The Tensor Diffusion approach as a novel technique for simulating viscoelastic fluid flows

Dissertation

zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

Der Fakultät für Mathematik der

Technischen Universität Dortmund

vorgelegt von

Patrick Westervoß

im Januar 2021

Dissertation

The Tensor Diffusion approach as a novel technique for simulating viscoelastic fluid flows

Fakultät für Mathematik
Technische Universität Dortmund

Erstgutachter: Prof. Dr. Stefan Turek

Zweitgutachter: Prof. Dr. Christian Kreuzer

Tag der mündlichen Prüfung: 12.03.2021

“Jeder scheitert halt so gut er kann.”
aus *Sensationell* von Kapelle Petra

Acknowledgements

First of all, I would like to thank my supervisor Prof. Dr. Stefan Turek for giving me the opportunity to work in this very interesting field of research. In particular, I am very thankful for his constant support and input, which helped a lot to push my work to the current state.

I would also like to thank all members of the LS3 for creating such a nice working atmosphere and helping out in any case. Special thanks go out to Dr. Hogenrich Damanik, Dr. Abderrahim Ouazzi and Dr. Christoph Lohmann, who significantly contributed to the development of the proposed numerical approach as well as this thesis itself.

Furthermore, the financial support from ARLANXEO Deutschland GmbH regarding the research project "Flow simulation of polymer melts by means of integral constitutive equations", which provides the basis of this thesis, is gratefully acknowledged.

Most of all, I want to thank my family for always supporting me and believing in me.

Dortmund, January 2021

Patrick Westervoß

Abstract

In this thesis, the novel Tensor Diffusion approach for the numerical simulation of viscoelastic fluid flows is introduced. Therefore, it is assumed that the extra-stress tensor can be decomposed into a product of the strain-rate tensor and a (nonsymmetric) tensor-valued viscosity function. As a main potential advantage, which can be demonstrated for fully developed channel flows, the underlying complex material behaviour can be explicitly described by means of the so-called Diffusion Tensor. Consequently, this approach offers the possibility to reduce the complete nonlinear viscoelastic three-field model to a generalised Stokes-like problem regarding the velocity and pressure fields, only. This is enabled by inserting the Diffusion Tensor into the momentum equation of the flow model, while the extra-stress tensor or constitutive equation can be neglected. As a result, flow simulations of viscoelastic fluids could be performed by applying techniques particularly designed for solving the (Navier-)Stokes equations, which leads to a way more robust and efficient numerical approach. But, a conceptually improved behaviour of the numerical scheme concerning viscoelastic fluid flow simulations may be exploited with respect to discretisation and solution techniques of typical three- or four-field formulations as well. In detail, an (artificial) diffusive operator, which is closely related to the nature of the underlying material behaviour, is inserted into the (discrete) problem by means of the Diffusion Tensor. In this way, certain issues particularly regarding the flow simulation of viscoelastic fluids without a Newtonian viscosity contribution, possibly including realistic material and model parameters, can be resolved.

In a first step, the potential benefits of the Tensor Diffusion approach are illustrated in the framework of channel flow configurations, where several linear and nonlinear material models are considered for characterising the viscoelastic material behaviour. In doing so, typical viscoelastic flow phenomena can be obtained by simply solving a symmetrised Tensor Stokes problem including a suitable choice of the Diffusion Tensor arising from both, differential as well as integral constitutive laws. The validation of the novel approach is complemented by simulating the Flow around cylinder benchmark by means of a four-field formulation of the Tensor Stokes problem. In this context, corresponding reference results are reproduced quite well, despite the applied lower-order approximation of the tensor-valued viscosity. A further evaluation of the Tensor Diffusion approach is performed regarding two-dimensional contraction flows, where potential advantages as well as improvements and certain limits of this novel approach are detected. Therefore, suitable stabilisation techniques concerning the Diffusion Tensor variable plus the behaviour of deduced monolithic and segregated solution methods are investigated.

Contents

Chapter 1

Introduction

Even after several decades of research, the numerical simulation of viscoelastic fluid flows is still a challenging task, especially due to the complex material behaviour of such fluids characterised by a differential or integral constitutive equation. Moreover, the involved difficulties are amplified in case of simulating applications from the rubber industry, where specific types of viscoelastic materials are processed, in detail pure polymer melts. Consequently, numerical challenges do not arise only due to the complexity of the considered geometry or the underlying process itself. In addition, the computational difficulties are enhanced by the mathematical properties of the rheological model at hand together with the (realistic) material and model parameters applied therein, which is outlined in the following.

Typically, such material and model parameters are identified by approximating linear and nonlinear material data resulting from experimental measurements, such as storage and loss moduli or elongational and shear viscosities. To establish an accurate prediction of this (non)linear material behaviour, a so-called multi-mode approach needs to be applied in terms of the rheological modelling. Therefore, the continuous relaxation time spectrum, which might reach over several decades as in case of polymer melts, is approximated by means of a sufficiently large number of reasonably spreaded discrete relaxation times or modes. By doing so, an appropriate fitting of the experimental data is obtained by a superposition of approximations resulting for each mode. To maintain the approximation quality of the viscoelastic material behaviour in corresponding numerical flow simulations, multiple modes need to be considered in the flow model as well, which is based on the (Navier-)Stokes equations coupled with constitutive equations of differential or integral type. Regarding differential material laws, the described multi-mode approach probably results in a huge numerical effort, since for each mode a separate stress tensor needs to be regarded as independent flow variable leading to an increasing problem size according to the number of modes. Thus, this procedure is not practical in the context of reasonably simulating polymer melt flows, since usually a large number of modes is required to guarantee an adequate material description. In this regard, integral constitutive equations offer a suitable alternative at least from a numerical point of view, where the multiple modes do not explicitly lead to an increasing computational effort. Additionally, a corresponding modelling approach is preferred from a rheological perspective as well, as the "memory" of the material is taken into account in a much more suitable manner than in case of differential models.

A second issue resulting from the broad relaxation time spectrum of realistic materials is the well-known High Weissenberg Number Problem, which occurs in the context of numerical viscoelastic fluid flow simulations. As an example, the calculations might break down already at quite low relaxation times due to the lack of accurately approximating the exponentially growing stress profiles by means of polynomial functions, which are typically applied in Finite Element discretisations. Similarly, high relaxation times increase the complexity of the (discrete) problem, as the hyperbolic character of differential constitutive equations – commonly including the upper-convected time derivative – is intensified. But also concerning integral material laws, large relaxation times cause numerical difficulties because of an increasing computational effort. In fact, the resulting memory of the fluid reaches over a larger time frame, which accordingly needs to be taken into account in the numerical evaluation of the stress integral. Hence, suitable strategies are required for dealing with the higher relaxation times or Weissenberg numbers involved in simulating (realistic) viscoelastic fluid flows.

Moreover, considering pure polymer melts in the differential or integral viscoelastic flow model leads to further challenges, as such fluids do not consist of a solvent contribution to the viscosity. Thus, the corresponding diffusive operator vanishes from the momentum equation of the (Navier-)Stokes subproblem, which causes several difficulties regarding discretisation and solution techniques of the underlying three-field formulation. On the one hand, an additional stability condition regarding the choice of the approximating spaces with respect to the velocity and stress variables needs to be taken into account besides the usual velocity-pressure coupling. On the other hand, restrictions concerning the applicability of common solution methods are observed, in principle making multigrid solvers in terms of monolithic approaches unpractical or preventing a reasonable implementation of segregated schemes.

Contribution of the thesis

In this thesis, a novel approach to simulate viscoelastic fluid flows is proposed, which is intended to resolve at least some of the issues mentioned above. The underlying concept of this approach is to decompose the extra-stress tensor, which characterises the complex rheological material behaviour of the fluid, into a product of the strain-rate tensor and the so-called Diffusion Tensor, that is a (non-symmetric) tensor-valued viscosity. By inserting the stress decomposition into the momentum equation of the flow model, several potential benefits arise in the numerical approach to simulate viscoelastic fluid flows. This is illustrated below by taking into account various versions of the so-called Tensor Stokes problem with respect to both, differential as well as integral constitutive equations. In the ideal case, the pure Tensor Stokes problem describes the original viscoelastic flow solution by means of a generalised non-Newtonian Stokes-like problem, but with a nonlinear tensor-valued viscosity function. Thereby, the Diffusion Tensor would be given explicitly depending on the velocity field or its gradient, which is why the velocity and pressure fields are the only unknowns in this problem. Alternatively, the original differential or integral viscoelastic flow model is supplemented with an additional (algebraic) equation regarding the determination of the Diffusion Tensor based on the decomposition of the extra-stress. Hence, the four-field formulation of the Tensor Stokes problem is obtained, where a coupling of the

Diffusion Tensor and the original flow variables is enabled by means of the (symmetrised) stress decomposition applied in the momentum equation of the Stokes subproblem. Finally, a three-field formulation of the Tensor Stokes problem can be derived in the context of differential models by inserting the stress decomposition also into the constitutive equation, which results in a partial differential equation for calculating the Diffusion Tensor. At the same time, the extra-stress tensor is removed from the flow model, similar to the pure Tensor Stokes problem.

The potential benefits of the novel Tensor Diffusion approach are obvious: By means of the three- and four-field formulations of the Tensor Stokes problem, a diffusive operator is recovered in the momentum equation even of the non-solvent flow model, which for example conceptually allows a successful application of multigrid solvers or operator splitting techniques. Possibly, the stability regarding certain Finite Element approximations of the velocity and stress fields might be recovered as well. Most of all, actually deriving a generalised Tensor Stokes problem correlating to the underlying original nonlinear viscoelastic flow model would significantly improve the corresponding numerical approach. On the one hand, the size of the (discrete) problem is reduced significantly, since no extra-stress tensor – or even multiple tensors in the multi-mode approach of differential models – needs to be considered, which decreases the computational effort of the numerical solution approach. In similar way, no stress integral needs to be evaluated in terms of integral material laws, which reduces the computation time as no time-dependent ingredient is regarded at all – which is the case in the original integral model even for direct steady-state configurations. At the same time, no such thing as the hyperbolic character of the constitutive equations needs to be taken into account, and the High Weissenberg Number Problem might be irrelevant as well. On the other hand, numerical schemes especially designed for solving the (Navier-)Stokes equations can be applied in the context of simulating viscoelastic fluids in complex two- (or even three-)dimensional flow configurations. Since such techniques are in a further developed state than typical approaches considering viscoelastic three- or even four-field formulations, a more efficient as well as robust numerical framework could be obtained for predicting viscoelastic material behaviour. Overall, the novel Tensor Diffusion approach offers the possibility to significantly improve numerical simulations of (realistic) viscoelastic fluid flows, since various issues of existing numerical methods are potentially weakened or even removed.

In this work, the Tensor Diffusion approach is analysed in a first step in the context of fully developed channel flows, which allows a deeper insight into the underlying mathematical and numerical properties of the problem. It turns out, that this framework indeed offers the possibility to reduce the complete differential or integral viscoelastic flow model in certain cases to a generalised Tensor Stokes problem, which in principle allows to exploit the corresponding benefits described above. Thereby, the tensor-valued viscosity is expressed analytically depending on the velocity field or the shear rate of the flow, while a numerical determination is applied in the remaining cases. As a result, viscoleastic effects occuring in Poiseuille-like flows, including shear thinning or curved pressure contour lines, originally described by (non)linear differential and integral viscoelastic flow models are recovered from a Stokes-like problem. In a second step, prototypical or primal Finite Element techniques are proposed for determining the Diffusion Tensor numerically in terms of actual two-dimensional configurations, where no analytical modelling can be established at the present state of the approach. Naturally, an explicit model of the Diffusion Tensor regarding complex two-dimensional flow states is desired as well, which will be part of future work. However, the derived numerical framework is validated and evaluated in the context of the Flow around cylinder benchmark as well as the flow in a rounded contraction. Thus, the foundational applicability of the Tensor Diffusion approach is pointed out, but certain limits are detected as well. Thereby, integral models are treated only conceptually, while most of the findings are validated and evaluated by means of differential models due to the existing numerical framework and widely available references.

The structure of the thesis is as follows. A general framework regarding the Finite Element Method for simulating viscoelastic fluid flows is described in Chpt. 2 based on established approaches concerning differential as well as integral constitutive equations. After outlining the principal difficulties and challenges arising in this context, the Tensor Diffusion approach is proposed as a novel technique for potentially resolving at least some of the highlighted issues. In Chpt. 3, the methodoloy of calculating fully developed flow profiles with respect to viscoelastic constitutive equations is presented, which provides the basis for the initial validation of the Tensor Diffusion approach. In addition, structural limits of certain viscoelastic models are detected, which prevent the successful computation of arbitrary high relaxation times or – generally speaking – realistic material and model parameters. The validation of the Tensor Diffusion approach is performed in Chpt. 4 by means of quasi one-dimensional as well as complex two-dimensional flow configurations. In addition, this novel approach is evaluated with respect to applying multigrid techniques in terms of Newton's method for monolithically solving the discrete nonlinear systems. Moreover, the High Weissenberg Number Problem is taken into account in the numerical investigations as well, followed by analysing prototypical coupled and decoupled solution techniques based on the Tensor Diffusion approach. Concluding remarks as well as a summary of the thesis are given in Chpt. 5.

Chapter 2

Finite Element techniques for viscoelastic fluid flows

In this work, the numerical simulation of viscoelastic fluid flows by means of the Finite Element Method is investigated. Such highly viscous, “slow” or creeping flows are typically characterised by the well-known instationary Stokes equations, which determine the velocity vector field $\mathbf{u} = \mathbf{u}\left(t, \mathbf{x}\right)$ depending on the time $t \in \mathbb{R}$ and the spatial coordinate vector $\mathbf{x} \in \mathbb{R}^d$, $d \in \{2, 3\}$, based on the conservation of mass and momentum. In detail, the corresponding governing equations read

$$\rho \frac{\partial \mathbf{u}}{\partial t} - \nabla \cdot \mathbf{\Pi} = \mathbf{0}, \quad \nabla \cdot \mathbf{u} = 0 \tag{2.1}$$

under the assumption of isothermal and incompressible flows leading to a constant density ρ, which is usually set to unity in this work. A key quantity for describing the underlying material behaviour is constituted by the Cauchy stress tensor $\mathbf{\Pi} = \mathbf{\Pi}\left(t, \mathbf{x}\right)$ [1, 2], which can be written as

$$\mathbf{\Pi} = -p\mathbf{I} + 2\eta_s \mathbf{D}\left(\mathbf{u}\right) + \mathbf{\Sigma} \tag{2.2}$$

Here, $p = p\left(t, \mathbf{x}\right)$ denotes the scalar pressure field, $\mathbf{I}$ the unit tensor, $\mathbf{D}\left(\mathbf{u}\right) = \frac{1}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^\top\right)$ the strain-rate tensor, that is the symmetric part of the velocity gradient, and $\mathbf{\Sigma} = \mathbf{\Sigma}\left(t, \mathbf{x}\right)$ the extra-stress tensor. Furthermore, η_s refers to the solvent viscosity of the fluid, which specifies the amount of viscous contribution to the total stress. By means of specifying the components of the Cauchy stress tensor from Eq. (2.2), various rheological properties of the considered fluid can be covered. In detail, modelling the flow of pure Newtonian fluids includes a constant viscosity η_s together with a vanishing extra-stress tensor. A first extension towards predicting nonlinear material behaviour is given by so-called shear thinning fluids, where η_s is defined as a decaying function of the shear rate $\dot{\gamma} \approx \|\mathbf{D}\left(\mathbf{u}\right)\|$ [3]. Finally, viscoelastic effects are regarded based on the above decomposition of the total stress tensor $\mathbf{\Pi}$ by specifying the evolution of the extra-stress tensor $\mathbf{\Sigma}$ according to a constitutive equation of differential or integral type. A selection of such material models is presented in Sec. 2.1, followed by a description of corresponding Finite Element techniques in Secs. 2.2 and 2.3, respectively. In doing so, several challenges and difficulties in simulating viscoelastic fluid flows are detected, based on which a novel approach is proposed in Sec. 2.4 for potentially improving the numerical techniques highlighted beforehand.

2.1 Governing equations

For characterising the flow of a viscoelastic fluid, the Stokes equations (2.1) need to be supplemented with a constitutive equation, which describes the evolution of the extra-stress tensor $\mathbf{\Sigma}$ based on the velocity field $\mathbf{u}$. Commonly applied models for predicting the material behaviour of viscoelastic fluids are of differential type reading

$$\frac{\partial \mathbf{\Sigma}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{\Sigma} - \nabla \mathbf{u}^\top \cdot \mathbf{\Sigma} - \mathbf{\Sigma} \cdot \nabla \mathbf{u} + \mathbf{Z}\left(\Lambda, \eta_p, \mathbf{\Sigma}\right) = 2 \frac{\eta_p}{\Lambda} \mathbf{D}\left(\mathbf{u}\right) \tag{2.3}$$

where $\Lambda, \eta_p > 0$ denote the (single-mode) relaxation time and polymeric viscosity, respectively, of the fluid at hand. Typically, these material parameters are determined by fitting linear viscoelastic material data obtained from small oscillatory shear tests [4, 5, 6]. However, the Oldroyd-B or Upper-Convected Maxwell model (UCM, for $\eta_s = 0$ in Eq. (2.2)) are often considered for developing and improving numerical discretisation and solution techniques regarding viscoelastic fluid flow simulations due to the simple structure of the corresponding model function [1]. In detail, these material laws are obtained in case of considering

$$\mathbf{Z}\left(\Lambda, \eta_p, \mathbf{\Sigma}\right) = \frac{1}{\Lambda} \mathbf{\Sigma} \tag{2.4}$$

in Eq. (2.3) [4, 7]. But, no realistic or meaningful viscoelastic material behaviour can be characterised by means of the above model(s): On the one hand, a constant shear viscosity is predicted, and on the other hand an infinite elongational viscosity results from applying a constant (finite) strain rate over time [1, 6]. In contrast, viscoelastic materials show a so-called shear thinning behaviour, that is a decreasing shear viscosity is observed for increasing shear rates. Moreover, such materials do not consist of an elongational viscosity diverging to infinity. Thus, concerning the prediction of more realistic viscoelastic phenomena the model function can be chosen according to the Giesekus model, where

$$\mathbf{Z}\left(\Lambda, \eta_p, \mathbf{\Sigma}\right) = \frac{1}{\Lambda} \left(\mathbf{\Sigma} + \alpha \frac{\Lambda}{\eta_p} \mathbf{\Sigma} \cdot \mathbf{\Sigma} \right) \tag{2.5}$$

which results from the Oldroyd-B or UCM model by introducing an additional quadratic stress contribution [4, 8, 9]. In doing so, the so-called mobility factor $\alpha \in \left]0, 1\right]$ can be determined, such that a sufficiently accurate approximation of the experimentally measured shear thinning behaviour is obtained. At the same time, the divergence of the elongational viscosity is suppressed in case of $\alpha > 0$ [4, 6]. Likewise, nonlinear viscoelastic material behaviour may be characterised by the Phan-Thien Tanner model (PTT, [1, 4, 9, 10]), where choosing the function $\mathbf{Z}$ in Eq. (2.3) as

$$\mathbf{Z}_{\text{lin}}\left(\Lambda, \eta_p, \mathbf{\Sigma}\right) = \frac{1}{\Lambda} \left(1 + \kappa \frac{\Lambda}{\eta_p} \operatorname{tr}\left(\mathbf{\Sigma}\right) \right) \mathbf{\Sigma} \tag{2.6a}$$

$$\mathbf{Z}_{\text{exp}}\left(\Lambda, \eta_p, \mathbf{\Sigma}\right) = \frac{1}{\Lambda} \exp\left(\kappa \frac{\Lambda}{\eta_p} \operatorname{tr}\left(\mathbf{\Sigma}\right) \right) \mathbf{\Sigma} \tag{2.6b}$$

results in the linear or exponential version, respectively. Similar to the Giesekus model, also the PTT model is capable of predicting a shear thinning effect by specifying the nonlinear material parameter $\kappa \in \,]0,1]$, while the Oldroyd-B or UCM model is obtained for $\kappa = 0$.

