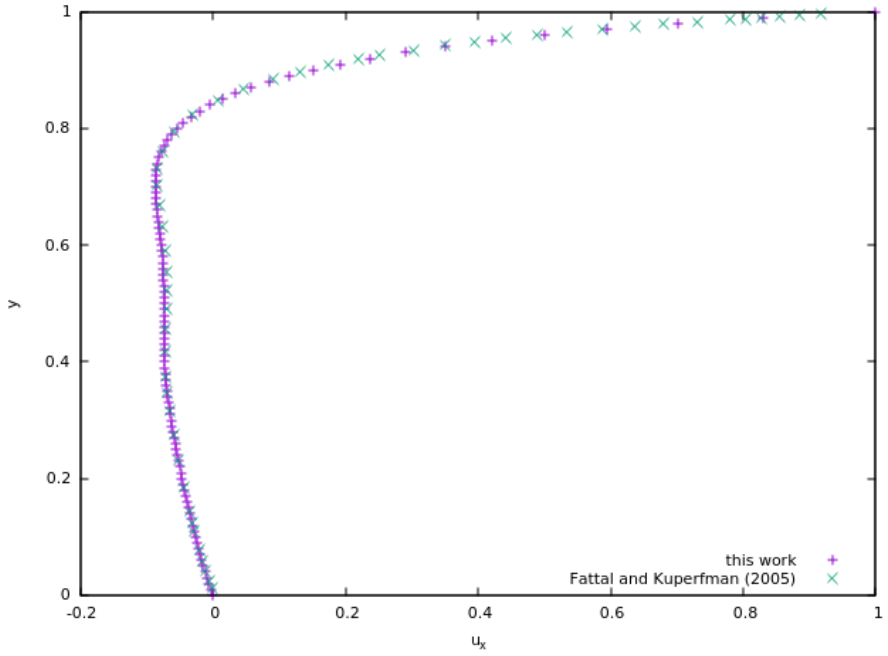
(a) Oldroyd-B streamlines for Wi $= 2$ and Re $\to 0$ taken as is from Fig.6 in [5] for N$=$ 256.



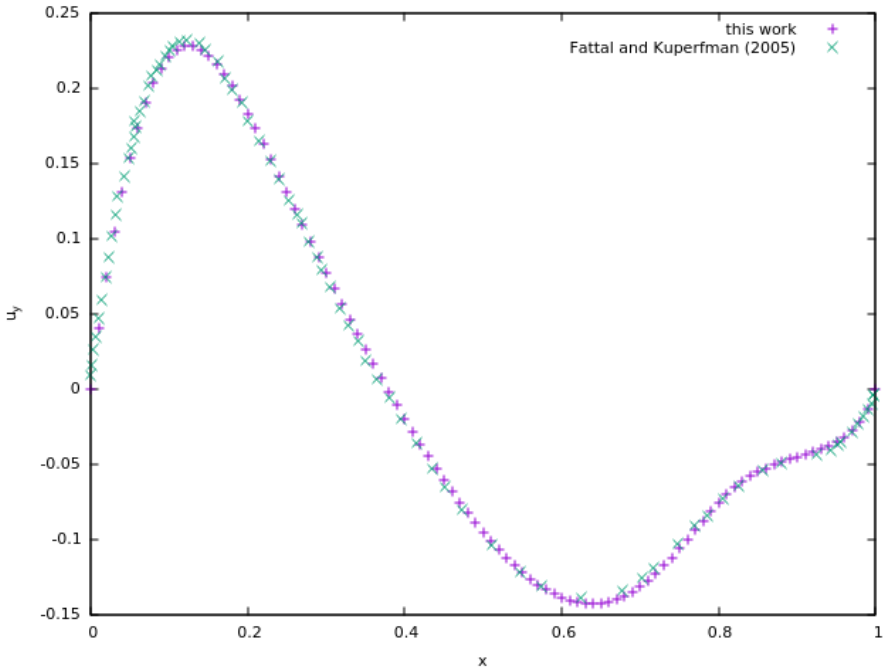
(b) Our Johnson-Segalman ($\xi = 0$) streamlines for Wi $= 2$ and Re $\to 0$

Figure 3.7: Comparison of streamlines in [5] with our results for Wi $= 2$

Figure 3.8 gives the variation of the vertical velocity along $y = 0.75$ as well as the horizontal velocity along $x = 0.5$. Again, our results ($\xi = 0$) are similar to the corresponding Oldroyd-B results of [5].



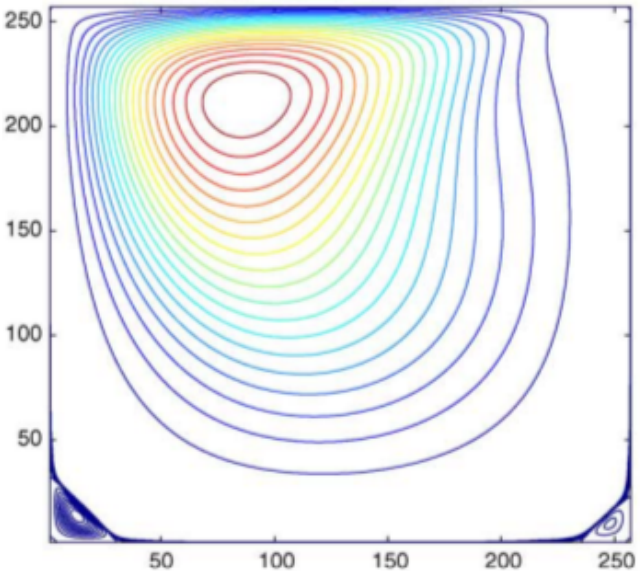
(a) Velocity at x = 0.5



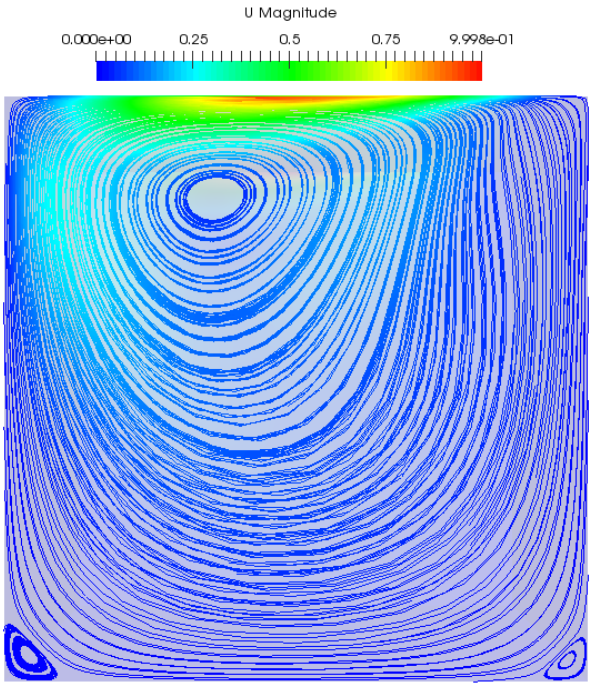
(b) Velocity at y = 0.75

Figure 3.8: Comparison of results of our the velocity profiles along $x = 0.5$ (a) and $y = 0.75$ (b) with the results of Fig. 7 of [5]

Figure 3.9 shows similar favourable agreement of our (LCR) results for Wi $= 3$ with the results of [5].



(a) Oldroyd-B streamlines for Wi $= 3$ and Re $\to 0$ taken as is from Fig.9 in [5] for N= 256.



(b) Our Johnson-Segalman ($\xi = 0$) streamlines for Wi $= 3$ and Re $\to 0$

Figure 3.9: Comparison of streamlines in [5] with our results for Wi $= 3$

We computed the results of Figure 3.10 using the DEVSS method. We notice also that our computations with the DEVSS method for the lid-driven cavity flow produces similar results as presented in [3]. In this case, we use the same flow conditions as indicated in [3]. For instance instead of the prescribed boundary condition for the stresses in the LCR solver, we use the zero gradient conditions. However, unlike Chen at al. who used a computational grid with 6400 control volumes, we kept the same mesh as described before.

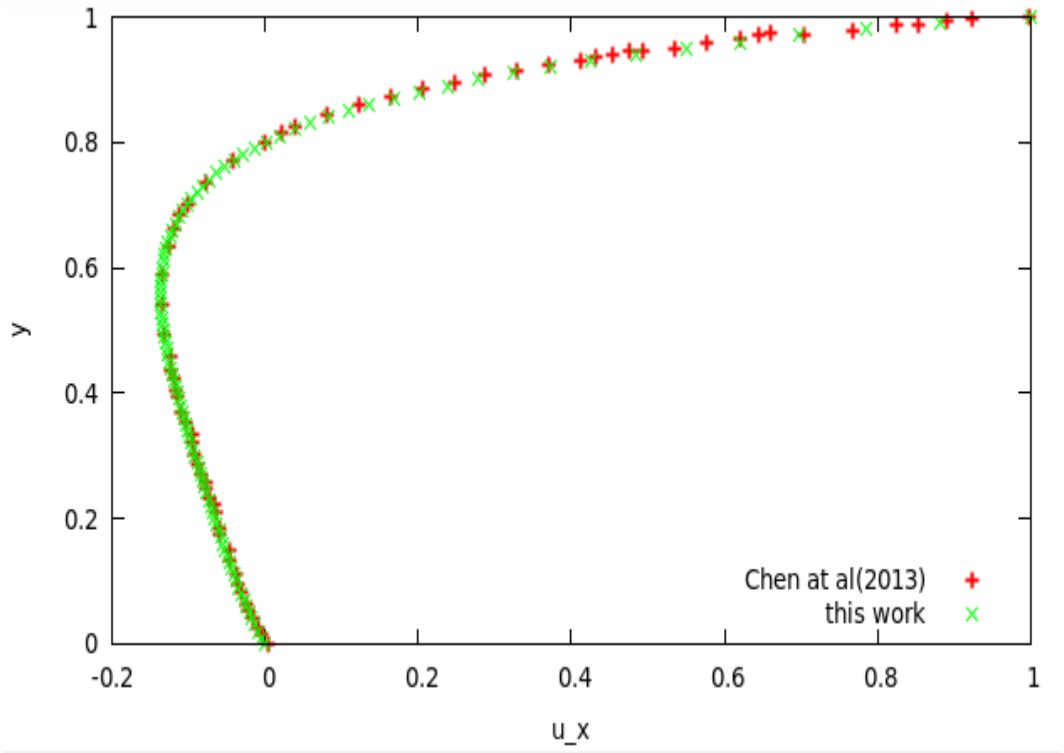


Figure 3.10: Comparison of our DEVSS results for the velocity component along $x = 0.491$ with $\mathrm{Wi} = 0.5$ and $\mathrm{Re} \to 0$ with those of [3].

For the following validation results, we consider a steady, incompressible and isothermal flow of an Oldroyd-B fluid in a lid-driven cavity as described in [19]. As before, our results are computed from our Johnson-Segalman solver with the relevant parameter values, (e.g. $\xi = 0$ reduces our model to an Oldroyd-B one and $\mathrm{Wi} = 0$ reduces our model to a Newtonian model etc.)

Taking $\mathrm{Wi} = 0$ shows that, see in Figures 3.11 and 3.12, our results are in good agreement with results presented in [19] for a corresponding Newtonian fluid flow. Yapici et. al., [19] used an un-regularised velocity at the moving lid and so did we for the comparison purposes.

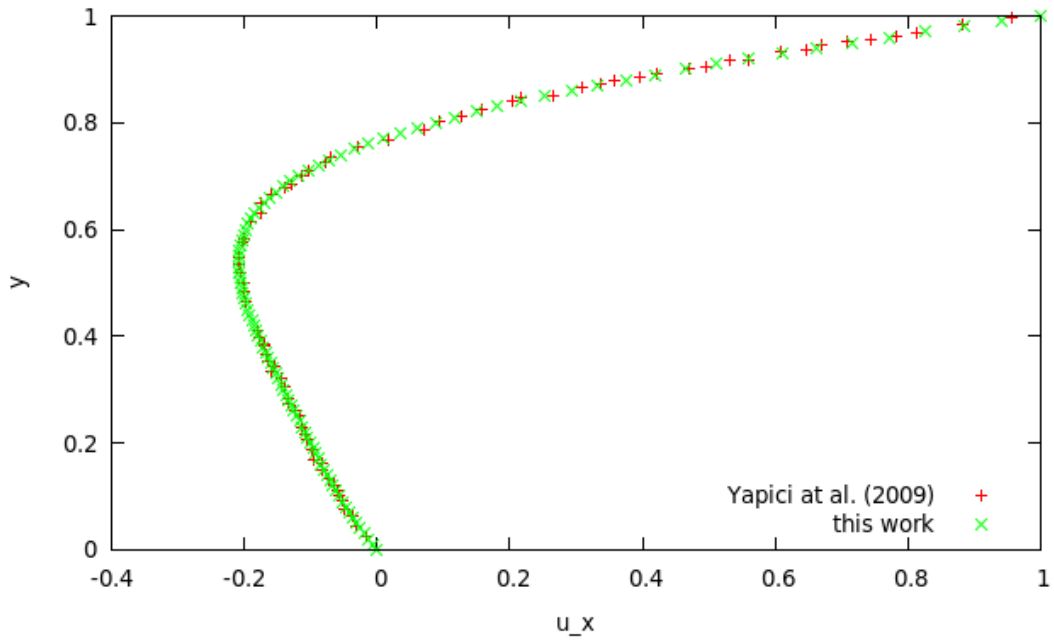


Figure 3.11: velocity component along $x = 0.5$ for Wi $= 0$ and Re $\to 0$ with *viscoelasticFluidFoam solver*.

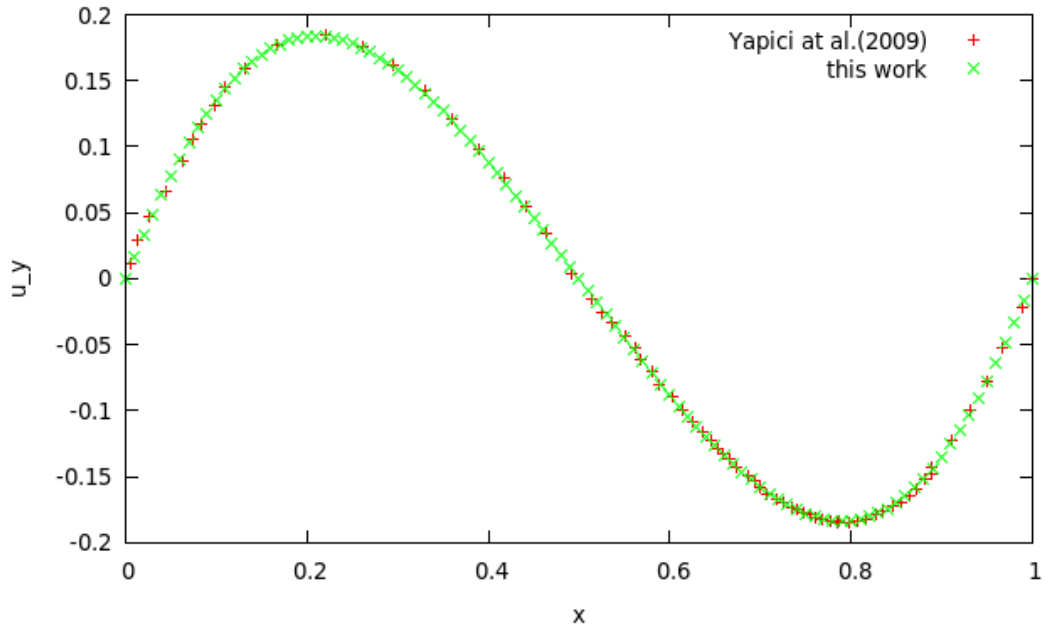
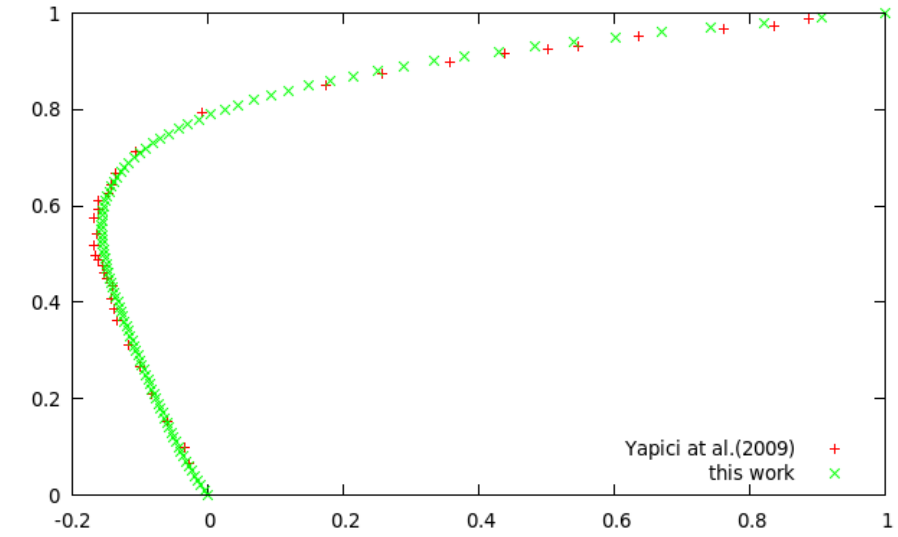


Figure 3.12: velocity component along $y = 0.5$ for Wi $= 0$ and Re $\to 0$ with *viscoelasticFluidFoam* solver.

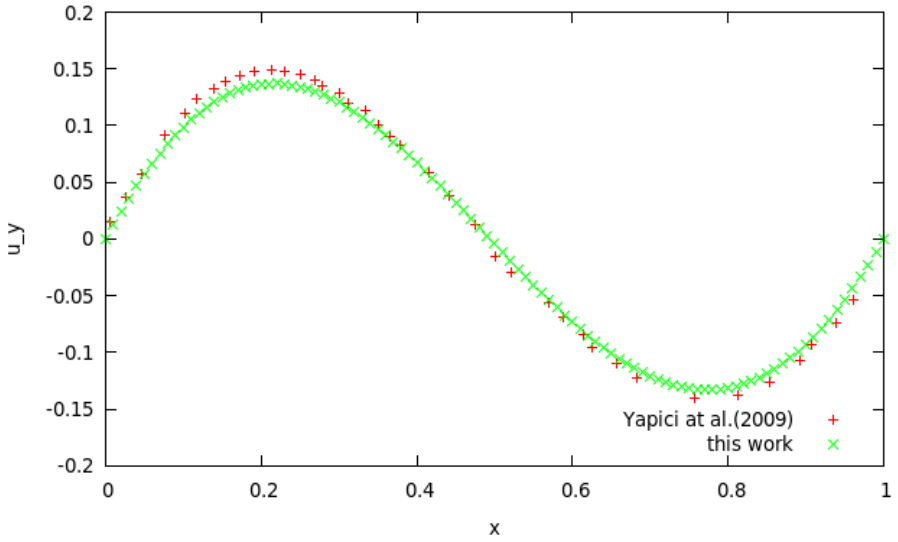
We, however, were not able to obtain convergence up to Wi $= 1$ with the DEVSS method. Indeed, we only get convergence for very small Weissenberg numbers (Wi < 0.05). In [17], convergence was only obtained for Wi < 0.02. It is worth mentioning that a finer mesh is used here with 93025 cells and a

viscosity ratio of $\beta = 0.3$, the same as in [19]. We also employ the SIMPLE algorithm instead of the PISO originally implemented for *visscoelasticFluidFoam*.

Despite the use of the regularised velocity given by Equation (3.2.1), we were still not able to obtain convergence up to Wi $= 1$ using the DEVSS approach. In fact, for Wi ≥ 0.5, a regularised profile fails to deal with values blowing up leading to huge stress gradient. To demonstrate the convergence of our results for Wi < 0.5, Figure 3.13 shows the velocity profiles for Wi $= 0.3$ along the lines $x = 0.5$ and $y = 0.5$ and shows, as expected, that we produce similar results as those in [19].



(a) Velocity at $x = 0.5$

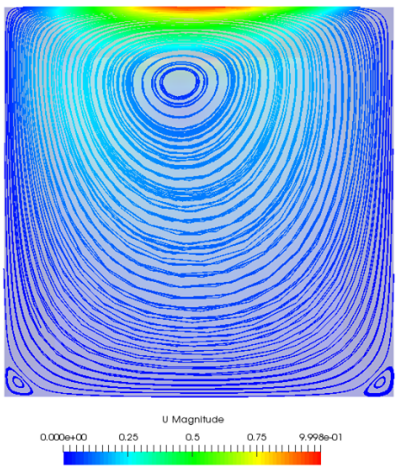


(b) Velocity at $y = 0.5$

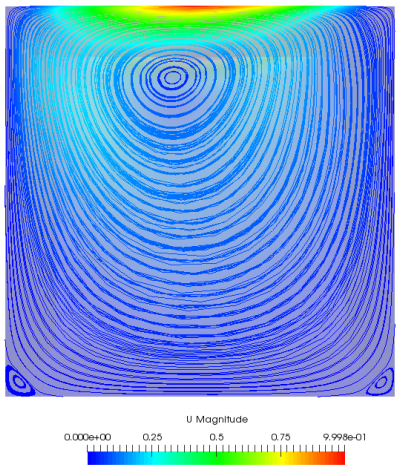
Figure 3.13: Comparison of the velocity profiles for Wi $= 0.3$ along $x = 0.5$ (a) and $y = 0.5$ (b) with results of [19]

Following the analysis in [19], we look at the effects of the Reynolds Number on the flow. Unlike [19] who investigated steady flow, we consider a transient isothermal flow of Oldroyd-B ($\xi = 0$) in the lid-driven cavity geometry.