In case of actually simulating flows of realistic viscoelastic materials, the above form of the differential flow model will not result in an adequate reproduction of experimental data, since the underlying relaxation time spectrum easily encompasses several decades [6]. Thus, considering multiple relaxation times or modes already in the rheological model is required to achieve a satisfactory description of the relaxation behaviour of the material. Therefore, the extra-stress tensor $\boldsymbol{\Sigma}$ as part of the Cauchy stress tensor $\boldsymbol{\Pi}$ from Eq. (2.2) is decomposed into a sum of single stress tensors according to $\boldsymbol{\Sigma} = \sum_{k=1}^{K} \boldsymbol{\Sigma}_k$. Thereby, each stress tensor is characterised by satisfying a separate constitutive equation similar to Eq. (2.3), in detail

$$\frac{\partial \boldsymbol{\Sigma}_k}{\partial t} + \mathbf{u} \cdot \nabla \boldsymbol{\Sigma}_k - \nabla \mathbf{u}^\top \cdot \boldsymbol{\Sigma}_k - \boldsymbol{\Sigma}_k \cdot \nabla \mathbf{u} + \mathbf{Z}\left(\Lambda_k, \eta_{p,k}, \boldsymbol{\Sigma}_k\right) = 2\frac{\eta_{p,k}}{\Lambda_k}\mathbf{D}\left(\mathbf{u}\right) \tag{2.7}$$

including separate parameters $\Lambda_k, \eta_{p,k}$ [4]. In fact, applying this so-called multimode approach involves a number of $K \gg 1$ modes Λ_k, $k = 1, \ldots, K$, for obtaining an appropriate approximation of (non)linear viscoelastic behaviour, particularly in terms of polymer melts [6]. Tranferring this issue to actual two- or three-dimensional simulations of viscoelastic fluid flows, each of the separate stress tensors $\boldsymbol{\Sigma}_k$ needs to be regarded as an independent numerical flow variable. Thus, a significant growth of the problem size can be expected for considering constitutive equations of differential type in numerical simulations of realistic viscoelastic fluids.

At least in this concern, a promising alternative approach is provided from a numerical point of view by describing the material behaviour of viscoelastic fluids via integral constitutive equations, which is outlined below. Furthermore, advantages are achieved compared to differential models from a rheological perspective as well, since the full deformation history is taken into account for characterising the current stress state. Thus, a suitable numerical treatment of integral material laws is of intensified interest, especially concerning the prediction of realistic viscoelastic material behaviour. As indicated above, integral viscoelastic models are intended to be solved within the Finite Element Method together with the Stokes equations, which are commonly implemented in an Eulerian frame of reference. In Ref. [11], several techniques for incorporating integral constitutive equations into a Finite Element framework are discussed, but most of them – like the so-called Streamline Finite Element Method or a decoupled Eulerian-Lagrangian approach – include a Lagrangian reference frame. Actually, the Lagrangian reference frame is an intuitive approach when considering integral models, since the deformation history of a fluid particle can be traced along the corresponding streamline. Unfortunately, this causes high numerical effort. Moreover, the need of re-meshing the computional domain during calculation arises in Lagrangian methods, which is avoided in Eulerian approaches due to considering a fixed fluid portion. Hence, a corresponding Eulerian framework is desired also for integral constitutive equations, which naturally consists of several challenges as well.

One of the most suitable approaches to handle integral material models in an Eulerian frame is the so-called Deformation Fields Method (DFM, [11, 12, 13, 14, 15]). A central object in this Eulerian scheme is the Finger tensor, a coordinate-frame invariant measure of the deformation occurring between a past time $t' \in \left]-\infty, t\right]$ and the current time t, given in Eulerian coordinates. In detail, at each time instant t' in the deformation history of the flow, a new Finger tensor or Deformation Field $\mathbf{B}_{t'} = \mathbf{B}_{t'}\left(s, \mathbf{x}\right)$ is created and evolved according to the evolution equation

$$\frac{\partial}{\partial s}\mathbf{B}_{t'}\left(s\right) + \mathbf{u}\left(s\right) \cdot \nabla \mathbf{B}_{t'}\left(s\right) - \nabla \mathbf{u}\left(s\right)^{\top} \cdot \mathbf{B}_{t'}\left(s\right) - \mathbf{B}_{t'}\left(s\right) \cdot \nabla \mathbf{u}\left(s\right) = \mathbf{0} \tag{2.8}$$

for $s \in [t', t]$ and fixed t', where $\mathbf{B}_{t'}\left(t'\right) = \mathbf{I}$. Following Refs. [4, 16], the above evolution equation corresponds to the definition of the Finger tensor as a symmetric positive (semi-)definite tensor consisting of a vanishing upper-convected derivative

$$\overset{\triangledown}{\mathbf{B}}_{t'}\left(s\right) := \frac{\partial}{\partial s}\mathbf{B}_{t'}\left(s\right) + \mathbf{u}\left(s\right) \cdot \nabla \mathbf{B}_{t'}\left(s\right) - \nabla \mathbf{u}\left(s\right)^{\top} \cdot \mathbf{B}_{t'}\left(s\right) - \mathbf{B}_{t'}\left(s\right) \cdot \nabla \mathbf{u}\left(s\right)$$

Hence, benefitial and physically meaningful properties concerning the resulting extra-stress tensor are established. However, the deformation and thus the state of the stress $\mathbf{\Sigma} = \mathbf{\Sigma}\left(t\right)$ of the flow at the current time instant t is computed in case of differential models only based on the (given) current velocity field $\mathbf{u} = \mathbf{u}\left(t\right)$, that is without explicitly considering the complete deformation history. In contrast, indeed the "memory" of viscoelastic materials is taken into account in terms of integral constitutive equations. In this regard, the stress at the current time instant t is built by integrating the deformation history of the fluid, that is all deformations applied in the past $t' \in \left]-\infty, t\right]$ described by the Deformation Fields $\mathbf{B}_{t'}$. In detail, integral models considered in numerical flow simulations are often of the so-called time-separable Rivlin-Sawyers (or Kaye-BKZ) type [1, 4, 11], where the extra-stress tensor is written as an infinite integral of the form

$$\mathbf{\Sigma}\left(t\right) = \int_{-\infty}^{t} m\left(t - t'\right) \left[\phi_1\left(I_1, I_2\right)\mathbf{B}_{t'}\left(t\right) + \phi_2\left(I_1, I_2\right)\mathbf{B}_{t'}^{-1}\left(t\right)\right] \,\mathrm{d}t' \tag{2.9}$$

Thereby, the memory of viscoelastic fluids is mimicked by means of the memory function m, which can be chosen as a superposition of exponentials [4, 11], in detail

$$m\left(s\right) = \sum_{k=1}^{K} \frac{\eta_{p,k}}{\Lambda_k^2} \exp\left(-\frac{s}{\Lambda_k}\right) \tag{2.10}$$

Thus, deformations applied at large past times $t' \ll t$ contribute less to the current stress and vanish for $t \to \infty$ when considering the memory function in the stress integral from Eq. (2.9). Furthermore, applying integral models in a numerical framework allows to keep the problem size fixed even in case of the multi-mode approach, because no additional numerical variables need to be considered when increasing the number K of modes shown in Eq. (2.10). However, the actual material model resulting from the stress integral (2.9) is specified by choosing the empirical functions $\phi_{1,2}$, which enable a modelling of nonlinear material behaviour depending on the two non-trivial invariants $I_{1,2}$ of the Finger tensor $\mathbf{B}_{t'}$, where

$$I_1\left(\mathbf{B}_{t'}\right) = \operatorname{tr}\left(\mathbf{B}_{t'}\right), \quad I_2\left(\mathbf{B}_{t'}\right) = \frac{1}{2}\left(\operatorname{tr}\left(\mathbf{B}_{t'}\right)^2 - \operatorname{tr}\left(\mathbf{B}_{t'}^2\right)\right)$$

Due to the incompressibility of the considered flow, $I_3 = \det\left(\mathbf{B}_{t'}\right) = 1$ holds regarding the third invariant [12]. For example, the integral version of the UCM model, introduced above as differential model in Eq. (2.4), is obtained by setting $\phi_1 \equiv 1$ and $\phi_2 \equiv 0$, where usually $\mathbf{B}$ is replaced by $\mathbf{B} - \mathbf{I}$ in Eq. (2.9) [4]. But, the above proposed general form is recovered for absorbing the additional (isotropic) contribution into the pressure in the momentum equation of the Stokes equations (see Sec. 3.2 or 4.1 for details). Another model frequently used in numerical investigations is the Papanastasiou-Scriven-Macosko model (PSM, [5, 12, 13, 14]), where the corresponding damping functions read

$$\phi_1\left(I_1, I_2\right) = \frac{1}{1 + \gamma_1\left(I_1 - 3\right) + \gamma_2\left(I_2 - 3\right)}, \quad \phi_2 \equiv 0 \tag{2.11}$$

and the UCM model is recovered for $\gamma_1 = \gamma_2 = 0$. From a rheological point of view, a quite popular model is represented by the integral constitutive equation obtained via

$$\phi_1\left(I_1, I_2\right) = f \exp\left(-n_1\sqrt{I - 3}\right) + (1 - f)\exp\left(-n_2\sqrt{I - 3}\right), \quad \phi_2 \equiv 0 \tag{2.12}$$

introduced by Wagner [5, 17, 18]. Here, a convex combination $I = \alpha I_1 + (1 - \alpha) I_2$ of the two non-trivial invariants is included besides the convex combination of the exponentials, hence $f, \alpha \in [0, 1]$, which results in the UCM model for $n_1 = n_2 = 0$. In addition, the empirical functions in the Wagner-Demarmels model [5] are defined as

$$\phi_1\left(I_1, I_2\right) = \frac{1 + \beta}{1 + \alpha\sqrt{\left(I_1 - 3\right)\left(I_2 - 3\right)}} \tag{2.13a}$$

$$\phi_2\left(I_1, I_2\right) = \frac{\beta}{1 + \alpha\sqrt{\left(I_1 - 3\right)\left(I_2 - 3\right)}} \tag{2.13b}$$

In contrast to the above models, a non-vanishing contribution regarding the inverse of the Finger tensor is present in case of $\beta > 0$, while the UCM model is recovered for $\alpha = \beta = 0$. Similar to the differential Giesekus or PTT model, also the nonlinear integral models presented above are capable of predicting typical viscoelastic material behaviour like shear thinning (see Ref. [5] or Sec. 3.2).

The above viscoelastic flow models are formulated regarding three-dimensional configurations, but two-dimensional flow states are considered in this work. Hence, the constitutive equations need to be expressed in terms of two spatial dimensions, which is done in case of differential constitutive laws by simply considering

$\boldsymbol{\Sigma} \in \mathbb{R}^{2\times 2}$. Following Ref. [19], the two-dimensional formulation of integral viscoelastic constitutive equations is derived from the above model functions by considering a pseudo three-dimensional Finger tensor of the form

$$\tilde{\mathbf{B}} = \begin{pmatrix} B_{11} & B_{12} & 0 \\ B_{12} & B_{22} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Based on the upper-left block matrix $\mathbf{B} \in \mathbb{R}^{2\times 2}$ of the tensor $\tilde{\mathbf{B}}$, it can be easily shown that

$$\begin{aligned} I_1\left(\tilde{\mathbf{B}}\right) &= \operatorname{tr}(\mathbf{B}) + 1, && (2.14a) \\ I_2\left(\tilde{\mathbf{B}}\right) &= \frac{1}{2}\left(\operatorname{tr}\left(\tilde{\mathbf{B}}\right)^2 - \operatorname{tr}\left(\tilde{\mathbf{B}}^2\right)\right) = \operatorname{tr}(\mathbf{B}) + \det(\mathbf{B}) && (2.14b) \end{aligned}$$

holds regarding the two non-trivial invariants, which mainly characterise the specific material model. Since $\det(\mathbf{B}) = 1$ in case of incompressible flows [12], the two invariants coincide and thus result in a single invariant of $\mathbf{B}$, that is $I = I_1 = I_2 = \operatorname{tr}(\mathbf{B}) + 1$. Hence, the two-dimensional versions of the integral viscoelastic flow models mentioned above are obtained by inserting the invariant I into the damping functions from Eqs. (2.11), (2.12) and (2.13). In detail, the damping functions evolve to

$$\phi_1 = \frac{1}{1+\gamma\left(\operatorname{tr}(\mathbf{B}) - 2\right)},\ \phi_2 = 0 \qquad (2.15a)$$

$$\phi_1 = f\exp\left(-n_1\sqrt{\operatorname{tr}(\mathbf{B}) - 2}\right) + (1-f)\exp\left(-n_2\sqrt{\operatorname{tr}(\mathbf{B}) - 2}\right),\ \phi_2 = 0 \qquad (2.15b)$$

$$\phi_1 = \frac{1+\beta}{1+\alpha\sqrt{\left(\operatorname{tr}(\mathbf{B}) - 2\right)^2}},\ \phi_2 = \frac{\beta}{1+\alpha\sqrt{\left(\operatorname{tr}(\mathbf{B}) - 2\right)^2}} \qquad (2.15c)$$

in case of the PSM, Wagner and Wagner-Demarmels model, respectively, where $\phi_{1,2} = \phi_{1,2}(I_1, I_2) = \phi_{1,2}(\operatorname{tr}(\mathbf{B}))$. Note, that the number of involved material parameters might be reduced compared to the three-dimensional case due to the coinciding invariants from Eq. (2.14).

In the following, the numerical treatment of the (stationary) Stokes equations coupled with constitutive laws of differential or integral type within the Finite Element Method is discussed. Therefore, the governing equations have to be complemented by suitable boundary conditions to obtain a well-posed mathematical problem from the corresponding differential as well as integral viscoelastic flow model. Thereby, a reasonable choice might be to prescribe Dirichlet data regarding the velocity field on all boundaries, while the stress fields are set accordingly only at the inflow boundary egde [20], for example to assume a fully developed flow at the entrance of the computational domain. On the remaining boundary edges, a so-called do-nothing boundary condition [21] is imposed. In this regard, the condition

$$p\mathbf{n} - 2\eta_s\mathbf{D}(\mathbf{u}) \cdot \mathbf{n} - \boldsymbol{\Sigma} \cdot \mathbf{n} = 0$$

including the outer normal vector $\mathbf{n}$ might be prescribed on the outflow boundary edge of the computational domain in case of assigning a do-nothing boundary condition to the full set of variables. A detailed study concerning corresponding in- and outflow profiles, especially regarding the velocity field, is performed in Chpt. 3. For now, the specific shape of the boundary data is left open. Note, that no Dirichlet data needs to be set regarding the pressure field, which is taken to be a function in the Lebesgue space $L^2(\Omega)$ (see Sec. 2.2.1) and is thus determined only up to an additive constant. This constant might be chosen, such that a zero pressure is obtained at a specific point in the computational domain or the integral mean value of the pressure vanishes.

However, constitutive equations of differential type are quite straightforward to apply within Finite Element simulations concerning viscoelastic fluid flows, as these equations can be treated in similar manner as the Stokes equations themselves. Nevertheless, a couple of issues need to be taken into account in this context, which is outlined in Sec. 2.2. Designing a suitable treatment of integral models within the Finite Element Method is a challenging task as well, where a possible procedure is described in Sec. 2.3.

2.2 Numerical treatment of differential constitutive equations

The focus of this work lies in considering direct steady-state configurations, which are relevant in many benchmark computations as well as simulations of industrial applications. Thus, a stationary version of the above differential viscoelastic flow models is taken into account in the following to discuss a suitable implementation within the Finite Element Method. For completeness, the numerical handling of the time-dependent case is briefly discussed at the end of this section. To obtain the stationary problem formulation of the corresponding set of equations presented in Sec. 2.1, the derivatives of the velocity as well as stress fields with respect to time are neglected. Consequently, the stationary differential viscoelastic flow model in the unknowns $(\mathbf{u}, \boldsymbol{\Sigma}, p)$ reads

$$\begin{aligned} -2\eta_s \nabla \cdot \mathbf{D}(\mathbf{u}) - \nabla \cdot \boldsymbol{\Sigma} + \nabla p &= \mathbf{0} && \text{(2.16a)} \\ \mathbf{u} \cdot \nabla \boldsymbol{\Sigma} - \nabla \mathbf{u}^\top \cdot \boldsymbol{\Sigma} - \boldsymbol{\Sigma} \cdot \nabla \mathbf{u} + \mathbf{Z}(\Lambda, \eta_p, \boldsymbol{\Sigma}) &= 2\frac{\eta_p}{\Lambda}\mathbf{D}(\mathbf{u}) && \text{(2.16b)} \\ \nabla \cdot \mathbf{u} &= 0 && \text{(2.16c)} \end{aligned}$$

In the following, a general framework regarding Finite Element Methods for differential viscoelastic fluids is presented, based on which an overview of state-of-the-art approaches is given in Sec. 2.2.2. Finally, the specific approach applied in this work is summarised in Sec. 2.2.3.

2.2.1 General framework

The nonlinear set of equations from Eq. (2.16) represents the strong form of the stationary differential viscoelastic flow model, where well-posedness as well as existence and uniqueness results are presented in Refs. [22, 23] at least for sufficiently small data, especially concerning the boundary values of the velocity field.

For discretising the above flow model by means of the Finite Element Method, a corresponding weak formulation needs to be derived. In doing so, the unknown fields $(\mathbf{u}, \boldsymbol{\Sigma}, p)$ are determined in some function spaces $\boldsymbol{V} \times \mathsf{S} \times Q$ of vector-, tensor- and scalar-valued functions, respectively, defined on a fixed two-dimensional computational domain $\Omega \subset \mathbb{R}^2$ with boundary Γ. In terms of the Galerkin method applied here, the above equations are multiplied by arbitrary test functions $(\mathbf{v}, \mathbf{S}, q)$ from the same set of function spaces $\boldsymbol{V} \times \mathsf{S} \times Q$, followed by an integration by parts resulting in

$$\int_\Omega \left(2\eta_s \mathbf{D}\left(\mathbf{u}\right) + \boldsymbol{\Sigma} - p\mathbf{I}\right) : \mathbf{D}\left(\mathbf{v}\right)\,\mathrm{dx} = \int_\Gamma \left(\boldsymbol{\Pi}\cdot\mathbf{n}\right)\cdot\mathbf{v}\,\mathrm{d}S\ \forall\mathbf{v}\in\boldsymbol{V} \tag{2.17a}$$

$$\begin{aligned}\int_\Omega \Big(\mathbf{u}\cdot\nabla\boldsymbol{\Sigma} - \nabla\mathbf{u}^\top\cdot\boldsymbol{\Sigma} - \boldsymbol{\Sigma}\cdot\nabla\mathbf{u}\ldots \\ +\mathbf{Z}\left(\Lambda, \eta_p, \boldsymbol{\Sigma}\right) - 2\frac{\eta_p}{\Lambda}\mathbf{D}\left(\mathbf{u}\right)\Big) : \mathbf{S}\,\mathrm{dx} &= 0\ \forall\mathbf{S}\in\mathsf{S}\end{aligned} \tag{2.17b}$$

$$\int_\Omega \left(\nabla\cdot\mathbf{u}\right) q\,\mathrm{dx} = 0\ \forall q\in Q \tag{2.17c}$$

As before, $\boldsymbol{\Pi}$ denotes the Cauchy stress tensor from Eq. (2.2) and $\mathbf{n}$ refers to the outer normal vector of Ω on Γ. However, the specific choice of the function spaces $\boldsymbol{V}$ and Q is well-known in case of considering the weak formulation of the standard Stokes equations, which arises from Eq. (2.17) by neglecting the nonlinear material behaviour and accordingly setting $\boldsymbol{\Sigma} \equiv \mathbf{0}$. In this context, $\boldsymbol{V} = \left(H_0^1\left(\Omega\right)\right)^2$ and $Q = L^2\left(\Omega\right)$, that is the typical Sobolev and Lebesgue spaces, are chosen with respect to the velocity and pressure variables [24, 25, 26, 27]. As an alternative being closer related to the differential viscoelastic flow model, the Stokes equations can be expressed by means of a three-field formulation, which results from Eq. (2.17) in the weak sense for $\Lambda = 0$, $\eta_s = 0$ and $\eta_p > 0$. In addition to $\boldsymbol{V}$ and Q taken from above, a common choice for the function space with respect to the stress is obtained via $\mathsf{S} = \left(L^2\left(\Omega\right)\right)^{2\times2}_{\mathrm{sym}}$ including the symmetry condition $\sigma_{12} = \sigma_{21}$ concerning the off-diagonal components of $\boldsymbol{\Sigma}$ [24, 28, 29, 30, 31]. Hence, the stress tensor is in principle obtained by means of an L^2-projection of the strain-rate tensor $\mathbf{D}\left(\mathbf{u}\right)$. When considering an actual viscoelastic fluid, that is $\Lambda > 0$ in Eq. (2.17b), the extra-stress tensor needs to be chosen from a subset $\mathsf{S} \subset \left(L^2\left(\Omega\right)\right)^{2\times2}_{\mathrm{sym}}$ consisting of a sufficiently higher regularity due to the convective term being present even in the weak formulation considered above. For example, the choice

$$\mathsf{S} = \left\{\mathbf{S} \in \left(L^2\left(\Omega\right)\right)^{2\times2}_{\mathrm{sym}} \mid \bar{\mathbf{u}}\cdot\nabla\mathbf{S} \in \left(L^2\left(\Omega\right)\right)^{2\times2}_{\mathrm{sym}}\right\}$$

is discussed in Ref. [32] for a viscoelastic Oseen-type model. Thereby, the velocity field is fixed in the convective term of the constitutive equation by means of a suitable solenoidal and sufficiently bounded velocity field $\bar{\mathbf{u}} \in \boldsymbol{V}$. At least in terms of this linearised Oldroyd-B model, a higher regularity is imposed on the stress variable, which provides existence and uniqueness results of the corresponding continuous and numerical solutions. But, an extension towards the original viscoelastic flow model can not be specified straightforward. Instead, the convective term may be partially integrated as well by applying the same procedure as proposed below in Eq. (2.56). Thereby, the corresponding derivative can be shifted in the weak formulation to the test function, similar to Eq. (2.58a), by explicitly taking into account the continuity equation (2.16c). By means of this conservative formulation of the constitutive equation, the required regularity concerning the extra-stress tensor is decreased, which results in $\mathsf{S} = \left(L^2\left(\Omega\right)\right)^{2\times 2}_{\mathrm{sym}}$ as described in Ref. [33] for Jeffreys constitutive law, that is basically the Oldroyd-B model in case of neglected deformation terms. However, the weak formulation presented in Eq. (2.17b) is considered in this work, since only first-order derivatives of the velocity and stress fields occur and no further partial integration is necessarily required.

Regarding the spatial discretisation of the weak formulation presented in Eq. (2.17), a triangulation $\mathcal{T}_h$ of the (two-dimensional) computational domain Ω is defined consisting of discrete elements $T_1, \ldots, T_M$ of quadrilateral shape. The representative mesh size is defined as the maximum diameter of all elements, that is $h := \max_{i\in\{1,\ldots,M\}} \mathrm{diam}\left(T_i\right)$. Based on this triangulation, discrete function spaces $\boldsymbol{V}_h$, S_h and Q_h approximating $\boldsymbol{V}$, S and Q are defined by specifying corresponding sets of vector-, tensor- and scalar-valued basis functions $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{N_u})$, $(\boldsymbol{\Phi}_1, \ldots, \boldsymbol{\Phi}_{N_\sigma})$ and $\left(\chi_1, \ldots, \chi_{N_p}\right)$, respectively. By doing so, the approximated solutions

$$\mathbf{u}_h = \sum_{i=1}^{N_u} u_i \boldsymbol{v}_i, \quad \boldsymbol{\Sigma}_h = \sum_{j=1}^{N_\sigma} \sigma_j \boldsymbol{\Phi}_j, \quad p_h = \sum_{k=1}^{N_p} p_k \chi_k \tag{2.18}$$

are obtained regarding the velocity, stress and pressure fields, where the corresponding numbers of degrees of freedom related to the triangulation $\mathcal{T}_h$ are denoted by N_u, N_σ and N_p. Furthermore, $\boldsymbol{u} := (u_i)_{i=1}^{N_u}$, $\boldsymbol{\sigma} := (\sigma_j)_{j=1}^{N_\sigma}$ and $\boldsymbol{p} := (p_k)_{k=1}^{N_p}$ denote the vectors containing the degrees of freedom of the corresponding field variable in the discretised domain Ω_h. By means of the discrete solutions, spatial approximations of the differential operators from the strong form in Eq. (2.16) can be defined as

$$\begin{aligned}
\mathcal{L}_{ij} &= \int_\Omega 2\eta_s \mathbf{D}\left(\boldsymbol{v}_j\right) : \mathbf{D}\left(\boldsymbol{v}_i\right)\, \mathrm{d}\mathbf{x}\ \forall i,j \in \{1,\ldots N_u\} && \text{(2.19a)}\\
\mathcal{C}_{ij} &= \int_\Omega \boldsymbol{\Phi}_j : \mathbf{D}\left(\boldsymbol{v}_i\right)\, \mathrm{d}\mathbf{x}\ \forall i \in \{1,\ldots,N_u\}, j \in \{1,\ldots N_\sigma\} && \text{(2.19b)}\\
\mathcal{B}_{ij} &= \int_\Omega \chi_j \mathbf{I} : \mathbf{D}\left(\boldsymbol{v}_i\right)\, \mathrm{d}\mathbf{x}\ \forall i \in \{1,\ldots,N_u\}, j \in \{1,\ldots N_p\} && \text{(2.19c)}\\
\mathcal{K}_{ij}\left(u_j\right) &= \int_\Omega \Big(u_j \boldsymbol{v}_j \cdot \nabla \boldsymbol{\Phi}_j - \nabla\left(u_j \boldsymbol{v}_j\right)^\top \cdot \boldsymbol{\Phi}_j - \boldsymbol{\Phi}_j \cdot \nabla\left(u_j \boldsymbol{v}_j\right) \ldots \\
&\qquad \cdots + \mathbf{Z}\left(\Lambda, \eta_p, \boldsymbol{\Phi}_j\right)\Big) : \boldsymbol{\Phi}_i\, \mathrm{d}\mathbf{x}\ \forall i,j \in \{1,\ldots,N_\sigma\} && \text{(2.19d)}\\
\mathcal{D}_{ij} &= \int_\Omega 2\frac{\eta_p}{\Lambda} \mathbf{D}\left(\boldsymbol{v}_j\right) : \boldsymbol{\Phi}_i\, \mathrm{d}\mathbf{x}\ \forall i \in \{1,\ldots,N_\sigma\}, j \in \{1,\ldots N_u\} && \text{(2.19e)}
\end{aligned}$$

based on the above weak formulation. Thereby, the degree of freedom of the velocity field in Eq. (2.19d) is related to the degree of freedom of the stress variable in a suitable way. However, the discrete operators presented above denote approximations of the operators from Eq. (2.16) according to

$$\begin{gathered}
\mathcal{L}\boldsymbol{u} \approx 2\eta_s \nabla \cdot \mathbf{D}\left(\mathbf{u}\right), \quad \mathcal{C}\boldsymbol{\sigma} \approx \nabla \cdot \boldsymbol{\Sigma}, \quad \mathcal{B}\boldsymbol{p} \approx \nabla p \\
\mathcal{K}\left(\boldsymbol{u}\right)\boldsymbol{\sigma} \approx \mathbf{u} \cdot \nabla \boldsymbol{\Sigma} - \nabla \mathbf{u}^\top \cdot \boldsymbol{\Sigma} - \boldsymbol{\Sigma} \cdot \nabla \mathbf{u} + \mathbf{Z}\left(\Lambda, \eta_p, \boldsymbol{\Sigma}\right) \\
\mathcal{B}^\top \boldsymbol{u} \approx \nabla \cdot \mathbf{u}, \quad \mathcal{D}\boldsymbol{u} \approx 2\frac{\eta_p}{\Lambda}\mathbf{D}\left(\mathbf{u}\right)
\end{gathered}$$

Finally, a discrete version of the weak formulation from Eq. (2.17) is obtained, which can be written in matrix-vector notation as

$$\begin{pmatrix} \mathcal{L} & \mathcal{C} & \mathcal{B} \\ \mathcal{D} & \mathcal{K}\left(\boldsymbol{u}\right) & 0 \\ \mathcal{B}^\top & 0 & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{\sigma} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \mathbf{r}_u \\ \mathbf{r}_\sigma \\ \mathbf{r}_p \end{pmatrix} \tag{2.20}$$

Note, that the boundary values are included in the components $\mathbf{r}_u$, $\mathbf{r}_\sigma$ and $\mathbf{r}_p$ of the right-hand side.

In this context, a couple of numerical challenges may arise. First of all, the approximating spaces regarding the velocity and pressure fields need to obey a compatibility condition [34]. By satisfying this so-called inf-sup or LBB condition (named after Ladyzhenskaya, Babuška and Brezzi)

$$\sup_{\mathbf{v} \in \boldsymbol{V}_h} \frac{\int_\Omega \left(\nabla \cdot \mathbf{v}\right) q \, \mathrm{d}\mathbf{x}}{\|\mathbf{v}\|_{1,\Omega}} \geq \alpha \|q\|_{0,\Omega} \qquad \forall\, q \in Q_h \tag{2.21}$$

for a constant $\alpha > 0$ independent of the mesh, a stable approximation of the velocity-pressure subproblem is provided. Here, $\|.\|_{1,\Omega}$ and $\|.\|_{0,\Omega}$ denote the norms with respect to the Sobolev space $H^1\left(\Omega\right)$ and Lebesgue space $L^2\left(\Omega\right)$, respectively, defined regarding the continuous computational domain Ω. Furthermore, a similar stability condition is also imposed on the velocity-stress coupling according to Ref. [35], which is why the discrete velocity and stress spaces $\boldsymbol{V}_h$ and S_h need to satisfy

$$\sup_{\boldsymbol{\Sigma} \in \mathsf{S}_h} \frac{\int_\Omega \boldsymbol{\Sigma} : \mathbf{D}\left(\mathbf{v}\right) \, \mathrm{d}\mathbf{x}}{\|\boldsymbol{\Sigma}\|_{0,\Omega}} \geq \gamma \|\mathbf{v}\|_{1,\Omega} \qquad \forall\, \mathbf{v} \in \boldsymbol{V}_h \tag{2.22}$$

where again $\gamma > 0$ is a mesh-independent constant. In case the above inequality is violated, suitable stabilisation techniques need to be applied for recovering a stable approximation of the flow variables. But following Ref. [35], this additional LBB-condition may be omitted in case of a non-vanishing solvent viscosity $\eta_s > 0$ in the momentum equation (2.16a). Besides the choice of a stable approximation of the velocity-pressure-stress formulation, various numerical challenges arise when elaborating the general concept of a Finite Element discretisation presented above. For example, a suitable numerical treatment of the convection term or the hyperbolic character of the constitutive equation (2.16b), which is emphasised for increasing relaxation times, needs to be taken into account for avoiding spurious oscillations in the numerical solution. But even in that case, the calculations break down at already low or moderate relaxation times or Weissenberg numbers, which is known as the High Weissenberg Number Problem [36, 37]. In fact, the numerical simulations might diverge due to the loss of the positive definiteness of the conformation tensor, which is linked straightforward to the extra-stress tensor, caused by numerical inaccuracy or instability [1, 2]. In addition to the above issues regarding discretisation techniques, the choice of suitable solution methods turns out to be non-trivial as well. As suggested by means of Eq. (2.20), the resulting discrete nonlinear problem can be solved in a monolithic fashion, which might allow for deriving a fast, efficient and robust numerical scheme. But instead, applying a decoupled solution approach can be attractive as well due to the involved high computational costs of fully coupled approaches.

2.2.2 Overview of established FE approaches

In the following, the numerical challenges indicated above as well as further aspects of Finite Element techniques concerning differential viscoelastic flow models are outlined by means of a more or less brief discussion of selected approaches.

For example, the choice of a stable approximation of the velocity, pressure and stress fields is addressed in Ref. [24] based on the fully coupled stationary formulation of the differential viscoelastic model from Eq. (2.20). In detail, the velocity and pressure solutions are approximated by means of biquadratic and bilinear shape functions, respectively, together with the stress being approximated via $n \times n$ bilinar subelements of the approximation of the velocity field. Thus, stable numerical results can be obtained for $n = 3$ or $n = 4$ in terms of the stick-slip problem or a flow through an abrupt contraction. Moreover, the convection term in the constitutive equation is treated by the Streamline Upwind Petrov Galerkin (SUPG) or a non-consistent Streamline Upwind (SU) stabilisation, where the latter is preferable in non-smooth geometries [38].

In fact, the convective contribution in the constitutive equation is stabilised via SUPG or SU by various numerical approaches to simulate differential viscoelastic fluids, while the main differences occur in the treatment of the velocity-stress coupling. In Ref. [39], the stress computation is decoupled from the momentum and continuity equation by updating the stress fields based on the constitutive equation successively on each element together with SU stabilisation. Naturally, this technique requires a special ordering of the elements for having the updated stresses available in upwind direction in terms of the stabilisation procedure. However, the velocity and pressure fields are approximated via the Stokes pair Q_2/P_1, which is also applied in this work (see Sec. 2.2.3). Furthermore, piecewise discontinuous biquadratic shape functions are implemented with respect to the stress, where the degree of freedom in the cell centre is neglected due to consistency regarding the velocity gradient. But at the same time it is indicated, that instead a bilinear stress approximation may avoid spurious oscillations in the context of SU.

The two approaches presented above mainly address the choice of a stable velocity-stress approximation together with a numerical treatment of the convection in the constitutive equation via upwinding techniques. In contrast, a fundamental technique called the Elastic-Viscous Stress Splitting (EVSS) regarding the stabilisation of the velocity-stress coupling is introduced in Ref. [40]. In detail, a diffusive operator acting on the velocity field is added into the momentum equation of the Stokes problem by decomposing the extra-stress tensor into a viscous and an elastic component. Thus, a corresponding operator is present also in the case of no solvent, that is $\eta_s = 0$, as the strain-rate tensor is weighted in principle with the total viscosity $\eta_0 = \eta_s + \eta_p$ instead of only the solvent contribution. Hence, the differential viscoelastic problem is regularised, since the ellipticity of the subproblem formed by the momentum and continuity equation is established or emphasised, offering the use of special numerical algorithms designed for such kind of equations. At the same time, the need to consider a stable velocity-stress approximation according to Eq. (2.22) is removed [35], which allows a corresponding discretisation by means of polynomials of the same degree on each element, for example via biquadratic polynomials as done in this work (again see Sec. 2.2.3). Nevertheless, the EVSS consists of some disadvantages due to considering the elastic stress as primal numerical stress variable instead of the extra-stress tensor [41]. On the one hand, the constitutive equation needs to be transformed for characterising the evolution of the elastic stress, which is not possible in general, for example in case of conformation tensor-based models. On the other hand, the upper-convected derivative of the strain-rate tensor $\mathbf{D}$ occurs in the transformed version of the prototypical constitutive equation from Eq. (2.3), which requires some further attention. In detail, one might perform an additional integration by parts (similar to Eq. (2.56)), which would require a special treatment of the arising boundary integral given in Ref. [42] at the outflow of the computational domain. Alternatively, the strain-rate tensor can be approximated in terms of an additional numerical variable, which avoids higher-order terms in the weak formulation.

Similarly, the strain-rate tensor is considered as additional independent numerical unknown in the Discrete Elastic-Viscous Stress Splitting (DEVSS) proposed in Ref. [41], based on which artificial diffusion is introduced into the momentum equation. But, no change in the stress variable or the constitutive equation is carried out and hence no higher-order terms regarding derivatives of the velocity field need to be taken into account. Instead, an additional diffusive operator is inserted into the left- as well as the right-hand side of the momentum equation, which results in the (stationary) set of equations

$$\begin{aligned} -2\left(\eta_s+\alpha\right)\nabla\cdot\mathbf{D}\left(\mathbf{u}\right)+\nabla p &= \nabla\cdot\left(\boldsymbol{\Sigma}-2\alpha\mathbf{E}\right) && \text{(2.23a)}\\ \mathbf{u}\cdot\nabla\boldsymbol{\Sigma}-\nabla\mathbf{u}^\top\cdot\boldsymbol{\Sigma}-\boldsymbol{\Sigma}\cdot\nabla\mathbf{u}+\mathbf{Z}\left(\Lambda,\eta_p,\boldsymbol{\Sigma}\right) &= 2\frac{\eta_p}{\Lambda}\mathbf{D}\left(\mathbf{u}\right) && \text{(2.23b)}\\ \mathbf{E} &= \mathbf{D}\left(\mathbf{u}\right) && \text{(2.23c)}\\ \nabla\cdot\mathbf{u} &= 0 && \text{(2.23d)} \end{aligned}$$

where $\alpha > 0$ is an artificial model parameter. Naturally, the original version of the flow model from Eq. (2.16) stays unchanged when applying the DEVSS on the continuous level. But on the discrete level, the additional equation (2.23c) leads to approximation errors in the newly introduced variable $\mathbf{E}$ with respect to the strain-rate tensor $\mathbf{D}$, which results in additional (stabilising) contributions in the discrete system. In fact, a diffusive operator acting on the velocity field is obtained in the momentum equation even in the non-solvent case $\eta_s = 0$, which is why for example the compatibility condition from Eq. (2.22) regarding a stable velocity-stress approximation is suppressed. Consequently, the DEVSS shows improved properties concerning the stability of the numerical method, but leads to an increased problem size due to the original problem being extended to a four-field formulation in $(\mathbf{u}, \boldsymbol{\Sigma}, \mathbf{E}, p)$. Thus, there will be an increasing effort in terms of a fully coupled solution approach, which is why a decoupled approach is proposed in Ref. [41]. The corresponding procedure consists of successively solving Eqs. (2.23a), (2.23d) for $(\mathbf{u}, p)$ with the stress acting as a body force, Eq. (2.23b) again in combination with SU for $\mathbf{\Sigma}$ and Eq. (2.23c) for $\mathbf{E}$. In contrast to approaches based on the original discrete system from Eq. (2.20), such a segregated solution scheme is applicable in terms of the DEVSS even for $\eta_s = 0$, since a regular system regarding the velocity and pressure is obtained from Stokes subproblem due to $\alpha > 0$.