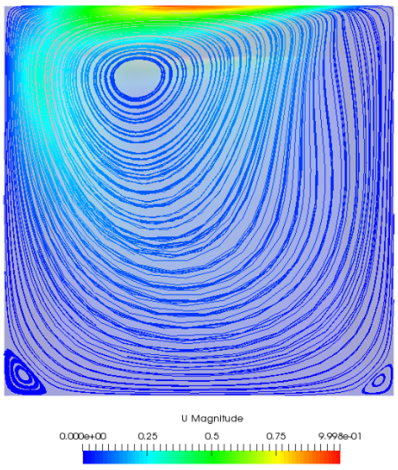
- **Creeping flow.**
  Figures 3.14 and 3.15 show the evolution of streamlines with increasing Wi. We notice that the higher the Wi, the correspondingly bigger the downstream corner vortex becomes, which compares well with the results of [5].



(a) Wi $= 0.7$

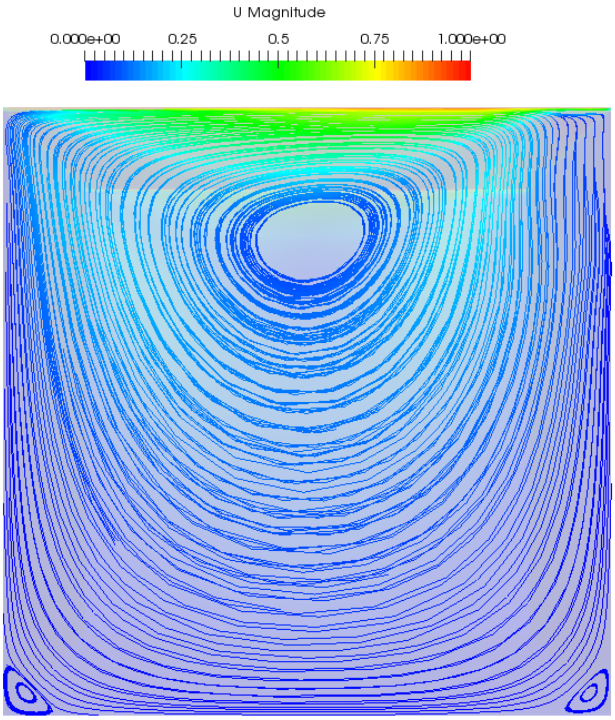


(b) Wi $= 1$

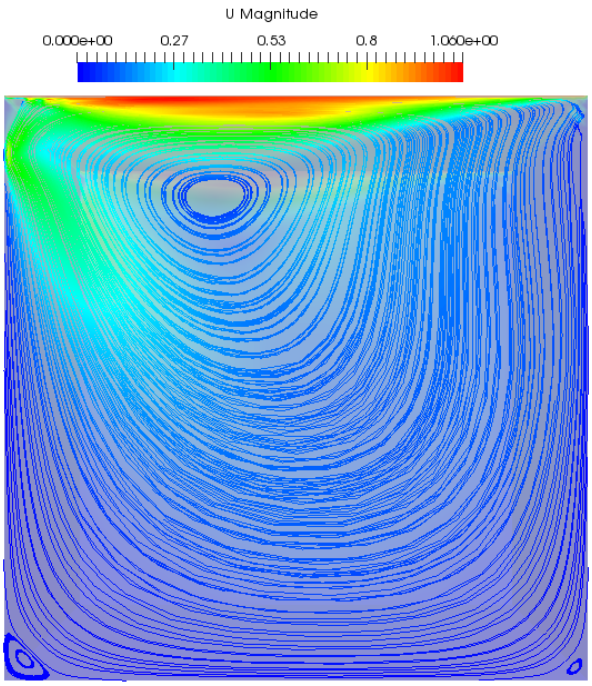


(c) Wi $= 5$

Figure 3.14: Streamlines for $Re = 0.01$ (LCR approach).

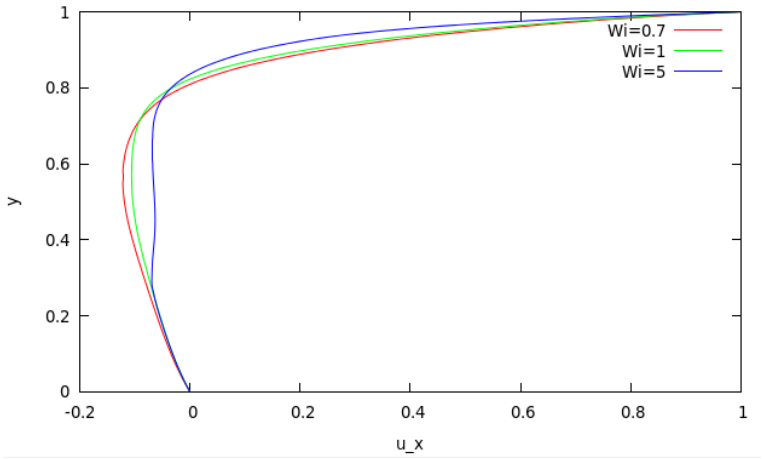


(a) Wi $= 0.3$

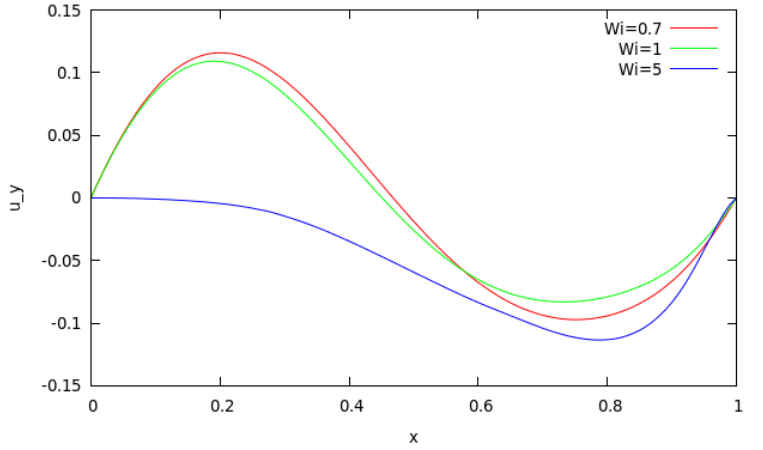


(b) Wi $= 0.7$

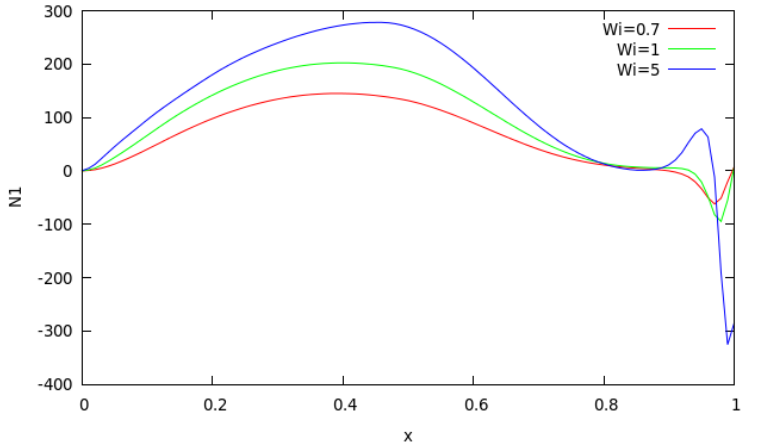
Figure 3.15: Streamlines for $Re = 0.01$ (DEVSS method).



(a) velocity profile at $(0.5, y)$



(b) velocity profile at $(x, 0.5)$



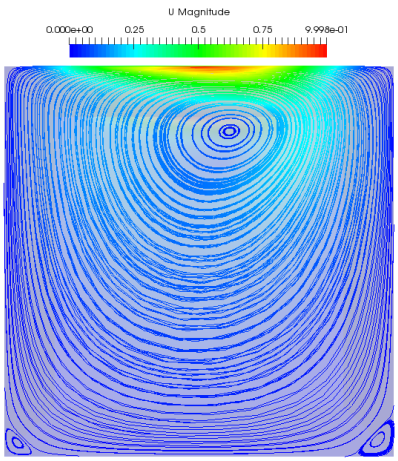
(c) Profile of N1 at $(x, 0, 9989)$

Figure 3.16: Profiles under creeping flow conditions.

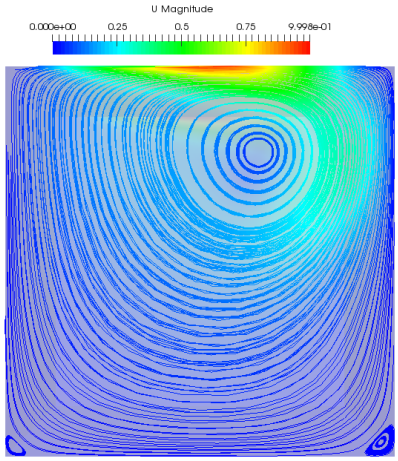
Figure 3.16 shows that as the Weissenberg Number increases, the minimal horizontal velocity also increases. On the contrary, the minimal vertical horizontal decreases. This is similar to the

observations of [19]. As they did not use any method of stabilization, their results for creeping flows are limited up to Wi $= 1$. For our case, when using the DEVSS approach, the critical value of Weissenberg number that can be used is Wi < 0.1. At Wi $= 0.1$, we are not able to get convergence even with very small time-step or relaxation factors even though the simulations are still stable. This is due to the fact that the original *viscoelasticFluidFoam* solver is based on the PISO algorithm which is proven to present convergence issues under creeping flow conditions.