Thus, one of the benefits of applying the (D)EVSS is the ability to solve the stationary differential viscoelastic flow model in a decoupled fashion, even if the solvent viscosity vanishes. For example, this is a key feature for simulating pure polymer melts, that is $\eta_s = 0$, where multiple modes and thus extra-stress tensors need to be considered simultaneously in the flow model, which makes a monolithic approach unpractical. The same problem occurs in the context of transient simulations of viscoelastic fluids in case of neglected inertia in addition to the solvent viscosity. In detail, the density is set to $\rho = 0$ causing also the time-derivative of the velocity field to vanish from the momentum equation. Actually, this is consistent to reducing the Navier-Stokes equations to the case of creeping flows, where the convective operator in the momentum equation is dropped, too. Consequently, the only present time-derivative in the differential model occurs in the constitutive equation (2.3). Naturally, a fully implicit temporal discretisation in terms of a coupled solution approach leads to a robust and accurate prediction of the transient flow behaviour, but large numerical effort is needed. Thus, a decoupled solution approach may be of interest, which at first glance is applicable only in case of a present solvent contribution. In the non-solvent case (and also for neglected inertia), a suitable decoupled transient solution approach of second order is proposed in Ref. [43]. Besides the typical stabilisation via DEVSS and SUPG regarding the velocity-stress coupling and the stress convection, the (implicitly treated) stress in the momentum equation is replaced by an expression derived from the constitutive equation. In detail, the constitutive equation is discretised semi-implicitly with respect to time, where the stresses are taken explicitly and the velocity is taken implicitly. Thus, accordingly substituting the extra-stress tensor leads to a re-coupling of the velocity field in the momentum equation even for $\eta_s = \rho = 0$. An extension of this approach to the case $\rho > 0$ – but still $\eta_s = 0$ – which imposes restrictions on the time step for avoiding spurious oscillations due to the present convection operator, is proposed in Ref. [44].

In the numerical approaches described above, stabilisation techniques are applied mainly regarding the coupling of the velocity and stress approximation as well as the hyperbolic character of the constitutive equation. At the same time, usually a stable approximation is chosen concerning the velocity and pressure fields. In an alternative approach based on a Galerkin Least-Squares (GLS) formulation as proposed in Ref. [45], a global stabilisation is applied taking into account all numerical variables as well as the convective term in the constitutive equation. Thus, by adding penalising terms in form of least-squares expressions of the residuals, the compatibility conditions from Eqs. (2.21) and (2.22) are suppressed, in principle contributing to stable simulations even for low equal-order approximations of the numerical unknowns. Moreover, the set of nonlinear equations can be converted into a set of partial differential equations of first order by considering the velocity gradient as independent numerical variable via the DEVSS, both allowing a relatively simple numerical implementation as well as a cheap calculation of the flow model. But at the same time, the problem size is increased due to the additional variable approximating the strain-rate tensor.

In Refs. [36, 37, 46] a stabilisation approach concerning the High Weissenberg Number Problem is introduced. Thereby, the numerical simulations break down at relatively small values of the Weissenberg number $\mathrm{We} = \Lambda \frac{u_c}{l_c}$, which specifies the nonlinearity of the problem based on the relaxation time Λ as well as the characteristic velocity u_c and length l_c. Naturally, the hyperbolic character of the constitutive equation is enhanced for increasing relaxation times, which causes several numerical challenges. At the same time, the source of this numerical problem is traced back to an exponential growth of the stress variable observed for increasing relaxation times, which might not be captured by polynomial approximations applied in the context of a Finite Element discretisation. Hence, the positive definiteness of the conformation tensor $\mathbf{T}$, which is provided on an analytical level [16], may get lost in terms of numerical computations due to such inaccuracy, further leading to oscillations or instabilities. Therefore, a suitable variable transformation may be applied, where the logarithm of the conformation tensor, which is defined according to

$$\boldsymbol{\Sigma} = \frac{\eta_p}{\Lambda} \left(\mathbf{T} - \mathbf{I} \right) \tag{2.24}$$

instead of the extra-stress tensor itself is considered as numerical variable. By doing so, the exponential growth of the stress is balanced and the positive definiteness of the conformation tensor is preserved. Since $\mathbf{T}$ is not only positive definite, but also symmetric according to the extra-stress tensor $\boldsymbol{\Sigma}$, it can be diagonalised offering the possibility to introduce the matrix-logarithm

$$\boldsymbol{\Psi} = \log \left(\mathbf{T} \right) = \mathbf{R} \log \left(\mathbf{L} \right) \mathbf{R}^{\top}$$

as new stress variable, where $\mathbf{R}$ and $\mathbf{L}$ denote suitable rotation and diagonal matrices, respectively [47]. By following Refs. [36, 37, 46], applying a certain decomposition of the velocity gradient in the constitutive equation (2.3) yields the corresponding Log-Conformation Representation (LCR), in detail

$$\frac{\partial \boldsymbol{\Psi}}{\partial t} + \mathbf{u} \cdot \nabla \boldsymbol{\Psi} - \left(\boldsymbol{\Omega} \cdot \boldsymbol{\Psi} - \boldsymbol{\Psi} \cdot \boldsymbol{\Omega} \right) - 2\mathbf{B} = \mathbf{H} \left(\boldsymbol{\Psi} \right) \tag{2.25}$$

for the matrix-logarithm $\boldsymbol{\Psi}$, where the matrices $\boldsymbol{\Omega}$ and $\mathbf{B}$ arise from decomposing the velocity gradient. In case of the differential models mentioned above, the model function $\mathbf{H}$ in Eq. (2.25) takes the form

$$\mathbf{H} \left(\boldsymbol{\Psi} \right) = \frac{1}{\Lambda} \left(\mathbf{I} - \alpha \exp \left(\boldsymbol{\Psi} \right) \cdot \left(\exp \left(-\boldsymbol{\Psi} \right) - \mathbf{I} \right) \right) \cdot \left(\exp \left(-\boldsymbol{\Psi} \right) - \mathbf{I} \right)$$

concerning the Giesekus model for $\alpha \in \left]0, 1\right]$, where the Oldroyd-B or UCM model are obtained for $\alpha = 0$ [47]. Following Ref. [48], the model function evolves to

$$\begin{aligned}\mathbf{H}_{\text{lin}}(\boldsymbol{\Psi}) &= \frac{1}{\Lambda}\left(1+\kappa\frac{\Lambda}{\eta_p}\left(\operatorname{tr}\left(\exp\left(\boldsymbol{\Psi}\right)-\mathbf{I}\right)\right)\right)\left(\exp\left(-\boldsymbol{\Psi}\right)-\mathbf{I}\right)\\ \mathbf{H}_{\text{exp}}(\boldsymbol{\Psi}) &= \frac{1}{\Lambda}\exp\left(\kappa\frac{\Lambda}{\eta_p}\left(\operatorname{tr}\left(\exp\left(\boldsymbol{\Psi}\right)-\mathbf{I}\right)\right)\right)\left(\exp\left(-\boldsymbol{\Psi}\right)-\mathbf{I}\right)\end{aligned}$$

in case of the PTT model concerning the linear or exponential version, respectively.

In Ref. [49], an alternative formulation of LCR is proposed, where no transformation of the constitutive equation is performed, but just a change regarding the stress variable. In detail, the conformation tensor is simply replaced by the matrix-exponential of $\boldsymbol{\Psi}$ in the corresponding material model, but the equation itself stays the same. Thus, possibly a simpler realisation of LCR is provided, since no decomposition of the velocity gradient needs to be applied. But, this alternative form provides in principle the same features as the original approach, that is the exponential growth of the stress variable is damped and the positive definiteness of the conformation tensor is guaranteed, even in numerical calculations.

A conceptually different, but also positivity preserving approach called the Contravariant Deformation Tensor formulation (CDT) is presented in Ref. [50], which consists of a multiplicative decomposition of the conformation tensor similar to a Cholesky-type decomposition. It turns out, that one deformation contribution is absent in the evolution equation of the contravariant deformation tensor compared to the constitutive equation arising with respect to the conformation tensor based on Eqs. (2.3) and (2.24). Thus, besides the ensured positive semi-definiteness of the conformation tensor, the High Weissenber Number Problem is avoided more or less by design, since a “natural” balance between convection and deformation is established in the considered evolution equation. But at the same time, the numerical effort is increased, since the contravariant deformation tensor is nonsymmetric, which requires to store one additional component compared to the extra-stress or conformation tensor. Note, that a similar approach is proposed regarding integral constitutive equation in Ref. [15], which will be discussed in Sec. 2.3.2.

2.2.3 The implemented numerical framework

The underlying numerical framework regarding two-dimensional differential viscoelastic flow models applied and extended in this work is validated and evaluated in Refs. [47] and particularly [51]. Thereby, the set of equations is treated in a monolithic fashion resulting in a (discrete) nonlinear system according to Eq. (2.20). Thus, the numerical handling considered here corresponds to the mixed or velocity-pressure-stress formulation [1]. In various approaches such as in Refs. [46, 39, 43, 44, 50], the flow problem is solved numerically in terms of operator splitting and/or pseudo time-stepping techniques to avoid the high computational costs of a fully coupled solution method, since only smaller subproblems have to be tackled successively. But, these approaches usually suffer from a relatively slow convergence speed, potentially consisting of large computation times to reach a steady-state solution. In contrast, solving for all unknowns simultaneously – in combination with a higher-order approximation of the flow variables as in the framework presented in Refs. [47, 51] – results in a robust, stable, accurate and efficient numerical approach, which is outlined in the following.

Discretisation techniques

For obtaining a high accuracy in approximating the velocity, stress and pressure fields, the corresponding set of unknowns $(\mathbf{u}, \mathbf{\Sigma}, p)$ is discretised via the approximation triple $Q_2/Q_2/P_1^{\text{disc}}$. In detail, the corresponding polynomial spaces are defined as

$$\begin{aligned} Q_2\left(T\right) &:= \left\{q \circ \Phi_T^{-1} \mid q \in \operatorname{span}\langle 1, x, y, xy, x^2, y^2, x^2y, y^2x, x^2y^2\rangle\right\} && \text{(2.26a)} \\ P_1^{\text{disc}}\left(T\right) &:= \operatorname{span}\langle 1, \tilde{x}, \tilde{y}\rangle && \text{(2.26b)} \end{aligned}$$

in case of two-dimensional configurations. Note, that the bilinear transformation $\Phi_T : \hat{T} \to T$ of the reference element $\hat{T} = [0,1]^2$ onto a physical element $T \in \mathcal{T}_h$ is taken into account when defining the biquadratic polynomial space Q_2 on an element $T \in \mathcal{T}_h$. In contrast, the space P_1^{disc} is defined by means of a local element-based coordinate system $\tilde{x}, \tilde{y}$ resulting from connecting the midpoints of opposite edges of T to maintain a higher-order convergence. In contrast, constructing P_1^{disc} in the same way as Q_2 in Eq. (2.26), that is by a composition with the bilinear mapping Φ_T, would reduce the order of convergence [27, 52]. However, the nine local degrees of freedom of the velocity and stress fields are the function values located in the corner and centre points of each element as well as in the midpoints of the element edges, establishing a globally continuous approximation. In contrast, the degrees of freedom regarding the pressure field are the function value plus the derivatives in both spacial dimensions located in the element midpoints, which leads to a piecewise continuous approximation allowing jumps across element edges. Finally, the velocity, stress and pressure solutions are approximated in the discrete spaces

$$\begin{aligned} \boldsymbol{V}_h &= \left\{\mathbf{v}_h \in \boldsymbol{V} \mid \mathbf{v}_{h|T} \in \left(Q_2\left(T\right)\right)^2 \ \forall T \in \mathcal{T}_h, \mathbf{v}_{h|\partial\Omega_h} = 0\right\} && \text{(2.27a)} \\ \mathsf{S}_h &= \left\{\mathbf{S}_h \in \mathsf{S} \mid \mathbf{S}_{h|T} \in \left(Q_2\left(T\right)\right)^{2\times 2}_{\text{sym}} \ \forall T \in \mathcal{T}_h\right\} && \text{(2.27b)} \\ Q_h &= \left\{q_h \in Q \mid q_{h|T} \in P_1\left(T\right) \ \forall T \in \mathcal{T}_h\right\} && \text{(2.27c)} \end{aligned}$$

being subspaces of $\boldsymbol{V} \times \mathsf{S} \times Q \subset \left(H_0^1\left(\Omega\right)\right)^2 \times \left(L^2\left(\Omega\right)\right)^{2\times 2}_{\text{sym}} \times L^2\left(\Omega\right)$ discussed in Sec. (2.2.1). Note, that in practice only three components of the discrete stress tensor need to be stored due to symmetry.

Obviously, the above choice of the discrete spaces provides a stable approximation of the velocity and pressure fields regarding the compatibility condition from Eq. (2.21), as the well-known Stokes pair is applied [27]. But, the additional inf-sup condition from Eq. (2.22) concerning the velocity-stress coupling is not satisfied, which at least for the non-solvent case $\eta_s = 0$ results in an unstable approximation. Consequently, a suitable stabilisation needs to be applied, which in this framework takes the form of the so-called Edge-Oriented FEM stabilisation (EOFEM, [47, 53]). Therefore, penalising terms $\mathcal{J}^u, \mathcal{J}^\sigma$ regarding jumps of the gradient of the (discrete) velocity and stress fields over element edges are added to the diagonal blocks of the (discrete) system from Eq. (2.20). In detail, the stabilising matrices read

$$\mathcal{J}^u_{ij} = \sum_{\text{edge } E} \gamma_u h_E \int_E [\nabla \boldsymbol{v}_j] : [\nabla \boldsymbol{v}_i] \, \mathrm{d}S \; \forall i,j \in \{1, \dots N_u\} \tag{2.28a}$$

$$\mathcal{J}^\sigma_{ij} = \sum_{\text{edge } E} \gamma_\sigma h_E^2 \int_E [\nabla \boldsymbol{\Phi}_j] : [\nabla \boldsymbol{\Phi}_i] \, \mathrm{d}S \; \forall i,j \in \{1, \dots N_\sigma\} \tag{2.28b}$$

where, $\gamma_u, \gamma_\sigma > 0$ denote mesh-independent stabilisation constants, h_E the length of the edge E and $[\boldsymbol{\xi}] = \boldsymbol{\xi}^+ - \boldsymbol{\xi}^-$ the jump over an element edge of a given quantity $\boldsymbol{\xi}$. Besides balancing the potentially unstable velocity-stress approximation, it is useful to add stabilisation in the present framework to the system anyways, since the hyperbolic character of the (differential) constitutive equation (2.16b) may cause unphysical numerical oscillations in the flow, which are not tackled here via SUPG or other techniques mentioned above. Moreover, the typical saddle-point structure as well as the original size of the problem is kept by making use of EOFEM stabilisation, which allows for applying numerical solvers especially designed for these kind of problems, for example the monolithic solution approach described below. In contrast, many alternative approaches providing a stable or stabilised approximation consist of several drawbacks like an increasing problem size in case of the stress approximation via subelements proposed in Ref. [24] or due to an additional variable in terms of the (D)EVSS from Ref. [40, 41]. However, the LCR of the constitutive equation will typically be considered in the numerical scheme applied in the following to reasonably treat the stress variables in case of higher relaxation times or Weissenberg numbers.

Solution methods

Applying the discretisation techniques described above results in a nonlinear system of similar structure as presented in Eq. (2.20), which is solved monolithically via Newton's method. Hence, a faster convergence speed compared to a simple fixed-point iteration is obtained, but the Jacobian matrix of the nonlinear problem or a suitable approximation needs to be provided. However, one nonlinear step of the iterative scheme reads

$$\mathrm{x}^{k+1} = \mathrm{x}^k - \omega^k \mathrm{A}^{-1}\left(\mathrm{x}^k\right) \mathrm{r}\left(\mathrm{x}^k\right) \tag{2.29a}$$

including a step width $\omega^k \in\,]0,1]$, which can be chosen adaptively to establish a global convergence of the nonlinear solver [47, 51]. Thereby, x denotes the vector of unknowns, r the residual and A the Jacobian matrix, which have the structure

$$\mathrm{x} = \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{\sigma} \\ \boldsymbol{p} \end{pmatrix}, \quad \mathrm{r}\left(\mathrm{x}\right) = \mathrm{r}\left(\boldsymbol{u}, \boldsymbol{\sigma}, \boldsymbol{p}\right) = \begin{pmatrix} \mathcal{L}\boldsymbol{u} + \mathcal{J}^u \boldsymbol{u} + \mathcal{C}\boldsymbol{\sigma} + \mathcal{B}\boldsymbol{p} - \mathrm{r}_u \\ \mathcal{D}\boldsymbol{u} + \mathcal{K}\left(\boldsymbol{u}\right)\boldsymbol{\sigma} + \mathcal{J}^\sigma \boldsymbol{\sigma} - \mathrm{r}_\sigma \\ \mathcal{B}^\top \boldsymbol{u} - \mathrm{r}_p \end{pmatrix} \tag{2.29b}$$

$$\mathrm{A}\left(\mathrm{x}\right) = \mathrm{A}\left(\boldsymbol{u}, \boldsymbol{\sigma}\right) = \begin{pmatrix} \mathcal{L} + \mathcal{J}^u & \mathcal{C} & \mathcal{B} \\ \mathcal{D} + \mathcal{K}_u\left(\boldsymbol{u}, \boldsymbol{\sigma}\right) & \mathcal{K}\left(\boldsymbol{u}\right) + \mathcal{J}^\sigma & 0 \\ \mathcal{B}^\top & 0 & 0 \end{pmatrix} \tag{2.29c}$$

based on the discrete operators from Eq. (2.19), where the stabilisation operators resulting from EOFEM are included as well. Furthermore, the subscript u of the operator $\mathcal{K}$ in the Jacobian matrix A denotes the corresponding derivative with respect to the velocity field. Note, that the extra-stress tensor is considered in Eq. (2.29) as numerical variable regarding the stress, although the LCR of the constitutive equations is mainly applied in the numerical investigations performed in Chpt. 4. Naturally, this causes slight changes in the discrete quantities presented above, since the matrix-logarithm of the conformation tensor represents the primal stress variable. But, the overall structure of the problem will stay the same.