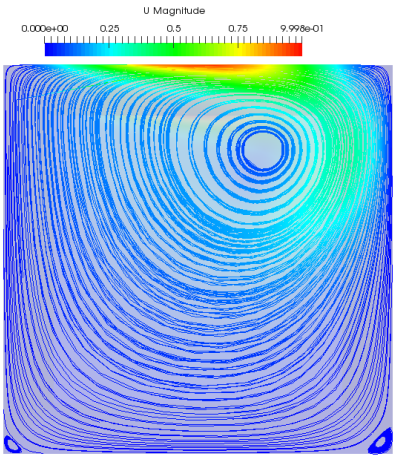
- **$\mathbf{Re = 100}$ case**



(a) Wi $= 0.7$



(b) Wi $= 1$

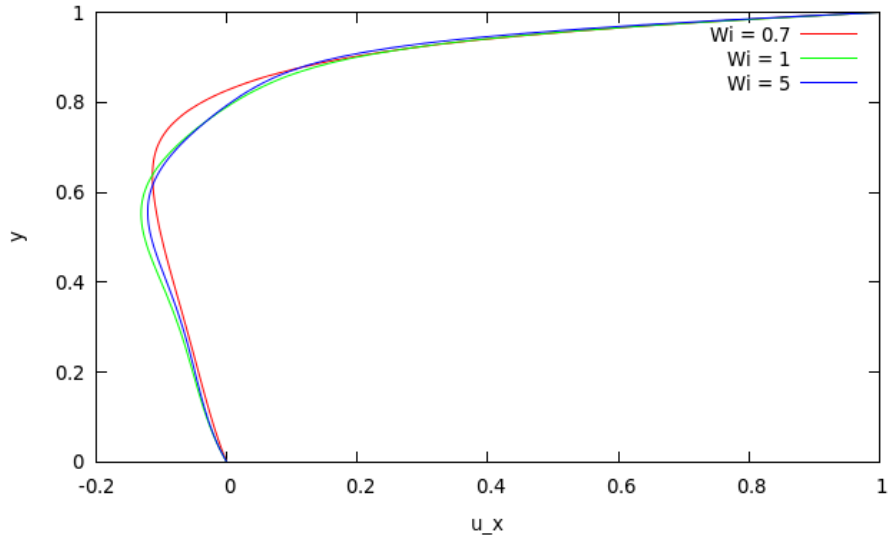


(c) Wi $= 5$

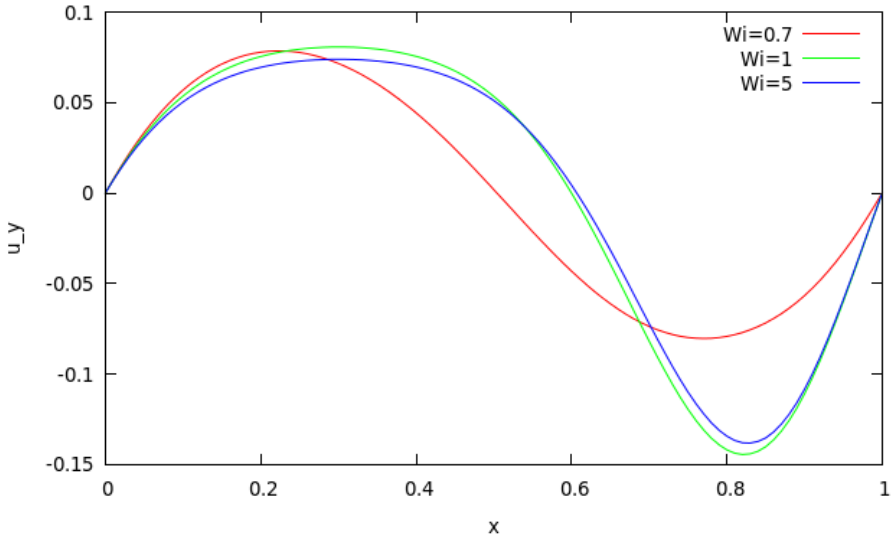
Figure 3.17: Streamlines for $\mathrm{Re} = 100$ (LCR approach)

Compared to the creeping flow case, the size of the upstream corner vortex has increased in size

as well as the primary vortex, see Fig. 3.17. On the other hand, the downstream corner vortex has become smaller. Figure 3.18 shows that increasing the Weissenberg Number for the case $\mathrm{Re} = 100$ has similar effects as for the creeping flow case, Re $= 0.01$, it reduces the size of the downstream vortices. Furthermore, we notice that the primary vortex becomes larger as Wi increases. Contrary to the creeping flow case, the minimal horizontal velocity is shifting downward, closer to the center of the lid. From Wi $= 0.7$ to Wi $= 1$, we can see that the value of the horizontal velocity is decreasing but from Wi $= 1$ to Wi $= 5$, there is a smaller increase. As for the vertical velocity, a decrease of both of its minimal and maximal values is observed with the increase in elasticity. However, the tendency observed in [19] is not conserved for higher Weissenberg number. In particular, their maximum value of Wi attainable for Re $= 100$ is Wi $= 0.7$. With the LCR approach, we can use Wi $= 5$ as predicted by Pimenta and Alves [15] and Fattal and Kupferman [4].



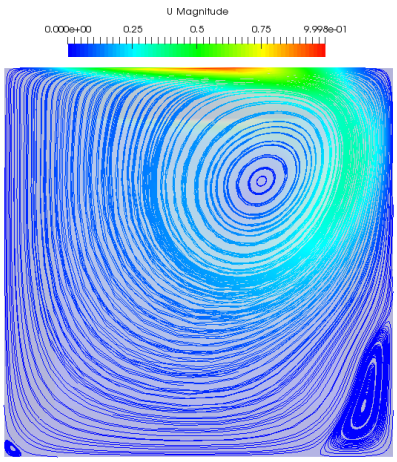
(a) velocity profile at $(0.5, y)$



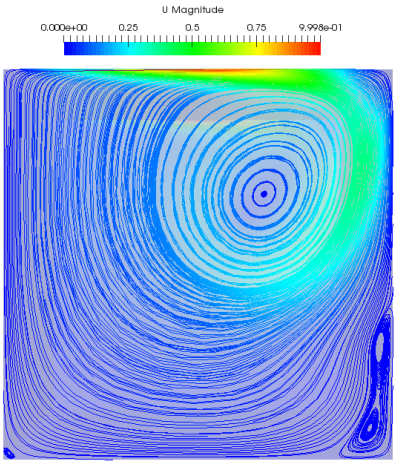
(b) velocity profile at $(x, 0.5)$

Figure 3.18: Profiles for $\mathrm{Re} = 100$.

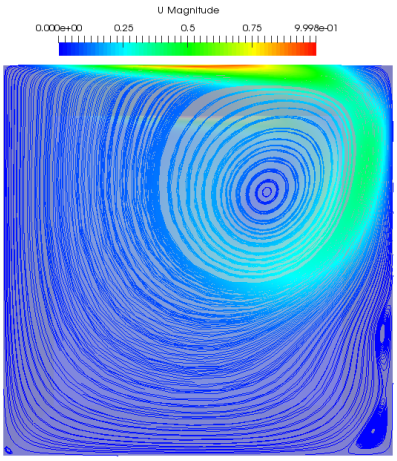
- **$\mathbf{Re = 400}$ case**
  We can observe from Fig. 3.19 that convection certainly impacts on the sizes of both the primary vortex and the upstream corner vortex. Results for $\mathrm{Re} = 400$ show a smaller primary vortex than for the creeping flow or for $\mathrm{Re} = 100$. The right corner vortex, however, significantly increases in size. As before, an increase in Wi makes the right corner vortex looks thinner.



(a) Wi $= 0.4$



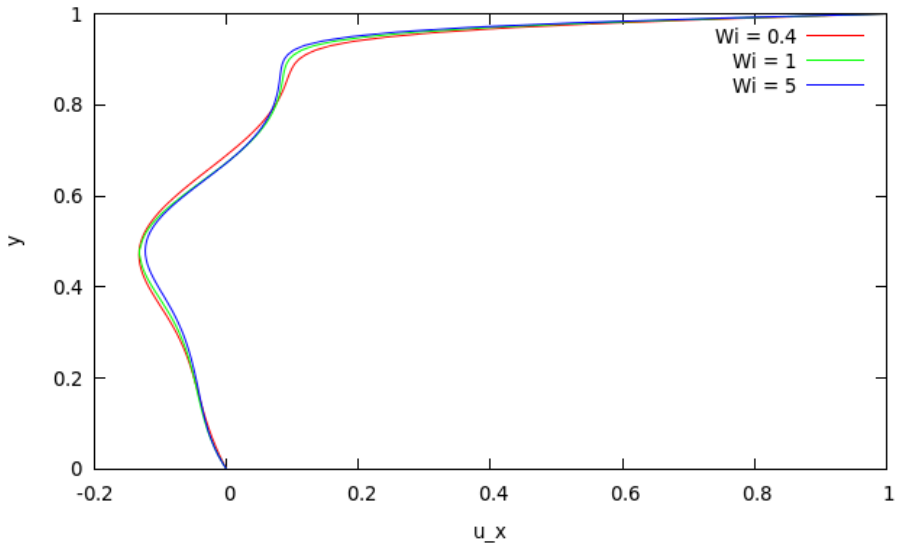
(b) Wi $= 1$



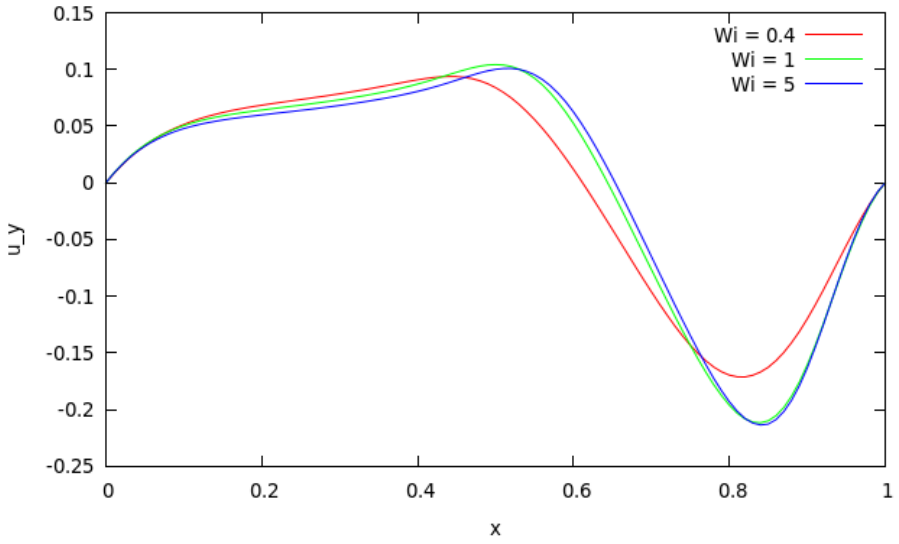
(c) Wi $= 5$

Figure 3.19: Streamlines for $\mathrm{Re} = 400$ (LCR approach)

Figure 3.20 shows that similar behaviour as for $\mathrm{Re} = 100$ is observed for $\mathrm{Re} = 400$ with regards the vertical and the horizontal velocities along $x = 0.5$ and $y = 0.5$ respectively.