In principle, the Jacobian matrix from Eq. (2.29c) can be computed analytically, which leads to discretising the Fréchet derivative of the continuous problem. By doing so, an exact Newton solver would be obtained, since numerical errors in terms of approximating the Jacobian matrix are avoided. But, a (complicated) derivation of the continuous problem is required, which has to be adapted once a different material model is chosen. Thus, computing the Jacobian matrix numerically via Finite Differencing is the conceptually much simpler procedure, which at the same time can be used in a "black box" style for any applied material model. But, the Newton scheme reduces to a quasi-Newton method, since the system matrix of the linear subproblems is only an approximated Jacobian. Nevertheless, in the present approach the entries of the (approximated) Jacobian are calculated by

$$\mathrm{A}_{ij}\left(\mathrm{x}\right) = \frac{1}{2\varepsilon}\left(\mathrm{r}_i\left(\mathrm{x}+\varepsilon \mathrm{e}_j\right) - \mathrm{r}_i\left(\mathrm{x}-\varepsilon \mathrm{e}_j\right)\right) \; \forall i,j \in \{1,\ldots,N_u+N_\sigma+N_p\} \tag{2.30}$$

where r_i refers to the row of the residual vector r from Eq. (2.29b) and e_j to the unit vector corresponding to degrees of freedom denoted by the indices i, j. Moreover, $\varepsilon > 0$ is a small perturbation parameter, which is typically chosen in the range of 10^{-8} and 10^{-6} in this work. Naturally, the choice of the perturbation parameter will affect only the convergence speed or the overall convergence behaviour, but not the solution itself.

Besides the calculation of the Jacobian, another important part of Newton's method is solving linear subproblems of the form

$$\mathrm{A}\left(\mathrm{x}^k\right)\bar{\mathrm{x}} = \mathrm{r}\left(\mathrm{x}^k\right) \tag{2.31}$$

for a correction $\bar{\mathrm{x}}$ of the current solution x^k. In principle, this can be done by means of a direct solver, although that might not be practical in many applications due to large problems sizes causing huge memory requirements and computation times. Consequently, iterative solution approaches offer a suitable alternative, where – aiming at the design of robust and efficient solvers – applying multigrid techniques for solving the above linear system is attractive, since a mesh-independent convergence behaviour can be obtained. Therefore, a (geometric) multigrid solver is designed based on a hierarchical sequence of meshes by specifying grid transfer operators plus a (direct) coarse grid solver as well as a smoothing scheme. Regarding the grid transfer, rather classical operators are applied in this context. For example, the restriction operator is realised by means of injection, where the values on the coarse mesh are set according to the corresponding values on the fine grid. In addition, biquadratic interpolation may be chosen concerning the prolongation. The general concept of multigrid solvers is described below following Refs. [54, 55, 56, 57], where the applied smoother plays an important role in the given framework.

As mentioned above, the goal is to solve the linear subproblem from Eq. (2.31) arising in terms of the Newton scheme. Starting from the linear system on the current (finest) mesh, a pre-smoothing step is performed, which may consist of few iterations of a basic iterative damped defect-correction technique of Richardson type. By doing so, the high-frequent components of the error between the current and actual solution are damped, while the low-frequent components are unaffected. Since these components appear to be of higher frequency on a coarser grid, the residual of the original linear problem with respect to the pre-smoothed solution is restricted to the next coarser mesh for further reduction of the error components. Therefore, the restricted residual is considered as the right-hand side of a linear problem on the coarser mesh level, which is solved for the correction of the solution of the original problem. The corresponding problem matrix is built in a suitable manner based on the original operator. After performing a few smoothing steps, the resulting residual is restricted further (including smoothing) until the coarsest mesh is reached and the coarse-grid problem is solved by a direct solver. Afterwards, optionally combined with few post-smoothing steps on each mesh level, the coarse-grid solution is prolonged back to the finest grid, where it is used to correct and thus update the solution of the original linear problem (2.31). In fact, the procedure described above corresponds to a so-called V-cycle, since the coarse-grid problem is only solved once, framed by restriction or prolongation operations from the finest to the coarsest level or vice versa, combined with applications of the smoother. As an alternative strategy, the coarse-grid solution might not be prolonged the full way up to the finest level, but intermediately can be restricted (together with smoothing) to the coarsest level again, before being prolonged to the finest mesh. Thus, the obtained W- or F-cycle includes at least one additional solution of the coarse-grid problem compared to the V-cycle.

A crucial component of the multigrid solver is the smoother, which is chosen to be of Vanka-type in the present framework, since it provides a suitable treatment of the zero blocks in the (approximated) Jacobian matrix from Eq. (2.29c), typically occurring for saddle-point problems. In fact, a standard global Jacobi or Gauß-Seidel smoother is not applicable in this context, as parts of these zero blocks would need to be inverted. In contrast, an update of the solution is computed within the applied Vanka smoother successively on small subdomains of the global mesh, where a local counterpart of the Jacobian matrix as well as the residual is extracted from the global versions. Actually, these quantities are built according to the degrees of freedom on one element or a small patch of elements leading to a local expression of the global system. This resulting small local system is solved “exactly” to update the unknowns corresponding to the specific element or patch, where solutions on already treated elements or patches are included in the local residual. Finally, a global update of the solution is obtained by collecting the local contributions. Consequently, the applied smoother can be interpreted as a block-wise Gauß-Seidel procedure.

Transient configurations

So far, discretisation and solution techniques regarding differential viscoelastic flow models are described for considering the direct steady-state formulation. In case of a time-dependent configuration, the basic ingredients of the numerical framework will be the same, extended by contributions arising from the discretisation of the governing equations with respect to time [47], which is illustrated below. Based on the original time-dependent flow model consisting of Eqs. (2.1), (2.2) and (2.3), applying an implicit time discretisation by means of the Fractional-Step-θ scheme together with the spatially discretised operators from Eq. (2.19) leads to the time- and space-discrete system

$$\begin{aligned}
\frac{\boldsymbol{u}^{l+1}-\boldsymbol{u}^{l}}{\Delta t} + \theta\left(\mathcal{L}\boldsymbol{u}^{l+1} + \mathcal{C}\boldsymbol{\sigma}^{l+1}\right) + \dots & \\
\dots + (1-\theta)\left(\mathcal{L}\boldsymbol{u}^{l} + \mathcal{C}\boldsymbol{\sigma}^{l}\right) + \mathcal{B}\boldsymbol{p}^{l+1} &= \mathbf{r}_u \qquad (2.32a)\\
\frac{\boldsymbol{\sigma}^{l+1}-\boldsymbol{\sigma}^{l}}{\Delta t} + \theta\left(\mathcal{K}\left(\boldsymbol{u}^{l+1}\right)\boldsymbol{\sigma}^{l+1} + \mathcal{D}\boldsymbol{u}^{l+1}\right)\dots & \\
+ (1-\theta)\left(\mathcal{K}\left(\boldsymbol{u}^{l}\right)\boldsymbol{\sigma}^{l} + \mathcal{D}\boldsymbol{u}^{l}\right) &= \mathbf{r}_\sigma \qquad (2.32b)\\
\mathcal{B}^\top \boldsymbol{u}^{l+1} &= \mathbf{r}_p \qquad (2.32c)
\end{aligned}$$

For simplicity, an equidistant time step width $\Delta t = t_l - t_{l-1}$ is chosen for computing the discrete unknowns $\left(\boldsymbol{u}^l, \boldsymbol{\sigma}^l, \boldsymbol{p}^l\right)$ at the discrete time instants $t_l, l > 1$, but in principle also an adaptive choice of Δt is possible. The initial solutions $(\boldsymbol{u}^0, \boldsymbol{\sigma}^0)$ may be set to zero or according to results of previous simulations, possibly interpolated from a coarser to the current mesh. In case of choosing $\theta = \frac{1}{2}$, the second-order Crank-Nicolson scheme is obtained from Eq. (2.32), which is applicable also in the case of neglected inertia and/or a vanishing solvent contribution due to the monolithic treatment of the set of equations. Thus, the design of special decoupling techniques leading to second-order transient schemes as done in Refs. [43, 44] may be avoided. As the main difference to the direct steady-state approach, the Newton scheme from Eq. (2.29) has to be applied in every time step in case of an instationary simulation. The corresponding global matrix from Eq. (2.20) is modified, such that a contribution of the form $\frac{1}{\Delta t}\mathcal{M}$ is added to the diagonal blocks regarding the velocity and stress fields. Note, that the actual mass matrix $\mathcal{M}$ depends on the specific spatial discretisation of the velocity and stress fields, respectively, that is the choice of the basis functions from Eq. (2.18). Furthermore, specific blocks of the matrix according to Eq. (2.32) are weighted with θ, while the explicit contributions enter the right-hand side of the nonlinear problem. Considering the linear subproblems (2.31) in terms of transient simulations, the system matrix is stabilised by the additional contributions $\frac{1}{\Delta t}\mathcal{M}$, which might improve the behaviour of the linear (multigrid) solver.

2.3 The Deformation Fields Method regarding integral material laws

As illustrated above, differential constitutive equations can be implemented within the Finite Element context more or less straightforward. But, numerical simulations involving such material models suffer from several issues, for instance regarding the prediction of the behaviour of realistic viscoelastic materials, because the required multi-mode approach or the HWNP may lead to serious difficulties. Consequently, an implementation of integral constitutive laws is of intensified interest, since an improved modelling approach is obtained from a rheological point of view and in addition, some of the numerical drawbacks of differential models are avoided. Therefore, the Deformation Fields Method is applied for treating integral constitutive equations together with the Stokes equations in a Finite Element framework. Naturally, a couple of major challenges arise in this context as well, which is illustrated in Secs. 2.3.1 and 2.3.2 by describing the basic developments of the Deformation Fields Method proposed in Refs. [12, 13, 14, 15]. At the same time, suitable ingredients are detected to extend the numerical framework of differential models presented in Sec. 2.2 or Refs. [47, 51] to considering integral models, which is outlined in Sec. 2.3.3.

2.3.1 The initial approach

As a starting point, the temporal as well as spatial discretisation of integral viscoelastic models is discussed by means of the initial approach to the Deformation Fields Method proposed in Ref. [13], which is transferred to the two-dimensional case described in Sec. 2.1. A key component of such integral models is to consider the deformation history of the flow by means of a stress integral. In doing so, the "memory" of the fluid is represented by the function m, which is taken to be exponentially decaying for increasing age of the deformation. Hence, the effect of deformations applied at a certain time instant in the deformation history decreases for marching forward in time, which is why a so-called cut-off time $s_c \in \left]-\infty, t\right[$ is introduced in Ref. [13], below which contributions to the deformation history are negligible small. Thus, an approximation of the stress integral from Eq. (2.9) according to

$$\boldsymbol{\Sigma}\left(t\right) \approx \int_{s_c}^{t} m\left(t-t'\right)\left[\phi_1\left(\operatorname{tr}\left(\mathbf{B}_{t'}\right)\right)\mathbf{B}_{t'}\left(t\right)+\phi_2\left(\operatorname{tr}\left(\mathbf{B}_{t'}\right)\right)\mathbf{B}_{t'}^{-1}\left(t\right)\right]\,\mathrm{d}t' \qquad (2.33)$$

is obtained. Obviously, it is not possible to consider the deformation history on a continuous level within a numerical framework, which is why the above stress integral needs to be discretised with respect to the reference time t'. Besides the memory function m, empirical functions related to the Finger tensor $\mathbf{B}_{t'}$ occur in the stress integral (2.33), which consequently have to be discretised with respect to the reference time as well to obtain an approximation of the whole deformation history. For simplicity, $\phi_2 \equiv 0$ is considered in the following, since the methodology applied for discretising $\phi_1\left(\operatorname{tr}\left(\mathbf{B}_{t'}\right)\right)\mathbf{B}_{t'} =: \mathbf{G}\left(\mathbf{B}_{t'}\left(t\right)\right)$ can be directly transferred to the case of a non-vanishing function ϕ_2. In the initial approach to the Deformation Fields Method proposed in Ref. [13], a fixed number of discrete times $s_i \in \left[0, \infty\right[, i = 0, \ldots, N$, is introduced covering the whole deformation history, such that $t - s_0 = t$, that is the current time instant, and $t - s_N = s_c$. Thereby, an optimal approximation of the deformation history should be obtained by appropriately choosing the discrete times s_i, which may be related to the memory of the material. Since the memory function m is exponentially decaying, it consists of a large gradient with respect to reference times t' close to t. Consequently, for small ages – or reference times close to the current time instant – the applied deformations are assumed to have a large influence on the current state of the flow, while deformations arising from large ages contribute less. Hence, it seems reasonable to apply a small age step width close to the current time instant t, while the step width can be increased for larger ages $t - t'$, for example by means of integer multiples, since corresponding numerical errors are damped due to the shape of the memory function [13]. For discretising the stress integral with respect to the reference time t', the included empirical function is approximated by interpolating the Deformation Fields assigned to the discrete reference times $t - s_i$. In detail, the resulting approximation

$$\mathbf{G}\left(\mathbf{B}_{t-s}\left(t\right)\right) \approx \sum_{i=0}^{N} \mathbf{G}\left[\mathbf{B}_{t-s_i}\left(t\right)\right] \xi_i\left(s\right) \tag{2.34}$$

is established by means of suitable one-dimensional basis functions ξ_i depending on $s \in [0, \infty[$, for example piecewise linear functions satisfying $\xi_i\left(s_j\right) = \delta_{ij}$, where $\xi_N\left(s_c\right) = 1$ is set due to the definition of the cut-off time s_c. Consequently, an approximation

$$\mathbf{\Sigma}\left(t\right) \approx \sum_{i=0}^{N} W_i \mathbf{G}\left[\mathbf{B}_{t-s_i}\left(t\right)\right], \quad W_i = \int_0^\infty m\left(s\right) \xi_i\left(s\right) \,\mathrm{d}s \tag{2.35}$$

of the stress integral from Eq. (2.33) is obtained, where the corresponding weights W_i need to be computed only initially during pre-processing.

As a next step, the Stokes equations (2.1) as well as the evolution equation (2.8), coupled by means of the approximated stress integral (2.35), need to be discretised regarding the time variables t and s, respectively. For simplicity, it is assumed that both equations are considered according to the same time scale, which is why the same time step width may be applied during discretisation. But naturally, the evolution equation of the Finger tensors might be solved on a different time scale than the Stokes equations, which would require suitable interpolation techniques to transfer the velocity field calculated from the Stokes equations (2.1) to the evolution equation (2.8). However, the evolution equation (2.8) of the Finger tensor is discretised in Ref. [13] with respect to time via the explicit Euler scheme giving

$$\frac{\mathbf{B}_{t_l-s_i}^{l+1} - \mathbf{B}_{t_l-s_i}^{l}}{\Delta t} + \tilde{\mathcal{K}}\left(\mathbf{u}^l\right) \mathbf{B}_{t_l-s_i}^{l} = 0 \tag{2.36}$$

concerning all s_i, $i = 0, \ldots, N$. In doing so, small time steps for obtaining accurate results are required, which makes this the weak spot of the initial approach to the Deformation Fields Method. Regarding the temporal discretisation of the Stokes equations (2.1), similar techniques as presented in Sec. 2.2 can be applied, for example the Backward Euler method as in Ref. [13] resulting for $\theta = 1$ in Eq. (2.32a). Note, that inertia are neglected in the initial approach proposed in Ref. [13] (and also in further developed approaches from Refs. [14, 15]), which implies $\rho = 0$ and thus leads to a vanishing derivative of the velocity field with respect to time, although the flow is still time-dependent. Obviously, this results in an ill-posed continuous or discrete problem when considering the Stokes equations (2.1) for the non-solvent case $\eta_s = 0$ including the divergence of the stress tensor as a right-hand side. Consequently, the Discrete Elastic-Viscous Stress Splitting (DEVSS, [41]) is applied in Ref. [13] similar to differential models, while also other techniques are applicable as proposed in Ref. [15] or in Sec. 2.4. Concerning the spatial dimension, the evolution equation (2.8) regarding the fields $\mathbf{B}_{t'}$ is discretised in Ref. [13] by a discontinuous Galerkin scheme including piecewise bilinear polynomials. However, the (spatially discretised) operator $\tilde{\mathcal{K}}$ from Eq. (2.36) may be defined similarly to the corresponding operator in case of differential models from Eq. (2.19d). In addition, the velocity and pressure variables are approximated by means of the well-known LBB-stable Stokes pair Q_2/P_1^{disc}.

It becomes clear, that the resulting discrete system of equations needs to be solved via operator splitting, since the full set of Deformation Fields assigned to the discrete reference times $t_l - s_i$, $i = 0, \ldots, N-1$, is updated by performing one time step in Eq. (2.36) based on the velocity field $\mathbf{u}^l$. In doing so, the fields $\mathbf{B}^{l+1}_{t_l-s_0}, \ldots, \mathbf{B}^{l+1}_{t_l-s_{N-1}}$ are obtained, and at the same time the oldest field $\mathbf{B}^{l}_{t^l-s_N}$ is deleted. In addition, the updated fields are shifted regarding the reference times according to $t_l - s_i \to t_{l+1} - s_{i+1}$ for $i = 0, \ldots, N-1$ and a new field $\mathbf{B}^{l+1}_{t_{l+1}}$ assigned to $t_{l+1} - s_0 = t_{l+1}$ is created, which keeps the number of $N+1$ fields constant. By means of the resulting fields $\mathbf{B}^{l+1}_{t_{l+1}-s_0}, \ldots, \mathbf{B}^{l+1}_{t_{l+1}-s_N}$, the approximated stress integral $\mathbf{\Sigma}^{l+1}$ is computed from Eq. (2.35), which enters the right-hand side of the Stokes equations (2.32a) and (2.32c) for computing an updated velocity and pressure solution. Note, that this straightforward procedure of updating, creating and deleting fields can be implemented only, if the age discretisation consists of a constant step width, that is $s_{i+1} - s_i = \Delta t \ \forall i$. But this makes sense only regarding (discrete) reference times $t - s_i$ close to the current time t, while the age step width should be increased for discrete times $s_i \gg 0$. In this case, it is not practical to delete the oldest field assigned to s_N after a new field is created at the new time instant t_{l+1}: Since $s_N = s_{N-1} + k\Delta t$ for a fixed $k \in \mathbb{N}$, the field $\mathbf{B}^{l}_{t_l-s_{N-1}}$ needs to be evolved for at least k time steps, until $t_l - s_{N-1} \to t_{l+k} - s_N = s_c$ forces the field to being deleted. Instead, some intermediate field is discarded once shifting $t_l - s_i \to t_{l+j} - s_{i+1}$ for specific $i \in \{0, \ldots, N-1\}$ and $j \in \mathbb{N}$, where $s_{i+1} = s_i + j\Delta t$, would assign the discrete field to the next discrete time, which is in fact still blocked by the (not yet reassigned) field $\mathbf{B}_{t_l-s_{i+1}}$ [13].