(a) velocity profile at $(0.5, y)$

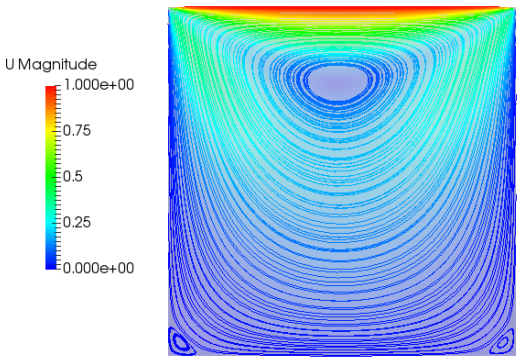


(b) velocity profile at $(x, 0.5)$

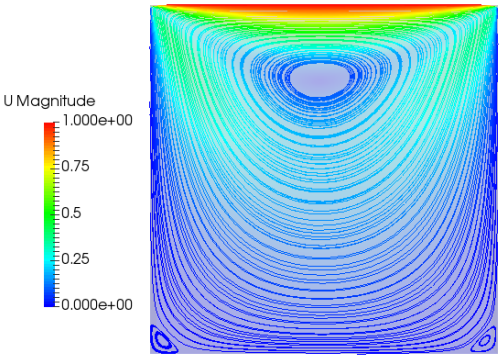
Figure 3.20: velocity profiles for different Wi for $\mathrm{Re} = 400$

Results for a fully Johnson-Segalman model $\xi = 0.8$.

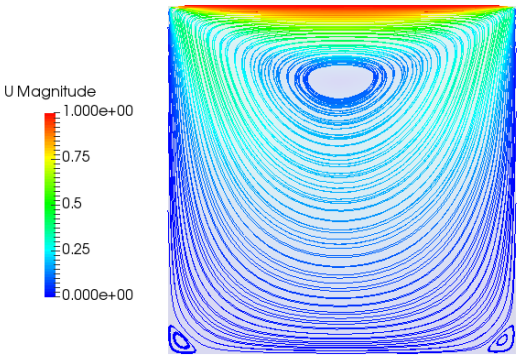
As for the planar contraction flow, we present predictions for the non-zero ξ case. As we can see, under "creeping flow condition ($\mathrm{Re} = 0.01$) we could go up to $\mathrm{Wi} = 1$ which is not the case for the Oldroyd-B. Figure 3.21 shows the evolution of the streamlines with the Weissenberg number while Figure 3.22 gives the velocity profiles along $x = 0.5$ and $y = 0.5$.



(a) Wi $= 0.3$

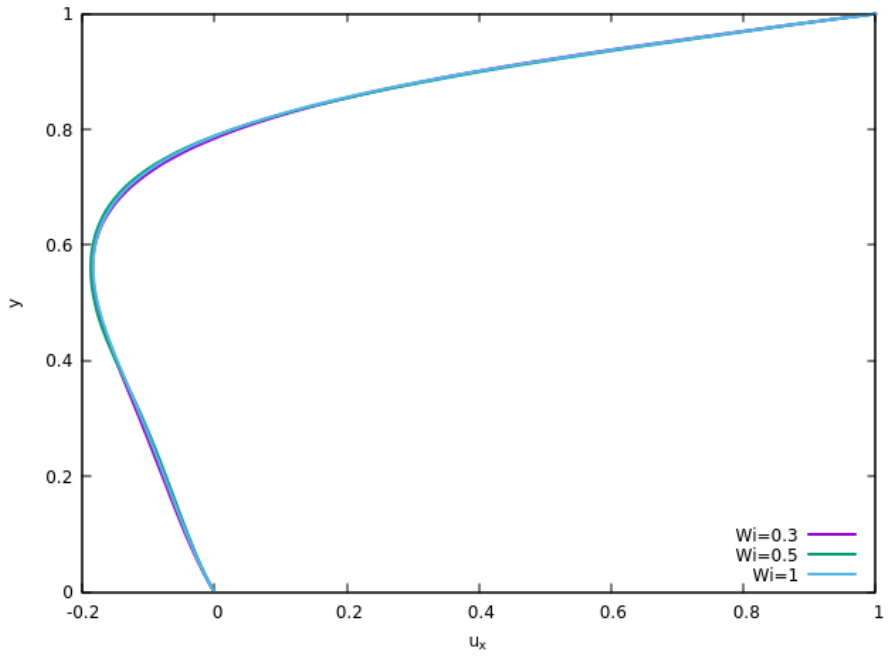


(b) Wi $= 0.5$

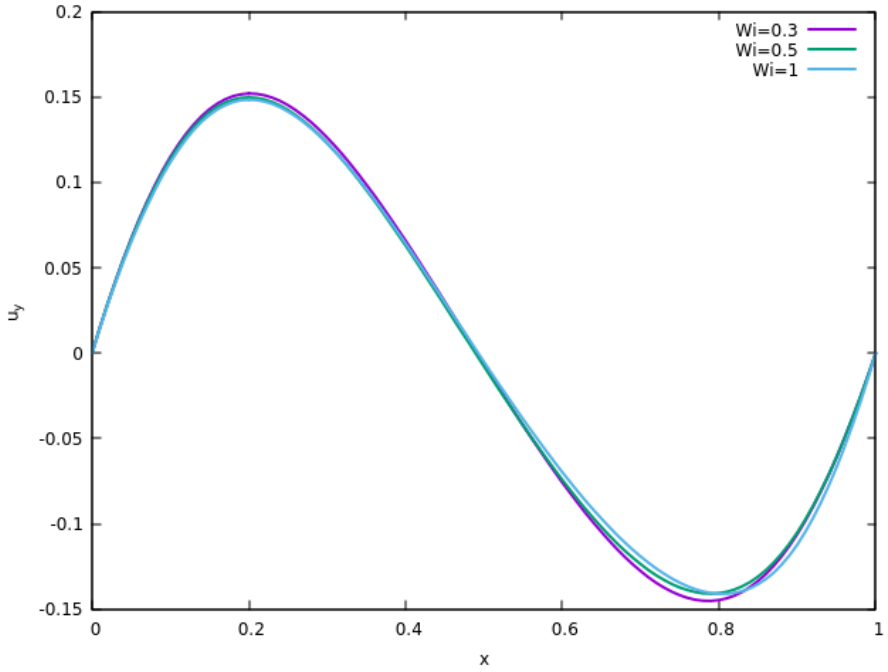


(c) Wi $= 1$

Figure 3.21: Streamlines for $Re = 0.01$ (DEVSS approach).



(a) velocity profile at $(0.5, y)$



(b) velocity profile at $(x, 0.5)$

Figure 3.22: velocity profiles for different Wi for $\mathrm{Re} = 0.01$

4. Shear banding in 1D shear flow

4.1 Flow geometry and conditions

In this section, we consider the flow geometry shown in Figure 4.1.

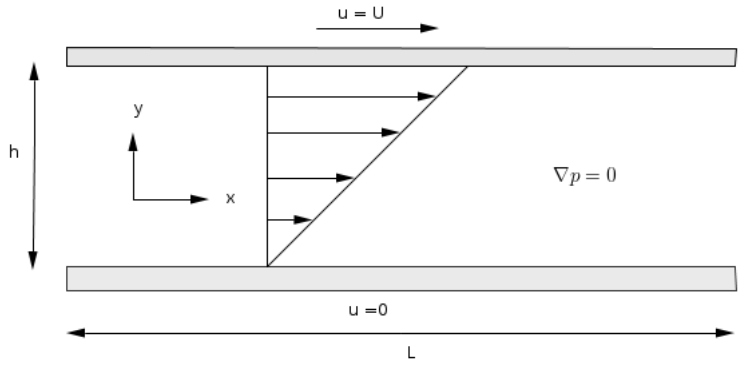


Figure 4.1: Geometry of coeuette flow

A viscoelastic fluid is enclosed between two parallel plates with the top plate moving at a speed U and the bottom plate stationary. If the viscoelastic fluid is described by the Johnson-Segalman constitutive model, then it has been widely demonstrated, including in our own previous work, [16] that shear band would appear in the flow for certain values of material parameters. In our previous work, [16], the finite difference method was employed for the numerical solution and the numerical algorithms were implemented in the Python$^{\mathrm{TM}}$ software. Investigation of both the non-isothermal and isothermal cases were also done in [16]. In the present work, we focus only on the isothermal case as the aim is to assess the viability of our finite volume method (FVM) solver as implemented on the OpenFOAM® software platform.

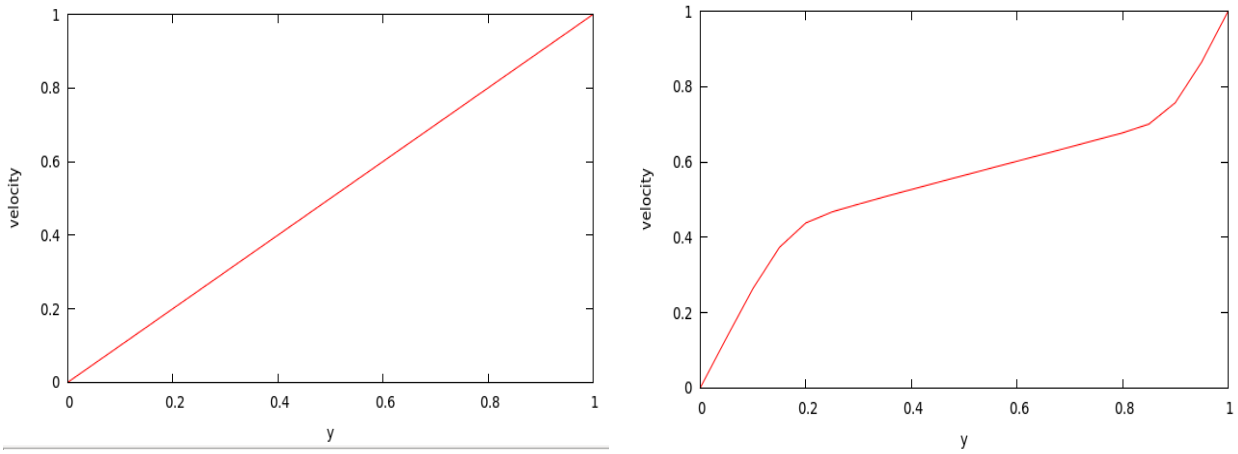
Shear bands are flow instabilities that are characterized by discontinuity of the shear rate. Two main theories have been advanced to explain the origin of shear bands which have been observed in the flow of certain viscoelastic fluids. In particular, shear banding has been theoretically and computationally demonstrated to arise as constitutive instability due to a non-monotonic relationship between shear-rate and shear-stress. An alternative theory has demonstrated that shear banding results from flow inhomogeneities even when the underlying constitutive model has a monotonic relationship between shear-rate and shear-stress. The Johnson-Segalman model is one typical constitutive equation that gives non-monotonic relationships between shear-rate and shear-stress for certain values of the material parameters. Our previous work, [16], indeed demonstrated the shear banding phenomena for homogenous flow of fluids governed by the Johnson-Segalman model.

In the present study, we focus on the FVM method implemented on the OpenFOAM® platform. For the boundary conditions, we impose the zero-gradient condition except for the velocity at the walls for which we impose the no-slip conditions. The computational grid is composed of 400 hexahedral cells.

4.2 Results

Unless otherwise stated, the following parameters values will be used; $Re = 1$, $\Delta t = 0.005$, $\beta = 0.95$, $\Delta x = \Delta y = 0.01$ and $\xi = 0.8$. The following results are based on our Johnson-Segalman solver with the DEVSS stabilization methodology.

4.2.1 Development of shear banding. Figure 4.2 shows that there is no instability (shear banding) observed in the velocity profile for Wi $= 0.47$. On the other hand, shear banding is observed for Wi $= 2$.



Wi $= 0.47$ Wi $= 2$

Figure 4.2: Velocity profiles for the shear flow of a Johnson-Segalman fluid.

Figures 4.3 – 4.7 are presented under the shear banding conditions (Wi $= 2$) and show a comparison of the results from our previous finite difference calculations, [16], with our present finite volume results. There is general qualitative agreement of results. The results, however, also show the deficiency of the DEVSS technique and hence that we would need to investigate the LCR and other stabilization methods of the full Johnson-Segalman algorithm going forward.

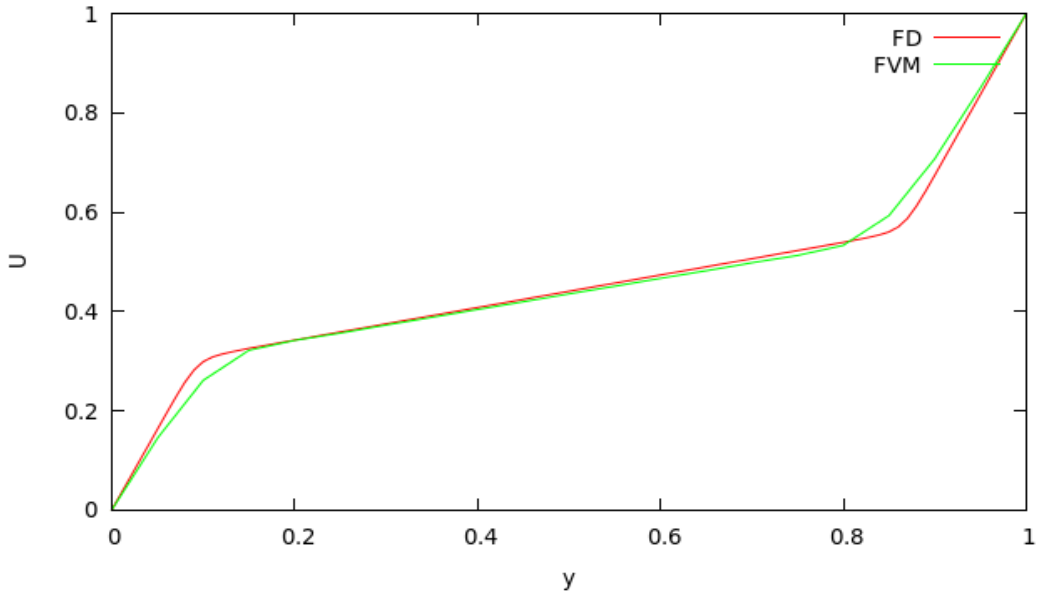


Figure 4.3: Comparison of velocity profiles for the shear flow of a Johnson-Segalman fluid with Wi $= 2$.

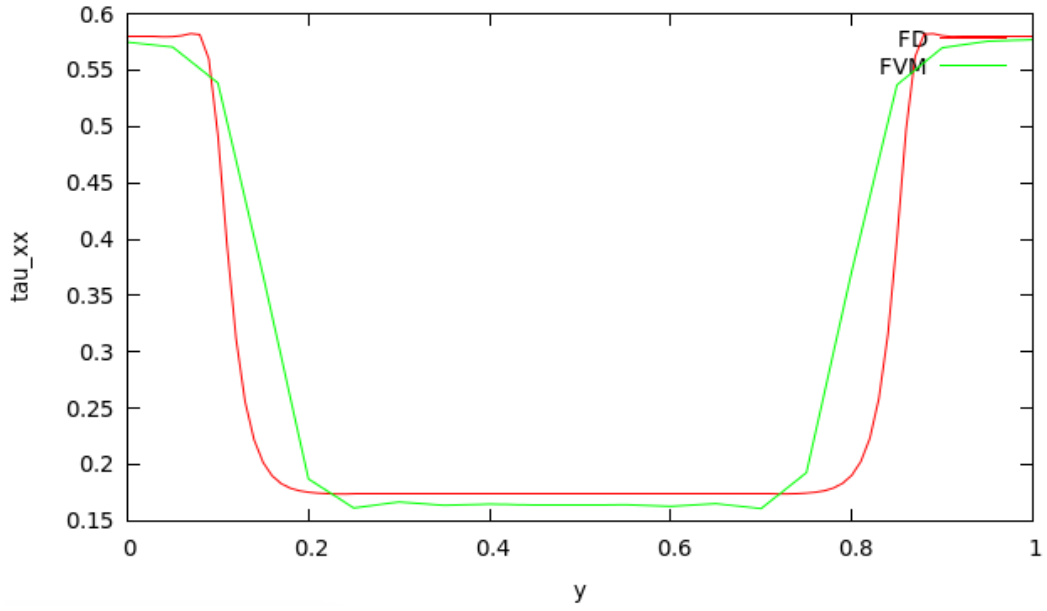


Figure 4.4: Comparison of τ_{xx} stress profiles for the shear flow of a Johnson-Segalman fluid with Wi $= 2$.

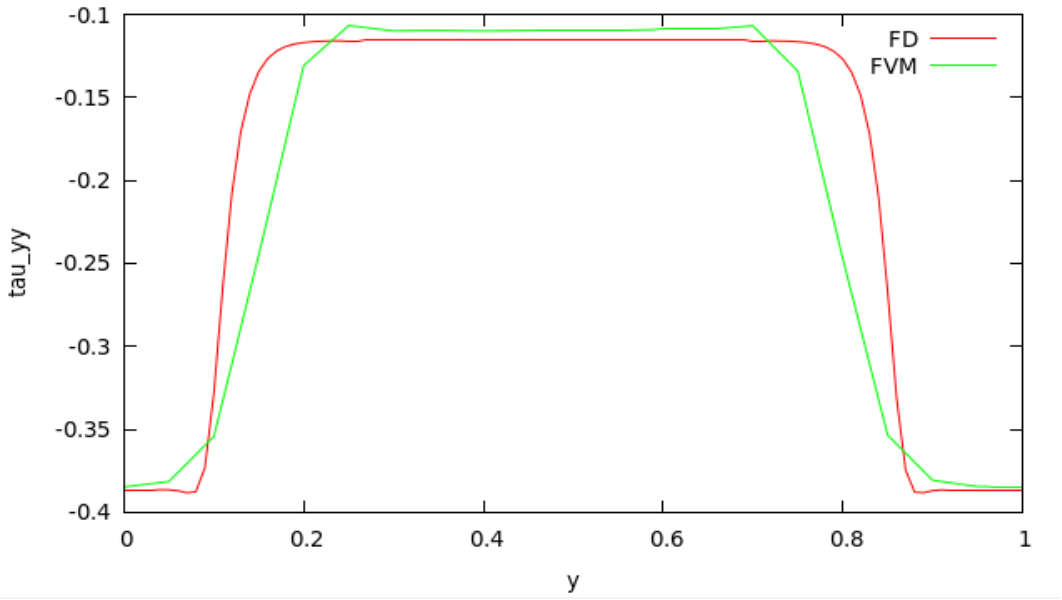


Figure 4.5: Comparison of τ_{yy} stress profiles for the shear flow of a Johnson-Segalman fluid with Wi $= 2$.

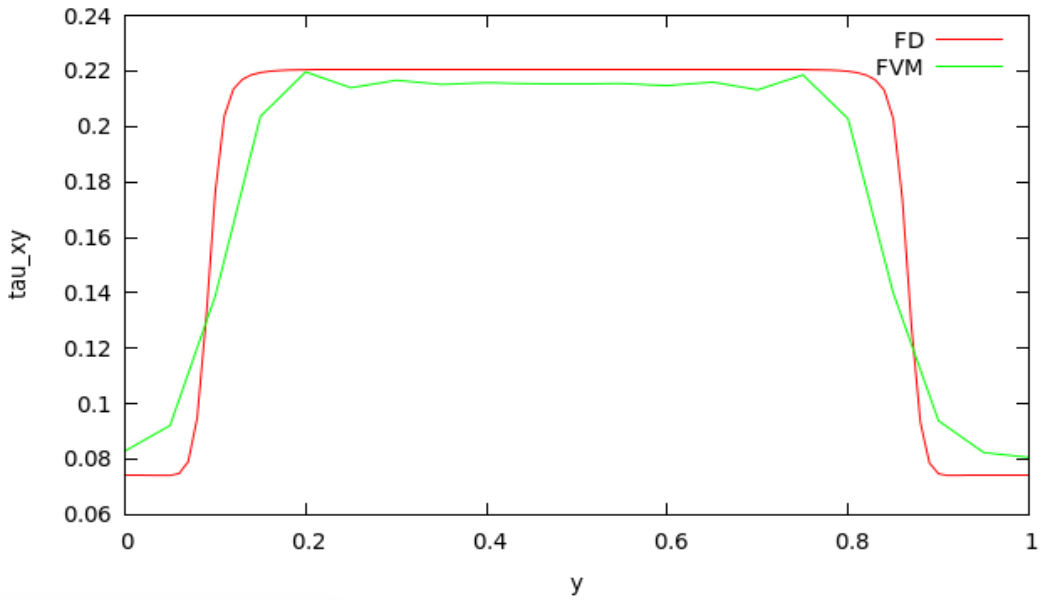


Figure 4.6: Comparison of shear stress profiles, τ_{xy}, for the shear flow of a Johnson-Segalman fluid with Wi $= 2$.