In Ref. [13] it is found, that considering about 100 discrete Deformation Fields leads to quite accurate and stable simulation results, although a time discretisation error remains since the fields do not reach a steady state, even when the flow itself becomes stationary. Furthermore, initially small defects occurring in the Deformation Fields are intensified by the persisting evolution. In the long term, this will affect the accuracy in calculating the stress tensor, since the numerical oscillations might not be damped by the exponentially decaying memory function anymore. In this regard, the cut-off time s_c may be shifted closer to the current time t leading to a shorter evolution time of the (oscillating) fields. Hence, the numerical perturbations could be limited, since the age of the Deformation Fields and thus the build-up of oscillations is reduced [13]. But naturally, a reduced cut-off time also decreases the accuracy of the approximated stress integral from Eq. (2.33), again affecting the quality of the numerical results. Consequently, a reasonable choice of the time step for evolving the Deformation Fields in combination with a suitable discretisation of the reference time and an appropriate setting of the cut-off time is of interest to obtain stable and accurate numerical results. Obviously, a higher-order scheme approximating the evolution equation (2.8) with respect to time is an important point for improving the initial approach to the Deformation Fields Method, as already mentioned in Ref. [13].

In summary, a fixed number of (discrete) fields $\mathbf{B}_{t-s_i}$, $i = 0, \ldots, N$, is stored within the initial approach to the Deformation Fields Method from Ref. [13], which are convected and deformed over time. Consequently, having the velocity field available at the time instant t from the Stokes equations Eq. (2.1), the Deformation Fields need to be updated by performing one time step in Eq. (2.36) regarding each of the fields $\mathbf{B}_{t-s_i}$. Additionally, a new field $\mathbf{B}_{t+\Delta t}$ is initialised and an existing field is deleted to keep the number of fields constant. Moreover, the initially computed weights W_i, which are linked to the specific discrete time s_i, need to be reassigned to different fields when initialising, shifting and and deleting fields. Furthermore, the Deformation Fields do not become steady even when the flow is steady, which might lead to time discretisation errors potentially causing a loss of the positive (semi-)definiteness of the fields. In addition, instabilities are observed associated with the choice of the cut-off time, which are thus not present in case of an equivalent differential model formulation, but affecting the stability and accuracy of the numerical method [13]. Hence, several developments of the Deformation Fields Method have been proposed based on this initial approach concerning the issues discussed above.

2.3.2 Further developments

In a first improved method presented in Ref. [14], the age $\tau := t - t'$ is introduced as an additional time scale leading to re-defined Deformation Fields of the form $\mathbf{B}(t, \tau, \mathbf{x}) := \mathbf{B}_{t-\tau}(t, \mathbf{x})$. Recall, that a Deformation Field $\mathbf{B}_{t'}(t)$ in the initial approach contains information about the deformation history from the absolute reference time t' to the current time t. In contrast, the transformed fields $\mathbf{B}(t, \tau)$ characterise the deformation of a fluid particle during a fixed time frame $[t - \tau, t]$ back from the current time. Thus, the deformation history of the fluid is considered within a fixed age relative to the current time t by introducing τ as additional time scale. In contrast, an absolute time interval is taken into account from the cut-off time s_c to the current time in the original approach. However, based on the re-defined fields the evolution equation of the Finger tensor changes to

$$\begin{aligned} \frac{\partial}{\partial t}\mathbf{B}(t,\tau) + \frac{\partial}{\partial \tau}\mathbf{B}(t,\tau) + \mathbf{u}(t) \cdot \nabla \mathbf{B}(t,\tau) \ldots & \\ \cdots - \nabla \mathbf{u}(t)^{\top} \cdot \mathbf{B}(t,\tau) - \mathbf{B}(t,\tau) \cdot \nabla \mathbf{u}(t) &= 0 \end{aligned} \qquad (2.37)$$

together with the initial and “boundary” condition $\mathbf{B}(0, \tau) = \mathbf{B}(t, 0) = \mathbf{I}$. Thus, the flow is at rest at the initial time $t = 0$ and the fields at the current time instant t are initialised with age $\tau = 0$ in a state of equilibrium. Naturally, also the stress integral from Eq. (2.9) is affected by the newly introduced time scale, since a corresponding transformation of the integral gives

$$\boldsymbol{\Sigma}(t) \approx \int_0^{\tau_c} m(\tau) \left[\phi_1 (\mathrm{tr}(\mathbf{B}(t,\tau)))\, \mathbf{B}(t,\tau) + \phi_2 (\mathrm{tr}(\mathbf{B}(t,\tau)))\, \mathbf{B}^{-1}(t,\tau)\right] \mathrm{d}\tau \qquad (2.38)$$

Here, a suitable cut-off age τ_c is directly incorporated, because contributions to the stress integral again are assumed to be negligible small for large ages $\tau > \tau_c$.

Compared to the evolution equation (2.8) from the initial approach in Ref. [13], the modified evolution equation (2.37) consists of an additional term, that is the derivative of the Deformation Fields with respect to the age variable. By discretising the evolution equation concerning the re-defined Deformation Fields $\mathbf{B}(t,\tau)$ with respect to both, the actual time t as well as the age scale τ, a global coupling of the discrete fields during evolution is introduced due to the additional derivative [14], which is absent in the original method. Consequently, this modified Deformation Fields Method does not seem to provide an improved formulation at first glance. But, there are some significant advantages obtained in case of assigning the fields to the age τ instead of the reference time t'. In fact, the fields become steady when the complete flow reaches a stationary state, that means for a velocity field indepenent of t. Therefore, the fields $\mathbf{B}(t,\tau)$ can be written as $\mathbf{B}(t,\tau) = \mathbf{B}_{t-\tau}(t) = \mathbf{B}_{t-\tau}(\tau + t') = \mathbf{B}_{t'}(\tau)$ since the age is defined as $\tau := t - t'$ for fixed t'. Thus, the evolution equation (2.37) gives

$$\begin{aligned} 0 &= \frac{\partial}{\partial t}\mathbf{B}(t,\tau) + \frac{\partial}{\partial \tau}\mathbf{B}_{t'}(\tau) + \mathbf{u}\cdot\nabla\mathbf{B}_{t'}(\tau)\dots \\ &\quad \dots - \nabla\mathbf{u}^{\top}\cdot\mathbf{B}_{t'}(\tau) - \mathbf{B}_{t'}(\tau)\cdot\nabla\mathbf{u} \\ &= \frac{\partial}{\partial t}\mathbf{B}(t,\tau) + \overset{\triangledown}{\mathbf{B}}_{t'}(\tau) \end{aligned} \tag{2.39}$$

where the upper-convected time derivative of $\mathbf{B}(t,\tau) = \mathbf{B}_{t'}(\tau)$ is regarded with respect to the age variable τ instead of t by taking into account, that $\mathbf{u}$ is independent of any time variable. Hence, the fields $\mathbf{B}(t,\tau)$ indeed reach a stationary state regarding the actual time t, since the upper-convected time derivative of the Deformation Fields in the evolution equation (2.39) vanishes according to Eq. (2.8). But naturally, the fields are evolved with respect to the age scale. Another feature of the modified approach is provided by offering the possibility to discretise the age τ independently of the actual time step. In contrast, the discretisation of the reference time t' in the initial approach is in some way linked to the actual time step applied in the Stokes equations (2.1) and/or the evolution equation (2.8) of the Finger tensor. Additionally, there is no need to create or delete fields due to considering a “fixed” part $[t-\tau, t]$ of the deformation history, furthermore avoiding the need of re-assigning the weights in the resulting approximated stress integral.

Accordingly, more stable and accurate results are obtained concerning the numerical behaviour of the modified approach to the Deformation Fields Method compared to the original approach proposed in Ref. [13]. A significant improvement is established in Ref. [14] by evolving the discrete Deformation Fields forward in time by a second-order Adams-Bashforth scheme as well as introducing the age as an additional time scale. Thus, numerical oscillations arising from the time discretisation are damped and do not keep growing even for stationary flows, since also the Deformation Fields reach a stationary state, although being persistently evolved with respect to the age scale. As a result, the simulation quality regarding integral models taking into account the improved version of the Deformation Fields Method is comparable to differential models. This does not hold for the original approach, since numerical instabilities occur already for lower We than in the differential case [13]. Furthermore, the corresponding age discretisation can be performed independently of the actual time step, in contrast to discretising the reference time in the original approach. But, slight drawbacks arise in terms of the improved approach due to the additional age derivative, since a smaller time step needs to be chosen regarding the stability in the age-stepping. In addition, a coupling of the discrete fields is established, which affects the corresponding updating strategy.

So far, no detailed remark is made concerning the discretisation of the age scale, which needs to consist of discrete age points concentrated close to $\tau = 0$, while at the same time the age mesh should get coarser for larger ages. In Ref. [14], a stretched mesh regarding τ is considered, where the step widths $\Delta\tau_j = \tau_{j+1} - \tau_j$, $j = 0, \ldots, N-1$, are linked via $\Delta\tau_{j+1} = \gamma\Delta\tau_j$. Here, the factor γ is determined from a one-dimensional auxiliary problem adapting the (decaying) memory of the flow, once N, $\Delta\tau_0$ and τ_c are specified. A similar strategy for specifying the discrete ages τ_j is presented in Ref. [12], where the age scale is defined by a mapping $\tau : [0,1] \to [0,\infty]$ as $\tau(r) = -\beta \ln(1-r)$ for a parameter β related to the relaxation time Λ. By doing so, a uniform mesh in r-space automatically generates a mesh in τ-space, where points are concentrated at $\tau = 0$ and thus small age steps are applied for small ages. In fact, the resulting discretisation in τ resembles the memory of the viscoelastic fluid characterised by the (exponentially decaying) memory function, which is realised by taking into account the inverse mapping $r(\tau)$. Furthermore, the fields $\mathbf{B}^*(t,\tau) := \frac{\eta_p}{\Lambda^2}\exp\left(-\frac{\tau}{\Lambda}\right)\mathbf{B}(t,\tau)$ are considered as Deformation Fields in Ref. [12], which results in $\mathbf{B}^*(t,\tau) \to 0$ for $t \to \infty$, since also $\tau := t - t' \to \infty$. Note, that the fields $\mathbf{B}$ grow quadratically in time for simple flow configurations as outlined in Sec. 3.2, which is why the modified fields $\mathbf{B}^*$ behave much more predictable than the original fields $\mathbf{B}$. The corresponding evolution equation

$$\begin{aligned} \frac{\partial}{\partial t}\mathbf{B}^*(t,r) + \frac{1-r}{\beta}\frac{\partial}{\partial r}\mathbf{B}^*(t,r) + \mathbf{u}(t)\cdot\nabla\mathbf{B}^*(t,r)\ldots \\ \cdots - \nabla\mathbf{u}(t)^\top \cdot \mathbf{B}^*(t,r) - \mathbf{B}^*(t,r)\cdot\nabla\mathbf{u}(t) + \frac{1}{\Lambda}\mathbf{B}^*(t,r) &= 0 \end{aligned} \tag{2.40}$$

is obtained by inserting the fields $\mathbf{B}^*(t,\tau)$ into Eq. (2.37) and realising $\tau = \tau(r)$. In doing so, an additional contribution arises in the evolution equation (2.40), which has the form of the model function from Eq. (2.4) in case of the differential Oldroyd-B or UCM model. However, the full integral viscoelastic model proposed in Ref. [12] is obtained from coupling the evolution equation (2.40) with the Stokes equations (2.1) and the transformed version

$$\begin{aligned} \mathbf{\Sigma}(t) &= \int_0^\infty \phi_1(\mathrm{tr}(\mathbf{B}(t,\tau)))\,\mathbf{B}^*(t,\tau)\,\mathrm{d}\tau \\ &= \int_0^1 \phi_1(\mathrm{tr}(\mathbf{B}(t,r)))\,\mathbf{B}^*(t,r)\,\frac{\beta}{1-r}\,\mathrm{d}r \end{aligned} \tag{2.41}$$

of the stress integral from Eq. (2.38) for $\phi_2 \equiv 0$. Note, that the memory function is absorbed into the fields $\mathbf{B}^*$, while the damping function still corresponds to the original fields $\mathbf{B}$. An approximation of the above integral with respect to r can be established by the Composite-Trapezoidal Rule giving

$$\mathbf{\Sigma}(t) \approx \sum_{j=1}^{N} W_j \phi_1 \left(I_1 \left(t, r_j\right), I_2 \left(t, r_j\right)\right) \mathbf{B}^* \left(t, r_j\right) \frac{\beta}{1 - r_j}$$

for equidistant points $r_j = \frac{j-1}{N}$. In contrast to previous approaches, the cut-off time or age may be introduced implicitly, for example by setting the contribution corresponding to $r = 1$ or $j = N$ to zero in the above numerical integration together with $W_1 = \frac{1}{2N}$ and $W_j = \frac{1}{N}$ for $1 < j < N - 1$ [12].

Following Ref. [12], the numerical results obtained by evolving $\mathbf{B}^*$ instead of $\mathbf{B}$ tend to be of slightly higher accuracy compared to the approach proposed in Ref. [14], since errors caused in terms of the evolution equation are damped by the exponentially decaying part in the fields $\mathbf{B}^*$. Moreover, a further improvement – especially regarding computational effort – is achieved by updating only *one* field every time step, but *all* fields only every 5, 10 or 20 time steps. Thus, the computational costs are roughly of the same magnitude as compared to (single-mode) differential models, where only one stress tensor is considered, while the Deformation Fields Method involves about 100 Finger tensors. In fact, stability and accuracy of the numerical simulations are improved as well. However, this is just found by performing numerical experiments and no theoretical explanation is given. In addition, the above updating strategy is feasible only when computing to the steady-state, since the time-dependent behaviour of the Deformation Fields is not captured correctly [12].

So far, some major drawbacks of the initial version of the Deformation Fields Method proposed in Ref. [13] are removed and an improved method is obtained by following Refs. [12, 14], especially regarding the discretisation of the Finger tensors as well as the stress integral with respect to the reference time or age. But, still no improvement is established concerning the preservation of the positive (semi-)definiteness of the Deformation Fields $\mathbf{B}$ during numerical computations. When differential constitutive equations are considered, the positive definiteness of the underlying conformation tensor may be guaranteed by considering the corresponding matrix-logarithm (LCR, [36, 37, 46]) or the Contravariant Deformation Tensor (CDT, [50]) as primal flow variable. In principle, LCR might also be applicable with respect to the Deformation Fields, but it is unclear at the moment, whether these fields show an exponential growth that would need to be balanced as well, which is one of the key features of introducing LCR. In fact, considering the logarithm of the Deformation Fields might be an unnecessary effort. An alternative approach to guarantee the positive (semi-)definiteness of the Deformation Fields is presented in Ref. [15], where the deformation tensor $\mathbf{F}_{t'}(t)$ is evolved over time instead of the Finger tensor $\mathbf{B}_{t'}(t)$. Similar to the improved version of the Deformation Fields Method proposed in Ref. [14], introducing the age $\tau := t - t'$ as new time variable yields the evolution equation

$$\frac{\partial}{\partial t}\mathbf{F}(t, \tau) + \frac{\partial}{\partial \tau}\mathbf{F}(t, \tau) + \mathbf{u}(t) \cdot \nabla \mathbf{F}(t, \tau) - \nabla \mathbf{u}(t)^{\top} \cdot \mathbf{F}(t, \tau) = 0 \qquad (2.42)$$

regarding the deformation tensor $\mathbf{F}(t, \tau) := \mathbf{F}_{t-\tau}(t)$. The above evolution equation is supplemented with the initial condition $\mathbf{F}(0, \tau) = \mathbf{I}$ as well as the boundary condition $\mathbf{F}(t, 0) = \mathbf{I}$ with respect to the τ-axis. Based on the deformation tensor, the Finger tensor is calculated by $\mathbf{B} = \mathbf{F} \cdot \mathbf{F}^{\top}$ and is thus guaranteed to be positive semi-definite. Indeed, an improved behaviour of the resulting numerical approach is observed, since stability properties similar to applying LCR in the context of differential models are obtained. In Ref. [15], this phenomenon is traced back not only to the guaranteed positive semi-definiteness of the Finger tensor, but also to the evolution equation (2.42) being reduced by one deformation term compared to evolving the Finger tensor $\mathbf{B}$ according to Eq. (2.37), which is done similarly in case of the CDT formulation for differential models [50]. Note, that an additional component of the deformation tensor compared to the Finger tensor has to be stored in the numerical method, since $\mathbf{F}$ is not symmetric. In fact, it needs to be estimated, whether the resulting additional numerical effort pays off regarding an improved accuracy in the simulations.

2.3.3 Implementation of stationary integral models

In the following, the above numerical treatment of integral constitutive equations is summarised by illustrating an extension of the framework regarding differential models presented in Refs. [47, 51] or Sec. 2.2.3 towards integral material laws. Therefore, the stationary version of the integral viscoelastic fluid flow model is derived to be able to compute direct steady-state solutions according to the differential case. Concerning time-dependent configurations, the same techniques as presented in Refs. [12, 13, 14, 15] can be taken into account.

Apparently, the corresponding Stokes part of the integral model is of the same form as in Eq. (2.16a) and (2.16c), but the remaining equations are not formulated straightfoward regarding the direct steady-state. On the one hand, the stress tensor is computed by means of the time-dependent integral expression from Eq. (2.9) and on the other hand, the Finger tensors do not reach a steady state when evolved according to Eq. (2.8), even in case of a stationary velocity field [13]. Recall, that the fields indeed become stationary with respect to the actual time when introducing the age as additional time scale [14]. But, considering the age scale is redundant in the context of stationary flows, since the actually resulting evolution equation basically stays the same as can be realised from Eq. (2.39) by interchanging t and τ. Thus, a suitable way of dealing with the accumulating time errors of the non-stationary Deformation Fields mentioned in Ref. [13] also needs to be incorporated in case of the stationary integral model derived below. Based on the initial (time-dependent) approach to the Deformation Fields Method proposed in Ref. [13], in each time step a new Deformation Field $\mathbf{B}_{t'}$ is created and evolved depending on the velocity field $\mathbf{u}(s)$. Assuming a steady-state solution, the velocity field stays the same for all times s, especially in the evolution equation (2.8). Thus, all fields $\mathbf{B}_{t'}$ would give the same solution when evolved over the infinite time interval $s \in [t', \infty[$, since the initial condition $\mathbf{B}_{t'}(t') = \mathbf{I}$ holds for all t'. Consequently, it is sufficient to consider only one single Deformation Field $\mathbf{B}(s)$ in a "stationary" integral model, and the time variable in the stress integral (2.9) can be transformed according to $s := t - t'$. Hence, a quasi steady-state formulation of the integral viscoelastic model regarding the quantities $(\mathbf{u}, \boldsymbol{\Sigma}, p)$ is obtained, reading

$$
\begin{aligned}
0 &= -2\eta_s \nabla \cdot \mathbf{D}(\mathbf{u}) - \nabla \cdot \mathbf{\Sigma} + \nabla p && \text{(2.43a)}\\
\mathbf{\Sigma} &= \int_0^\infty m(s) \left[\phi_1 (\operatorname{tr}(\mathbf{B}(s)))\, \mathbf{B}(s) + \phi_2 (\operatorname{tr}(\mathbf{B}(s)))\, \mathbf{B}^{-1}(s)\right] \mathrm{d}s && \text{(2.43b)}\\
\nabla \cdot \mathbf{u} &= 0 && \text{(2.43c)}
\end{aligned}
$$

complemented by the evolution equation

$$
\frac{\partial}{\partial s}\mathbf{B}(s) + \mathbf{u} \cdot \nabla \mathbf{B}(s) - \nabla \mathbf{u}^\top \cdot \mathbf{B}(s) - \mathbf{B}(s) \cdot \nabla \mathbf{u} = \mathbf{0} \qquad \text{(2.43d)}
$$

which is solved in $s \in [0, \infty[$ for the Finger tensor $\mathbf{B}$ together with the initial condition $\mathbf{B}(0) = \mathbf{I}$. Note, that the stress integral from Eq. (2.43b) indeed takes a finite value independent of the time s, since the field $\mathbf{B}$ grows quadratically in time at least for simple flow configurations as outlined in Sec. 3.2, while the memory function is taken to be exponentially decaying.

Naturally, the numerical framework applied in the context of differential constitutive equations has to be extended by suitable components to cover (stationary) integral models by means of the DFM. Therefore, mainly the discretisation of the stress integral (2.43b) as well as the evolution equation (2.43d) with respect to the time scale s needs to be taken into account. In this context, a meaningful basis for extending the framework of differential models is proposed in Ref. [12], since approximating the stress integral via a simple (Composite-)Trapezoidal Rule can be implemented more or less straightforward. Moreover, a reasonable way to generate the discrete times $s_i \in [0, \infty[$ is illustrated in Ref. [12] as well, since the corresponding procedure is related to the shape of the memory function based on the mapping $\tau(r)$. Furthermore, the deformation tensor $\mathbf{F}$ could be considered within the integral material law, which gives the Finger tensor via $\mathbf{B} = \mathbf{F} \cdot \mathbf{F}^\top$ guaranteeing the positive (semi-)definiteness of the Deformation Fields [15]. At the same time, the technique of absorbing the memory function into the fields – leading to the quantity $\mathbf{B}^*$ – would need to be adjusted, since the stress integral from Eq. (2.43b) is written only implicitly depending on the field $\mathbf{F}$. But, including the exponentially decaying memory function into the evolution of the field seems reasonable in order to obtain damped time-discretisation errors, which occur even in a stationary setting [13]. However, the field $\mathbf{B}$ or $\mathbf{F}$ needs to be evolved over a wide time interval, in detail until contributions to the stress integral are negligible small, starting all over again from the initial time $s = 0$, once an updated (stationary) velocity field is available. Thus, the evolution equation of the field $\mathbf{F}$ or $\mathbf{B}$ should be discretised regarding the time s by applying an explicit scheme of higher order to obtain a fast and efficient, but also accurate time-stepping procedure.

In addition to the extensions mentioned above, the basic components required for implementing a suitable numerical treatment of integral models may be adopted from the existing differential framework. In detail, the spatial discretisation of the velocity and pressure fields as well as the Finger or deformation tensor, which represents the primal stress variable in case of integral models, could be implemented by means of the triple $Q_2/P_1^{\mathrm{disc}}/Q_2$ together with EOFEM stabilisation. Again, the latter may be applied anyways regarding the hyperbolic character of the evolution equation of the Deformation Fields. In contrast to differential models, the resulting discrete nonlinear system – consisting of the Stokes equations, the stress integral plus the evolution equation of the Finger or deformation tensor – must be solved via operator splitting techniques. By doing so, the contribution of the stress tensor in the momentum equation enters the right-hand side as a force term, while the discrete version of the evolution equation (2.43d) is solved with respect to $\mathbf{B}$ for a given discrete version of the velocity field $\mathbf{u}$. Following Refs. [13, 14, 15], the integral viscoelastic flow model might be treated in combination with the DEVSS to stabilise the numerical approach. At the same time, a well-posed problem as well as a useful decoupled solution approach can be obtained even in the non-solvent case, which can be established in similar way by means of the novel approach proposed in the following section.

2.4 The Tensor Diffusion approach

As described in Secs. 2.2 and 2.3, simulating viscoelastic fluid flows involves a number of crucial steps for designing a stable, robust, efficient and accurate numerical framework. Thereby, suitable discretisation and solution techniques need to be applied for properly handle the approximation of the flow variables, the particular type of constitutive equation, its hyperbolic character or the High Weissenberg Number Problem. A key component with respect to the properties of the underlying mathematical problem is represented by the amount of solvent contribution to the total or Cauchy stress tensor from Eq. (2.2). Among others, this quantity provides an essential regularisation of the subproblem constituted by the balance of mass and momentum. For example, the additional LBB condition (2.22) with respect to choosing a stable velocity-stress approximation can be neglected for $\eta_s > 0$ [35]. In addition, several challenges arise for $\eta_s = 0$ also concerning the solution procedure regarding the discrete nonlinear problems, which result from differential or integral flow models. Recall, that this is one of the reasons for introducing the (D)EVSS [40, 41] in the so-called non-solvent case, where the corresponding diffusive or elliptic operator vanishes from the momentum equation of the Stokes problem. To further illustrate these numerical difficulties, direct steady-state solution approaches are considered below, where the presence of a solvent viscosity plays an important role.

Based on the procedure outlined in Sec. 2.2, the discrete problem formulation corresponding to a single-mode stationary differential viscoelastic flow model reads

$$\begin{pmatrix} \mathcal{J}^u & \mathcal{B} & \mathcal{C} \\ \mathcal{B}^\top & 0 & 0 \\ \mathcal{D} & 0 & \mathcal{K}(\boldsymbol{u}) + \mathcal{J}^\sigma \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \\ \boldsymbol{\sigma} \end{pmatrix} = \begin{pmatrix} \mathbf{r}_u \\ \mathbf{r}_p \\ \mathbf{r}_\sigma \end{pmatrix} \tag{2.44}$$

after re-sorting the order of the unknowns to emphasise the Stokes subproblem. Apparently, only the operator resulting from the EOFEM stabilisation proposed in Eq. (2.28a) remains in the corresponding upper-left block matrix – in case it is applied at all. Thus, the Stokes part of the (differential) viscoelastic flow model yields a singular (sub-)problem in case of a vanishing solvent viscosity, since it basically consists of a zero diagonal when omitting stabilising terms. Consequently, several difficulties and challenges arise in the numerical approach, for example with respect to solving the linear subproblems resulting within Newton's method described in

Eq. (2.29). Obviously, no preconditioners involving the diagonal part of the global (Jacobian) matrix are applicable in terms of typical Krylov-space methods due to the vanishing diagonal blocks. In similar manner, applying multigrid techniques for solving these linear systems does not allow for the use of diagonal smoothers such as Jacobi or Gauß-Seidel defect correction techniques. Therefore, Vanka-like smoothers, which perform preconditioning locally on each element [54, 55, 56, 57], are proposed in Sec. 2.2.3 for appropriately treating the zero blocks typically occurring in saddle-point problems. But unfortunately, at least in the non-solvent case a non-robust behaviour of the multigrid solver involving these kind of smoothers is observed (see Sec. 4.2.2). At this stage, the only reliable procedure for successfully performing numerical simulations of non-solvent viscoelastic fluid flows within the framework presented in Sec. 2.2.3 is obtained by applying Newton's method in combination with a direct solver for tackling the arising linear systems. But, this limits the simulations to the range of small or moderate problem sizes, only. In this regard, operator splitting techniques might be taken into account for solving the nonlinear system from Eq. (2.44) in a decoupled fashion. Thereby, an update solution $\left(\boldsymbol{u}^{k+1}, \boldsymbol{p}^{k+1}\right)$ regarding the velocity and pressure fields, respectively, is intended to be calculated from the subproblem

$$\mathcal{B}\boldsymbol{p}^{k+1} + \mathcal{C}\boldsymbol{\sigma}^{k} = \mathbf{r}_u, \qquad \mathcal{B}^\top \boldsymbol{u}^{k+1} = \mathbf{r}_p \tag{2.45}$$

for a given stress solution $\boldsymbol{\sigma}^k$. Apparently, no reasonable solution – especially regarding the velocity field – can be computed from Eq. (2.45) to be applied in a next step of determining an updated stress field. Naturally, the same difficulties are obtained in case of non-solvent integral viscoelastic flow models, where the corresponding discrete problem formulation is not solvable in a monolithic fashion at all.

However, the simulation of realistic viscoelastic fluid flows, which typically do not include a solvent viscosity, provides further numerical challenges in case of both types of constitutive equations. Recall, that multiple modes need to be considered in terms of the differential material modelling, where an additional stress tensor occurs as separate numerical unknown for each mode or relaxation time. Hence, the size of the numerical problem increases accordingly, which could lead to a significant growth of the computational effort. This is also the case for integral models, although the multiple relaxation times just enter the memory function in the stress integral. But, considering multiple and thus probably larger relaxation times will increase the computational effort in calculating the stress integral. In detail, the memory function consists of a slower decrease for larger Λ, which is why a wider age interval needs to be regarded in the numerical integration of the extra-stress tensor as well as the Deformation Fields. Thus, the numerical effort will increase either way for simulating realistic viscoelastic fluids, since the problem size or computation time is increased due to the involved broad relaxation time spectrum. Moreover, also the HWNP or the hyperbolic character of the constitutive equation need to be treated appropriately in a corresponding numerical framework, as (very) high relaxation times are provided by means of such fluids.

Overall, the numerical challenges highlighted above are caused to a certain extend by taking into account realistic viscoelastic materials, among others consisting of a vanishing solvent viscosity. Consequently, some serious issues are caused by considering the stationary non-solvent Stokes equations as part of differential as well as integral viscoelastic flow models. Hence, corresponding Finite Element simulations suffer from difficulties regarding discretisation as well as solution techniques concerning the set of equations

$$-\nabla \cdot \mathbf{\Sigma} + \nabla p = \mathbf{0}, \qquad \nabla \cdot \mathbf{u} = 0 \tag{2.46}$$

As outlined above, in principle unstable or even non-solvable configurations are obtained, mainly due to the absent diffusive operator regarding the velocity field in the momentum equation from Eq. (2.46). In this regard, the novel Tensor Diffusion approach is introduced in the following, which is intended to improve numerical simulations of viscoelastic fluid flows with special emphasis on the non-solvent case. The underlying assumption of this approach, motivated by the findings presented in Sec. 4.1.1, is the existence of a decomposition of the extra-stress tensor according to

$$\mathbf{\Sigma} = \mathbf{M} \cdot \mathbf{D}(\mathbf{u}) \tag{2.47}$$

where the second-order tensor $\mathbf{M} = \mathbf{M}(\mathbf{u}, \nabla\mathbf{u})$ is called the Diffusion Tensor. Apparently, the above decomposition of the extra-stress tensor is of similar style as the solvent amount to the Cauchy stress tensor from Eq. (2.2) referred to by the scalar-valued viscosity η_s. Accordingly, inserting the stress decomposition (2.47) into Eq. (2.46) introduces a diffusive operator in the resulting the so-called Tensor Stokes problem. But, in general a nonsymmetric tensor-valued vsicosity is obtained in the corresponding momentum equation due to the shape of $\mathbf{M}$ (see Sec. 4.1.1 for details). It turns out, that an improved numerical behaviour is obtained for considering the symmetric version

$$\mathbf{\Sigma} = \frac{1}{2}\left(\mathbf{\Sigma} + \mathbf{\Sigma}^\top\right) = \frac{1}{2}\left(\mathbf{M} \cdot \mathbf{D}(\mathbf{u}) + \mathbf{D}(\mathbf{u}) \cdot \mathbf{M}^\top\right) \tag{2.48}$$

of the stress decomposition from Eq. (2.47). Note, that the original stress decompostion from Eq. (2.47) also satisfies the symmetrised version due to the symmetry of the extra-stress tensor $\mathbf{\Sigma}$. However, the resulting symmetrised Tensor Stokes problem reads

$$-\frac{1}{2}\nabla \cdot \left(\mathbf{M} \cdot \mathbf{D}(\mathbf{u}) + \mathbf{D}(\mathbf{u}) \cdot \mathbf{M}^\top\right) + \nabla p = \mathbf{0}, \qquad \nabla \cdot \mathbf{u} = 0 \tag{2.49}$$

where a diffusive operator is provided by means of the Diffusion Tensor $\mathbf{M}$. Note, that a non-vanishing solvent viscosity in the momentum equation from Eq. (2.49) may be absorbed into the Tensor Stokes operator by considering $\tilde{\mathbf{M}} = 2\eta_s\mathbf{I} + \mathbf{M}$ instead of $\mathbf{M}$. Moreover, the Diffusion Tensor can be regarded as $\mathbf{M} \in \mathbb{R}^{2\times2}$ or $\mathbb{R}^{3\times3}$ both for two-dimensional configurations, since the (symmetric) extra-stress and strain-rate tensors may be interpreted as tensor- or vector-valued quantities, respectively. The latter is obtained by making use of the so-called Voigt notation, which for instance provides additional degrees of freedom for determining the Diffusion Tensor in terms of the four-field formulation discussed below in Eq. (2.51), potentially leading to improved mathematical properties of $\mathbf{M}$. In the following, the potential benefits of simulating viscoelastic fluid flows by means of the Tensor

Diffusion approach are outlined. Therefore, several versions of the Tensor Stokes problem are considered, that is Eq. (2.49) possibly supplemented by a constitutive equation of differential or integral type. In doing so, major challenges and difficulties described in Secs. 2.2 and 2.3 regarding numerical simulations of viscoelastic fluid flows might be weakened or even removed.

Ideally it is assumed, that the Diffusion Tensor $\mathbf{M}$ is known or given depending on the velocity field $\mathbf{u}$ and/or its gradient according to an actual differential or integral viscoelastic flow problem. Hence, a well-posed problem regarding the velocity and pressure variables is derived from Eq. (2.46), that is even in the non-solvent case, when a diffusive operator acting on the velocity field is re-coupled into the momentum equation according to Eq. (2.49). Based on that, the nonlinear velocity and pressure solutions, originally characterised by the fully coupled problems (2.16) or (2.43), can be computed by simply solving the Tensor Stokes problem (2.49), which is a problem in $(\mathbf{u}, p)$ only. The according stress solution is just computed in post-processing fashion based on the velocity field calculated from Eq. (2.49). Hence, the stress variable and a corresponding stable or stabilised velocity-stress approximation do not need to be taken into account in the numerical framework. Thus, the original problem is simplified by expressing the complex rheology of the fluid, which is described originally by the constitutive equation, in terms of a tensor-valued viscosity function denoted by the Diffusion Tensor $\mathbf{M}$. Furthermore, a robust, efficient, accurate and stable numerical scheme can be used for solving the Tensor Stokes problem (2.49), since solution techniques especially designed for (generalised) Stokes problems are applicable in this context. In doing so, the overall numerical effort for computing the solution corresponding to the original viscoelastic flow problems from Eq. (2.16) or (2.43) would be reduced significantly. Similar holds regarding the simulation of realistic viscoelastic fluids, since multiple modes are not considered explicitly in the flow model. Moreover, among other things the hyperbolic character of the constitutive equation, the High Weissenberg Number Problem or a proper treatment of integral constitutive equations would not need to be (explicitly) taken into account. In addition, converting the full viscoelastic differential or integral flow model to a pure Tensor Stokes problem gives rise to substantial improvements regarding transient simulations, since capturing the time-dependent behaviour of viscoelastic fluid flows would only require a time-stepping scheme regarding a generalised Stokes problem instead of a fully coupled nonlinear system.

Unfortunately, no explicit or analytical modelling of the Diffusion Tensor $\mathbf{M}$ for general or complex flow configurations is available (yet). Hence, a suitable way of determining such a quantitiy numerically needs to be established based on the original (differential or integral) viscoelastic model. A straightforward implemenation for calculating the Diffusion Tensor is obtained by complementing the (differential) non-solvent steady-state viscoelastic model by an additional algebraic equation regarding $\mathbf{M}$. In doing so, the set of equations

$$\begin{aligned}
-\nabla \cdot \mathbf{\Sigma} + \nabla p &= \mathbf{0} && \text{(2.50a)}\\
\mathbf{u} \cdot \nabla \mathbf{\Sigma} - \nabla \mathbf{u}^\top \cdot \mathbf{\Sigma} - \mathbf{\Sigma} \cdot \nabla \mathbf{u} + \mathbf{Z}\left(\Lambda, \eta_p, \mathbf{\Sigma}\right) &= 2\frac{\eta_p}{\Lambda}\mathbf{D}\left(\mathbf{u}\right) && \text{(2.50b)}\\
\mathbf{M} \cdot \mathbf{D}\left(\mathbf{u}\right) - \mathbf{\Sigma} &= 0 && \text{(2.50c)}\\
\nabla \cdot \mathbf{u} &= 0 && \text{(2.50d)}
\end{aligned}$$

is obtained based on Eq. (2.16). Obviously, the discrete solution corresponding to $(\mathbf{u}, \boldsymbol{\Sigma}, p)$ resulting from Eq. (2.50) is not affected by the Diffusion Tensor $\mathbf{M}$, which is computed from Eq. (2.50c) in post-processing fashion only. That changes, when the (symmetrised) stress decomposition from Eq. (2.48) is inserted into the momentum equation, leading to

$$\begin{aligned}
-\frac{1}{2}\nabla \cdot \left(\mathbf{M} \cdot \mathbf{D}\left(\mathbf{u}\right) + \mathbf{D}\left(\mathbf{u}\right) \cdot \mathbf{M}^\top\right) + \nabla p &= \mathbf{0} && \text{(2.51a)}\\
\mathbf{u} \cdot \nabla \boldsymbol{\Sigma} - \nabla \mathbf{u}^\top \cdot \boldsymbol{\Sigma} - \boldsymbol{\Sigma} \cdot \nabla \mathbf{u} + \mathbf{Z}\left(\Lambda, \eta_p, \boldsymbol{\Sigma}\right) &= 2\frac{\eta_p}{\Lambda}\mathbf{D}\left(\mathbf{u}\right) && \text{(2.51b)}\\
\mathbf{M} \cdot \mathbf{D}\left(\mathbf{u}\right) - \boldsymbol{\Sigma} &= 0 && \text{(2.51c)}\\
\nabla \cdot \mathbf{u} &= 0 && \text{(2.51d)}
\end{aligned}$$

Naturally, both problem formulations are equivalent on the continuous level, but differ after applying a suitable discretisation procedure. Thus, the (discrete) velocity, stress and pressure solutions indeed are coupled with the Diffusion Tensor $\mathbf{M}$ by means of the Tensor Stokes operator in the momentum equation (2.51a). Similar to the DEVSS described in Eq. (2.23) or Ref. [41], the problem size is increased by introducing an additional variable, which is intended to provide a diffusive contribution in the momentum equation. But in terms of the Tensor Diffusion approach, the stress variable is replaced by means of the additional variable instead of adding a diffusive contribution to the left- as well as right-hand side of Eq. (2.16a). Moreover, the velocity coupling is introduced via the Diffusion Tensor related to the nature of the actual flow, since it is linked in a meaningful way to the stress tensor itself. In contrast, a Newtonian-like quantity is considered in terms of the DEVSS, where the diffusive contribution is defined somehow artificially by the parameter α in Eq. (2.23a). Thus, a potentially improved numerical scheme could be obtained based on the above Tensor Stokes problem. Naturally, the Diffusion Tensor may be inserted into (integral) viscoelastic models similarly to the DEVSS, which leads to an additional step in the corresponding operator splitting approach due to calculating the Diffusion Tensor from Eq. (2.47). Corresponding preliminary studies are performed in Sec. 4.2.4 of this work. Note, that it is not possible to calculate the Diffusion Tensor based on the symmetric decomposition from Eq. (2.48), as the resulting linear system with respect to the components of $\mathbf{M}$ is under-determined. Thus, the symmetrised operator occuring in Eq. (2.51a) actually needs to be built from Eq. (2.51c) within numerical simulations.