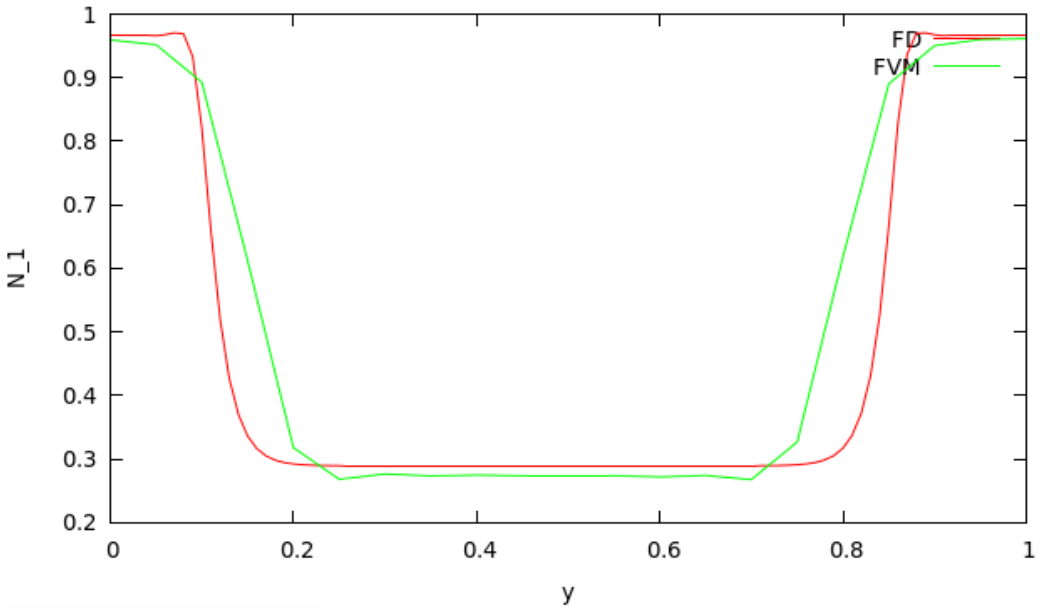


Figure 4.7: Comparison of first normal stress difference, N_1, for the shear flow of a Johnson-Segalman fluid with Wi $= 2$.

As would be expected, our solver demonstrates no shear banding instabilities for the Oldroyd-B model, $\xi = 0$, see Figures 4.8 – 4.12. Additionally, we notice that the LCR technique offers better stabilization than the DEVSS approach. Once we have comprehensively implemented and validated the LCR technique for the full Johnson-Segalman model, the shear banding results would need to be revisited. Such stabilization studies indeed forms the basis of our ongoing and future work.

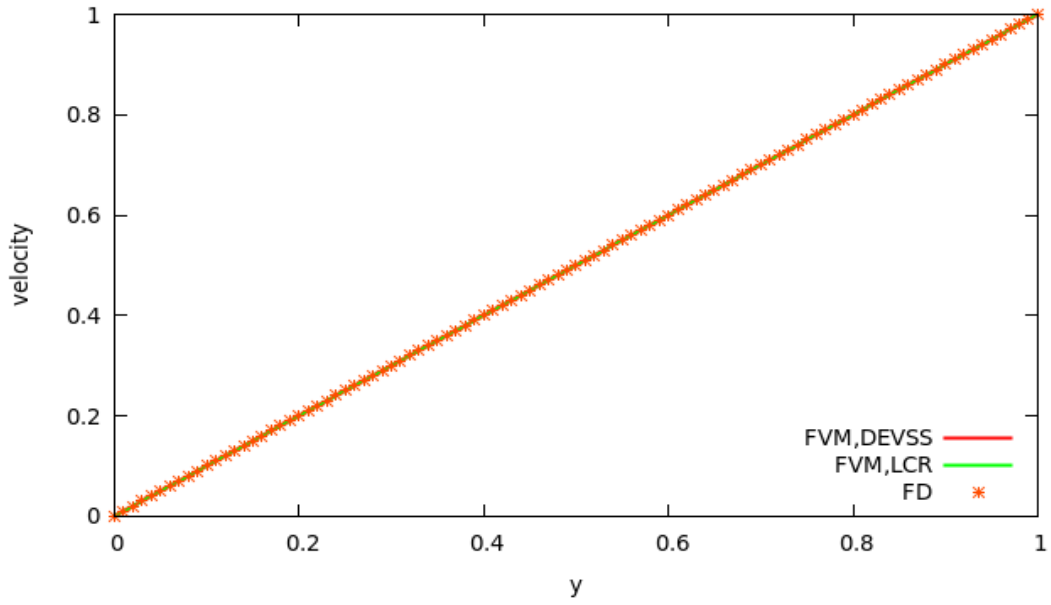


Figure 4.8: Comparison of velocity profiles for the shear flow of an Oldroyd-B fluid with Wi $= 2$.

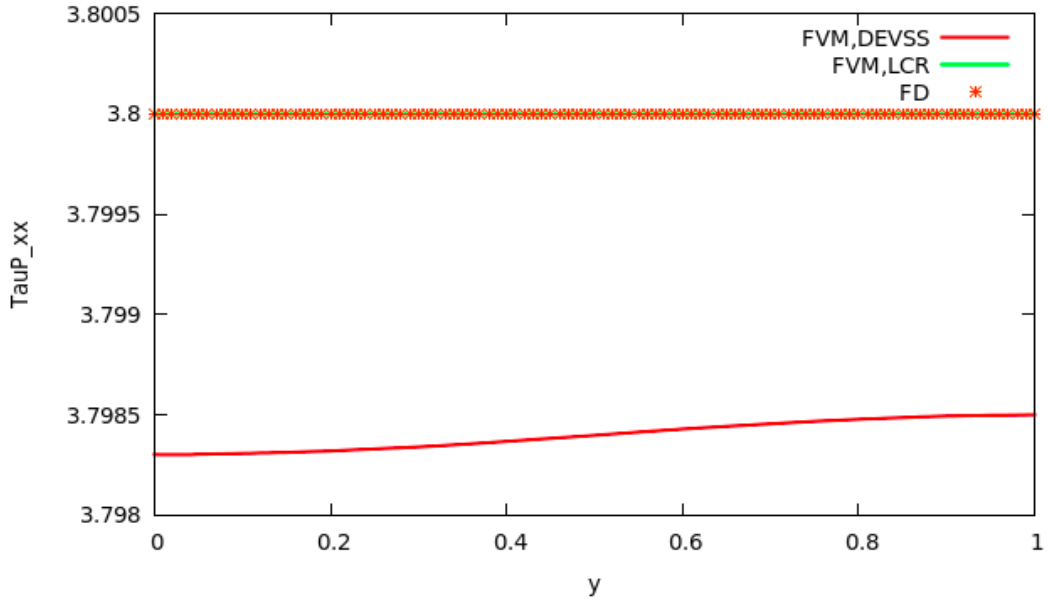


Figure 4.9: Comparison of τ_{xx} stress profiles for the shear flow of an Oldroyd-B fluid with Wi $= 2$.

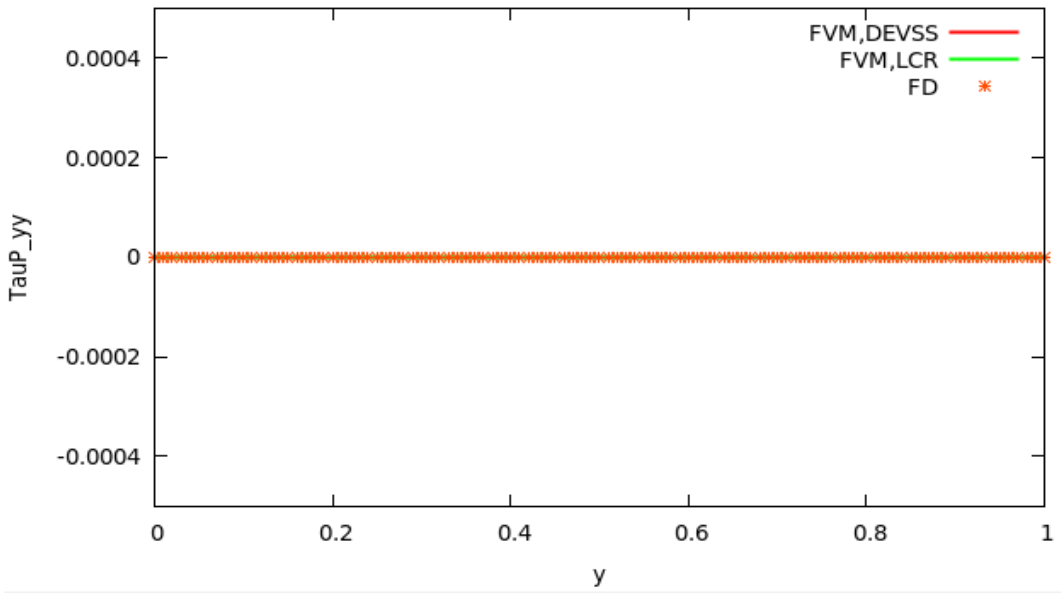


Figure 4.10: Comparison of τ_{yy} stress profiles for the shear flow of an Oldroyd-B fluid with Wi $= 2$.

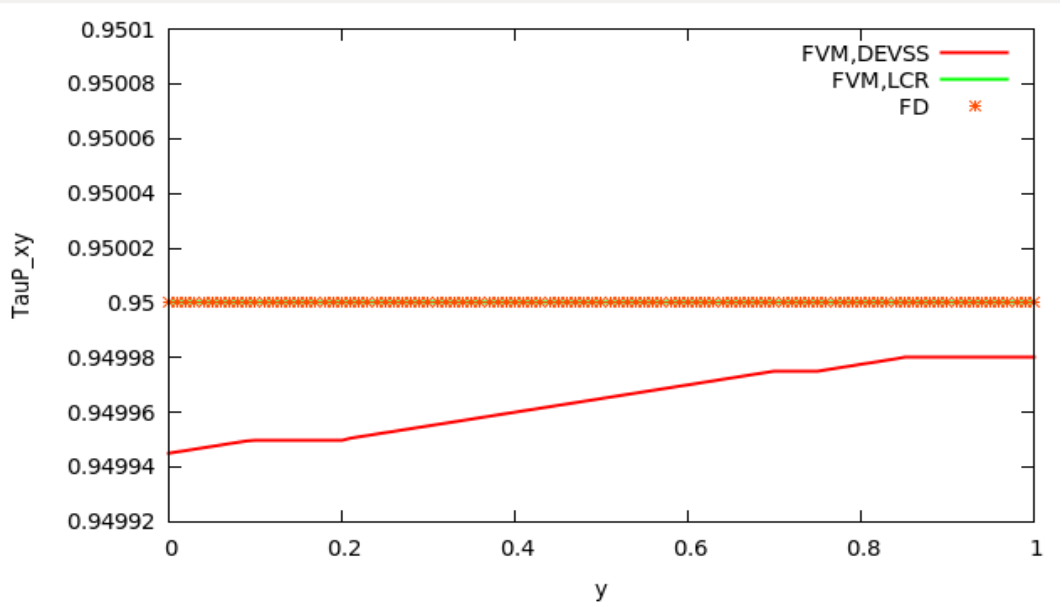


Figure 4.11: Comparison of shear stress profiles, τ_{xy}, for the shear flow of an Oldroyd-B fluid with $\mathrm{Wi} = 2$.

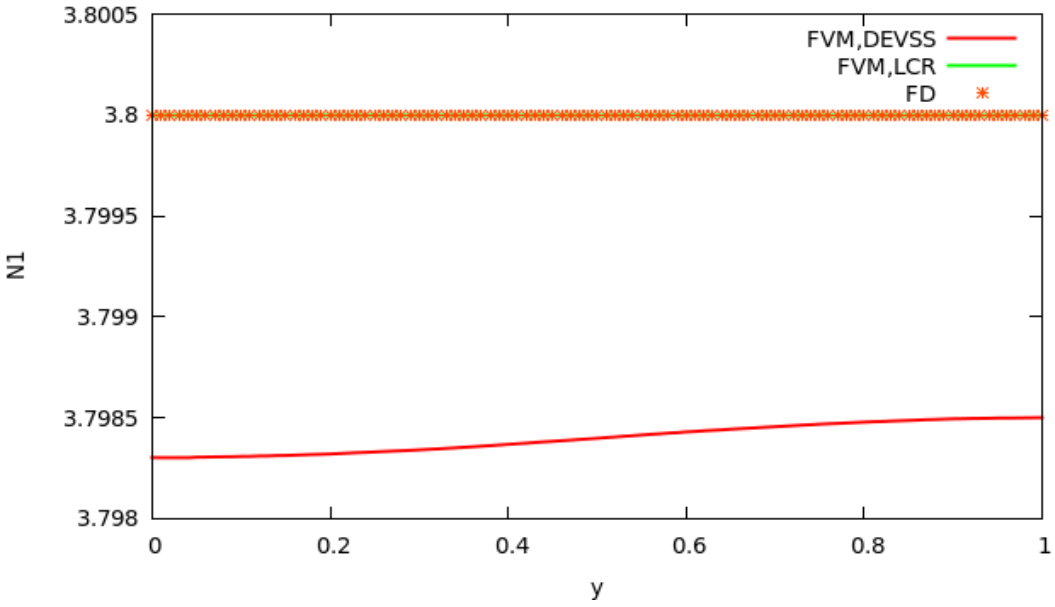
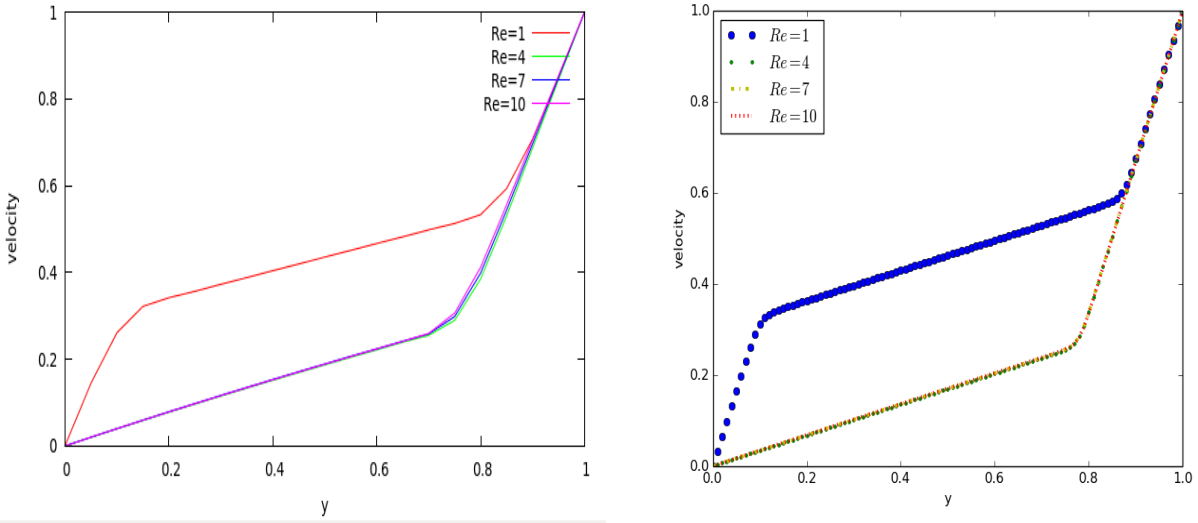


Figure 4.12: Comparison of first normal stress difference, N_1, for the shear flow of an Oldroyd-B fluid with $\mathrm{Wi} = 2$.

4.2.2 Parameter influence on the solutions. Figure 4.13 shows how the velocity profile responds to an increase in the Reynolds number. Both the FDM and FVM give similar results. For higher Reynolds number, the absence of shear bands near the stationary plate is observed. The Reynolds number measures the balance of convection and diffusion, in other words the balance of inertia forces and viscous forces. Convection dominated flows have correspondingly higher Reynolds numbers.



Finite volume method

Finite difference method

Figure 4.13: Velocity profiles for the shear flow of a Johnson-Segalman fluid with $\xi = 0.8$ and Wi $= 2$.

5. Conclusion

We have implemented a numerical solver on the open source software platform OpenFOAM® (Foam-Extend). The solver is based on the finite volume method and is used to study the flow of viscoelastic fluids which are governed by the Johson-Segalman constitutive model. Validation of solver is done via the two benchmark problems, namely, the 4:1 planar contraction flow and the lid-driven cavity flow. Due to their prevalence in the published literature, we have used Oldroyd-B results for such validation purposes, the Oldroyd-B model being obtained by taking $xi = 0$ in our Johnson-Segalman model.

Two stabilization approaches were also explored, the Discreet Elastic Viscous Stress Split (DEVSS) and the Log-Conformation Reformulation (LCR) methods. In order to take advantage of similar work that has already been donce by others on the OpenFOAM® software platform, the LCR technique was implemented in the *rheoFoam* solver whilst the DEVSS technique was implemented in the *viscoelasticFluidFoam* solver. Even though we are fully confident with, and have indeed clearly validated, our Oldroyd-B results using either stabilization method, the implementation of the LCR technique for the Johnson-Segalman model still requires further work and is currently the focus of our ongoing investigations. Indeed, we have already laid out the theoretical framework for the implementation of an LCR technique within the Johnson-Segalman model framework. The computational implementation is however far from trivial and hence remains a focus of our ongoing research efforts.

The above, notwithstanding, our present results, say for the Oldroyd-B case, clearly demonstrate the superiority of the LCR stabilization technique. Indeed, our previous results via the finite difference method (FDM) for the shear flow of Johnson-Segalman fluids shows that the FDM performs better than the finite volume methods (FVM) with a DEVSS stabilization. Similar results with Oldroyd-B fluids further confirm that the superiority of the LCR technique over the DEVSS method in comparison with the FDM. Even though the FDM seems to perform relatively well for the 1D shear flows, the FVM remain more desirable as they are better suited to complex geometries and unstructured meshes etc. It is for this reason therefore that comprehensive investigations of the stabilization of the FVM, say via the LCR technique, are sustained.

We have also studied the effect of convection (high Reynolds number) and elasticity (as measured by the Wissenberg number) on the viscoelastic fluid flow. Our results in this direction will also assist in cross-validation investigations by other researchers.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisor Dr Tiri Chinyoka for his constant support and availability and for his patient guidance. Throughout the duration of my master studies, he not only provided good material conditions but also offered constructive suggestions and assistance which were important for the accomplishment of this work.

I would also like to thank the Center for High Performance Computing (CHPC) in Cape Town for allowing us to use their high performance computing facilities and I would especially like to thank Dr Andrew Gill from the CHPC for his availability and assistance.

I acknowledge the financial support from AIMS South Africa and the University of Cape Town.

I extend warm thank to Jade Abuga for her precious and kind assistance and also to all people in the Mathematics Department at UCT.

Special thanks to my family and friends for their constant prayers and their great support. Last but not least, I am deeply grateful to the Almighty God without Who this journey would not even exist.

References

[1] A. Afonso, M. Oliveira, P. Oliveira, M. Alves, and F. Pinho. The finite volume method in computational rheology. In *Finite Volume Method-Powerful Means of Engineering Design (pp. Ch-7)*. InTech, 2012.

[2] M. Alves, P. Oliveira, and F. Pinho. A convergent and universally bounded interpolation scheme for the treatment of advection. *International Journal for Numerical Methods in Fluids*, 41(1):47–75, 2003.

[3] X. Chen, H. Marschall, M. Schäfer, and D. Bothe. A comparison of stabilisation approaches for finite-volume simulation of viscoelastic fluid flow. *International Journal of Computational Fluid Dynamics*, 27(6-7):229–250, 2013.

[4] R. Fattal and R. Kupferman. Constitutive laws for the matrix-logarithm of the conformation tensor. *Journal of Non-Newtonian Fluid Mechanics*, 123(2):281–285, 2004.

[5] R. Fattal and R. Kupferman. Time-dependent simulation of viscoelastic flows at high weissenberg number using the log-conformation representation. *Journal of Non-Newtonian Fluid Mechanics*, 126(1):23–37, 2005.

[6] J. L. Favero, A. R. Secchi, N. S. M. Cardozo, and H. Jasak. Viscoelastic flow simulation: development of a methodology of analysis using the software openfoam and differential constitutive equations. *Computer Aided Chemical Engineering*, 27:915–920, 2009.

[7] F. Habla, A. Woitalka, S. Neuner, and O. Hinrichsen. Development of a methodology for numerical simulation of non-isothermal viscoelastic fluid flows with application to axisymmetric 4: 1 contraction flows. *Chemical Engineering Journal*, 207:772–784, 2012.

[8] L. Holmes, J. Favero, and T. Osswald. Numerical simulation of three-dimensional viscoelastic planar contraction flow using the software openfoam. *Computers & Chemical Engineering*, 37: 64–73, 2012.

[9] R. I. Issa. Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of Computational Physics*, 62(1):40–65, 1986.

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

[11] F. Moukalled, L. Mangani, M. Darwish, et al. *The Finite Volume Method in Computational Fluid Dynamics*. Springer, 2016.

[12] C. Open. Openfoam user guide. *OpenFOAM Foundation*, 2(1), 2011.

[13] T.-W. Pan and J. Hao. Numerical simulation of a lid-driven cavity viscoelastic flow at high weissenberg numbers. *Comptes Rendus Mathematique*, 344(4):283–286, 2007.

[14] S. Patankar. *Numerical Heat Transfer and Fluid Flow*. CRC press, 1980.

[15] F. Pimenta and M. Alves. Stabilization of an open-source finite-volume solver for viscoelastic fluid flows. *Journal of Non-Newtonian Fluid Mechanics*, 239:85–104, 2017.

[16] F. Rahantamialisoa. Constitutive analysis of shear banding in homogeneous 1D simple shear flow of viscoelastic liquids. Master's thesis, AIMS, South Africa, 2015.

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

[18] H. K. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Pearson Education, 2007.

[19] K. Yapici, B. Karasozen, and Y. Uludag. Finite volume simulation of viscoelastic laminar flow in a lid-driven cavity. *Journal of Non-Newtonian Fluid Mechanics*, 164(1):51–65, 2009.