However, the four-field formulation of the Tensor Stokes problem presented in Eq. (2.51) is considered in Chpt. 4 in terms of numerical investigations for validating and evaluating the Tensor Diffusion approach. Therefore, the Diffusion Tensor $\mathbf{M}$ is approximated within the Finite Element framework described in Sec. 2.2.3 by means of element-wise constant polynomials. By defining corresponding basis functions $\left(\mathbf{N}_1, \dots, \mathbf{N}_{N_\mu}\right)$ based on the triangulation of the discrete computational domain Ω_h, an approximate solution of the Diffusion Tensor is defined as

$$\mathbf{M}_h = \sum_{i=1}^{N_\mu} \mu_i \mathbf{N}_i,$$

where the degrees of freedom located in the element midpoints are collected in $\boldsymbol{\mu} = \left(\mu_i\right)_{i=1}^{N_\mu}$. Furthermore, the additional discrete operators

$$
\begin{aligned}
\mathcal{T}_{ij}\left(\mu_j\right) &= \int_\Omega \frac{1}{2}\left(\mu_j \mathbf{N}_j \cdot \mathbf{D}\left(\boldsymbol{v}_j\right) + \mathbf{D}\left(\boldsymbol{v}_j\right) \cdot \left(\mu_j \mathbf{N}_j\right)^\top\right) : \mathbf{D}\left(\boldsymbol{v}_i\right) \,\mathrm{d}\mathbf{x} \\
&\quad \forall i, j \in \{1, \ldots, N_u\} \qquad (2.52a) \\
\mathcal{N}_{ij}\left(u_j\right) &= \int_\Omega \left(\mathbf{N}_j \cdot \mathbf{D}\left(u_j \boldsymbol{v}_j\right)\right) : \mathbf{N}_i \,\mathrm{d}\mathbf{x} \; \forall i, j \in \{1, \ldots, N_\mu\} \qquad (2.52b) \\
\mathcal{S}_{ij} &= \int_\Omega \mathbf{\Phi}_j : \mathbf{N}_i \,\mathrm{d}\mathbf{x} \; \forall i \in \{1, \ldots, N_\mu\},\; j \in \{1, \ldots, N_\sigma\} \qquad (2.52c)
\end{aligned}
$$

arise from considering the four-field formulation in Eq. (2.51) compared to the original problem in Eq. (2.16). Again, the nonlinear dependencies of the degrees of freedom regarding the Diffusion Tensor or velocity field in Eq. (2.52) are considered in a suitable way in the actual implementation. At first glance, the proposed lower-order approximation of $\mathbf{M}$ is somewhat surprising. In principle, a higher-order approximation is expected to be applied for obtaining an accurate reconstruction of the extra-stress tensor based on the Diffusion Tensor according to Eq. (2.47). But as indicated by the numerical investigations performed in Sec. 4.2.1, jumps or discontinuities of the Diffusion Tensor seem to occur naturally in the flow field, for example along the symmetry axis of a contraction. Thus, it indeed makes sense to approximate $\mathbf{M}$ by means of discontinuous Finite Elements like element-wise constant polynomials, which in principle allow jumps of the function values between neighbouring elements. Furthermore, no derivatives actually acting on $\mathbf{M}$ need to be considered at this fundamental stage of the Tensor Diffusion approach, which is why no higher-order approximation is necessarily required here. However, the numerical results presented in Chpt. 4 – especially concerning a viscoelastic contraction flow – show, that the jumps in the Diffusion Tensor along the symmetry axis are not resolved properly or are too large to provide a stable and converging numerical scheme. In detail, small values of the strain-rate tensor $\mathbf{D}\left(\mathbf{u}\right)$ are obtained in the problematic section of the computational domain, while the extra-stress tensor $\mathbf{\Sigma}$ is rather large, which causes large values or even a singular behaviour of the Diffusion Tensor (see Sec. 4.2.1). Hence, the Diffusion Tensor may be calculated numerically in an improved way by penalising the derivatives of $\mathbf{M}$ to avoid too drastic changes in the numerical variable. This can be implemented by determining the Diffusion Tensor by means of

$$
j\left(\mathbf{M}\right) := \frac{1}{2}\left(\|\mathbf{M} \cdot \mathbf{D}\left(\mathbf{u}\right) - \mathbf{\Sigma}\|_2^2 + \varepsilon \|\nabla \mathbf{M}\|_2^2\right) \to \min \qquad (2.53)
$$

for a penalisation parameter $\varepsilon > 0$, which would lead to an additional Laplacian or artificial diffusion operator acting on $\mathbf{M}$ in the strong formulation of the problem, that is Eq. (2.51c). Obviously, this is not feasible with the lower-order approximation of the Diffusion Tensor proposed above, which is why an alternative damping or stabilising technique is introduced. In detail, EOFEM stabilisation [47, 53] is applied with respect to $\mathbf{M}$, which results in a (discrete) penalisation term reading

$$
\mathcal{J}_{ij}^{\mu} = \sum_{\mathrm{edge} E} \gamma_\mu \int_E \left[\mathbf{N}_j\right] : \left[\mathbf{N}_i\right] \,\mathrm{d}S \; \forall i, j \in \{1, \ldots, N_\mu\} \qquad (2.54)
$$

for a parameter $\gamma_\mu > 0$. In contrast to the corresponding stabilisation regarding the velocity and stress fields presented in Eq. (2.28), the above operator affects the function values of $\mathbf{M}$ instead of its gradient. By doing so, a stabilisation technique is obtained actually suiting the approximation of the Diffusion Tensor as an element-wise constant quantity. Note, that EOFEM stabilisation with respect to the function values as presented in Ref. [53] may be interpreted in some sense as a bilinear form corresponding to a Laplacian. In detail, the jump terms can be regarded as the difference of the function values normal to the specific edge. Including a scaling with h_E^{-1}, that is the length of the edge, in the weighting factor of the above stabilisation operator results in an approximation of the gradient in Finite Difference fashion. Thus, basically the weak form of a Laplacian is obtained from a stabilisation operator similar to Eq. (2.54) for considering $\gamma_\mu h_E^{-1}$ instead of γ_μ only, which can be confirmed by means of numerical experiments. But in terms of the Tensor Diffusion approach, stabilising $\mathbf{M}$ by means of the discrete operator from Eq. (2.54) seems to provide more useful results than the scaled version from Ref. [53], especially in the context of the contraction flow discussed in Sec. 4.2. Nevertheless, an operator similar to the artificial diffusion arising from Eq. (2.53) is established with respect to the Diffusion Tensor by means of the EOFEM stabilisation proposed above, even for $\mathbf{M}$ being approximated in Q_0. However, based on the discrete operators from Eq. (2.19) and (2.52), the discrete nonlinear systems arising from the four-field formulations (2.50) and (2.51), that is in particular for the non-solvent case, are given as

$$\begin{pmatrix} \mathcal{J}^u & \mathcal{C} & 0 & \mathcal{B} \\ \mathcal{D} & \mathcal{K}(\boldsymbol{u}) + \mathcal{J}^\sigma & 0 & 0 \\ 0 & \mathcal{S} & \mathcal{N}(\boldsymbol{u}) + \mathcal{J}^\mu & 0 \\ \mathcal{B}^\top & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{\sigma} \\ \boldsymbol{\mu} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \mathbf{r}_u \\ \mathbf{r}_\sigma \\ \mathbf{r}_\mu \\ \mathbf{r}_p \end{pmatrix} \tag{2.55a}$$

$$\begin{pmatrix} \mathcal{T}(\boldsymbol{\mu}) + \mathcal{J}^u & 0 & 0 & \mathcal{B} \\ \mathcal{D} & \mathcal{K}(\boldsymbol{u}) + \mathcal{J}^\sigma & 0 & 0 \\ 0 & \mathcal{S} & \mathcal{N}(\boldsymbol{u}) + \mathcal{J}^\mu & 0 \\ \mathcal{B}^\top & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{\sigma} \\ \boldsymbol{\mu} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \mathbf{r}_u \\ \mathbf{r}_\sigma \\ \mathbf{r}_\mu \\ \mathbf{r}_p \end{pmatrix} \tag{2.55b}$$

which are considered in Chpt. 4 in terms of two-dimensional Finite Element simulations.

Based on the above four-field formulations, further potential benefits of the Tensor Diffusion approach compared to state-of-the-art numerical techniques for simulating viscoelastic fluid flows are outlined below. But beforehand, an advanced approach for calculating the Diffusion Tensor in terms of differential viscoelastic models is illustrated concerning future research work. Therefore, the extra-stress tensor $\boldsymbol{\Sigma}$ is replaced by the decomposition from Eq. (2.47) not only in the momentum equation (2.16a), but also in the constitutive equation (2.16b). Unfortunately, higher-order derivatives of the velocity field $\mathbf{u}$ occur in the resulting convective term similar to the EVSS proposed in Ref. [40]. Accordingly, one might introduce an additional variable for treating these derivatives numerically in a suitable way, which would again lead to an increased problem size as in case of the (D)EVSS or the four-field formulation from Eq. (2.51). Instead, the convective term may be transformed by explicitly taking into account, that the velocity field is divergence-free. Therefore, the identity

$$\begin{aligned}\nabla \cdot (\mathbf{P} \otimes \mathbf{v}) &= \frac{\partial}{\partial x_l} \left((P_{ij} \mathbf{e}_i \otimes \mathbf{e}_j) \otimes (v_k \mathbf{e}_k) \right) \cdot \mathbf{e}_l = (P_{ij} v_k)_{,l} \, \mathbf{e}_i \otimes \mathbf{e}_j \delta_{kl} \\ &= (P_{ij,k} v_k + P_{ij} v_{k,k}) \, \mathbf{e}_i \otimes \mathbf{e}_j \\ &= \mathbf{v} \cdot \nabla \mathbf{P} + (\nabla \cdot \mathbf{v}) \, \mathbf{P} = \mathbf{v} \cdot \nabla \mathbf{P} \qquad (2.56)\end{aligned}$$

can be derived for a second-order tensor field $\mathbf{P}$ and a solenoidal vector field $\mathbf{v}$ via usual tensor calculus following Einstein‘s summation convention. Here, “$\otimes$” denotes the dyadic product of two vectors, $\mathbf{e}_\iota$ for $\iota \in \{1, \dots, d\}$ refer to the (cartesian) basis vectors of $\mathbb{R}^d$ and the subscript $(, \iota)$ indicates the derivative with respect to the corresponding spatial dimension. By setting $\mathbf{P} = \mathbf{M} \cdot \mathbf{D}(\mathbf{u})$ and $\mathbf{v} = \mathbf{u}$, the three-field formulation

$$\begin{aligned}-\frac{1}{2} \nabla \cdot \left(\mathbf{M} \cdot \mathbf{D}(\mathbf{u}) + \mathbf{D}(\mathbf{u}) \cdot \mathbf{M}^\top \right) + \nabla p &= 0 && (2.57\text{a}) \\ \nabla \cdot \left[(\mathbf{M} \cdot \mathbf{D}(\mathbf{u})) \otimes \mathbf{u} \right] & \; \dots && \\ \dots - \nabla \mathbf{u}^\top \cdot (\mathbf{M} \cdot \mathbf{D}(\mathbf{u})) - (\mathbf{M} \cdot \mathbf{D}(\mathbf{u})) \cdot \nabla \mathbf{u} & \; \dots && \\ \dots + \mathbf{Z}(\Lambda, \eta_p, \mathbf{M} \cdot \mathbf{D}(\mathbf{u})) &= 2 \frac{\eta_p}{\Lambda} \mathbf{D}(\mathbf{u}) && (2.57\text{b}) \\ \nabla \cdot \mathbf{u} &= 0 && (2.57\text{c})\end{aligned}$$

is obtained from the stationary differential viscoelastic flow model from Eq. (2.16), where the extra-stress tensor $\mathbf{\Sigma}$ is replaced completely by $\mathbf{M} \cdot \mathbf{D}(\mathbf{u})$. Recall, that $\mathbf{\Sigma}$ is discretised within the original method by means of Q_2, which is why the Diffusion Tensor $\mathbf{M}$ can be approximated in the same way. But, the corresponding test and trial functions are set to be nonsymmetric – just as the Diffusion Tensor itself – in contrast to the basis functions of the discrete stress space. Accordingly, the full Diffusion Tensor needs to be stored within the numerical framework, while the number of actually considered components of the symmetric extra-stress tensor $\mathbf{\Sigma}$ is less. However, considering the transformed conservative form of the constitutive equation with respect to the Diffusion Tensor $\mathbf{M}$ presented in Eq. (2.57b) offers the possibility to shift the derivative in the convective term in the weak formulation to the test function avoiding higher-order derivatives of the velocity field. In addition, no derivatives with respect to $\mathbf{M}$ occur in the corresponding continuous weak formulation of Eq. (2.57). Thus, probably a lower regularity of the function space regarding the Diffusion Tensor is required compared to the space S discussed in Sec. 2.2.1. The discrete operators arising from Eq. (2.57b) in the weak sense read

$$\begin{aligned}\mathcal{P}_{ij}(u_j) &= \int_\Omega (\mathbf{N}_j \cdot \mathbf{D}(u_j \boldsymbol{v}_j)) : (u_j \boldsymbol{v}_j \cdot \nabla \mathbf{N}_i) \, \mathrm{d}\mathbf{x} \\ & \quad \forall i, j \in \{1, \dots, N_\mu\} && (2.58\text{a}) \\ \mathcal{Q}_{ij}(u_j) &= \int_\Omega \Big[-\nabla (u_j \boldsymbol{v}_j)^\top \cdot \mathbf{N}_j \cdot \mathbf{D}(u_j \boldsymbol{v}_j) - \mathbf{N}_j \cdot \mathbf{D}(u_j \boldsymbol{v}_j) \cdot \nabla (u_j \boldsymbol{v}_j) \dots \\ & \quad \dots + \mathbf{Z}(\Lambda, \eta_p, \mathbf{N}_j \cdot \mathbf{D}(u_j \boldsymbol{v}_j)) \Big] : \mathbf{N}_i \, \mathrm{d}\mathbf{x} \\ & \quad \forall i, j \in \{1, \dots, N_\mu\} && (2.58\text{b}) \\ \mathcal{E}_{ij} &= \int_\Omega 2 \frac{\eta_p}{\Lambda} \mathbf{D}(\boldsymbol{v}_j) : \mathbf{N}_i \, \mathrm{d}\mathbf{x}, \ \forall i \in \{1, \dots, N_\mu\}, j \in \{1, \dots N_u\} && (2.58\text{c})\end{aligned}$$

which occur in the discrete nonlinear system

$$\begin{pmatrix} \mathcal{T}\left(\boldsymbol{\mu}\right)+\mathcal{J}^{u} & 0 & \mathcal{B} \\ \mathcal{E} & \mathcal{P}\left(\boldsymbol{u}\right)+\mathcal{Q}\left(\boldsymbol{u}\right)+\mathcal{J}^{\mu} & 0 \\ \mathcal{B}^{\top} & 0 & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{\mu} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \mathbf{r}_{u} \\ \mathbf{r}_{\mu} \\ \mathbf{r}_{p} \end{pmatrix} \tag{2.59}$$

resulting from Eq. (2.57) including certain discrete operators introduced in Sec. 2.2 as well as the Tensor Stokes operator from Eq. (2.52a). Note, that the boundary integral arising from partially integrating the convective contribution in the constitutive equation, which is similar to the one given in Ref. [42] for the original EVSS, is considered in the corresponding right-hand side $\mathbf{r}_{\mu}$. In fact, the "PDE approach" proposed above, where the Diffusion Tensor is determined by means of a partial differential equation, might lead to an improved numerical scheme compared to the "algebraic approach" described in Eq. (2.51). Characterising the Diffusion Tensor by means of Eq. (2.57b) establishes a global coupling of the corresponding degrees of freedom due to the derivatives acting on $\mathbf{M}$, which could lead to a more sophisticated calculation. Furthermore, the problem size of the original approach is kept in case of the problem formulation from Eq. (2.57) by actually replacing the stress variable by the Diffusion Tensor, which leads to a reduced computational effort compared to four-field formulations. Thus, the PDE approach is quite attractive in case of differential constitutive laws, whereas it is not transferable to integral models – at least not that easily.

In this section, three formulations of the Tensor Stokes problem are proposed, that is the pure $(\mathbf{u}, p)$-problem from Eq. (2.49), the four-field formulation from Eq. (2.51) as well as the three-field formulation from Eq. (2.57). Since the conceptual advantages of the pure Tensor Stokes problem are already highlighted above, potential benefits particularly concerning the corresponding three- and four-field formulations are taken into account in the following. As a key feature, inserting the Diffusion Tensor into the Stokes part of the viscoelastic flow model provides a diffusive operator regarding the velocity field in the corresponding momentum equation. Thus, regularising effects may be expected regarding discretisation and solution techniques for simulating non-solvent viscoelastic fluid flows, while a comparable operator or velocity coupling is absent in case of the according original (unstabilised) problem from Eq. (2.44). For example, the non-vanishing Tensor Stokes operator in the systems from Eq. (2.55b) or (2.59) offers the possibility to apply diagonal smoothers or preconditioners within multigrid or Krylov-space methods. Furthermore, Vanka-like smoothers or the multigrid solver in general have the potential to show a stabilised behaviour, since these techniques are designed for linear systems arising from elliptic problems [57], which might be provided by the Tensor Stokes (sub)problem. Moreover, the need of satisfying the additional LBB condition regarding a stable choice of the velocity-stress approximation – or velocity-Diffusion Tensor approximation in case of Eq. (2.57) – is potentially removed. Another significant effect of the Tensor Diffusion approach is established in terms of operator splitting techniques regarding differential as well as integral models, which are not applicable in case of the original non-solvent model formulation. But, such solution approaches are more or less unavoidable especially regarding stationary integral models, since a fully coupled treatment turns out to be not practical. When introducing the proposed Diffusion Tensor $\mathbf{M}$ into the corresponding system, a well-posed problem is obtained by the (discretised) Tensor

Stokes subproblem of the full differential or integral flow model. Hence, a velocity and pressure solution can be obtained by solving

$$\mathcal{T}\left(\boldsymbol{\mu}^{k}\right)\boldsymbol{u}^{k+1}+\mathcal{B}\boldsymbol{p}^{k+1}=\mathbf{r}_{u},\quad \mathcal{B}^{\top}\boldsymbol{u}^{k+1}=\mathbf{r}_{p}$$

for $\left(\boldsymbol{u}^{k+1},\boldsymbol{p}^{k+1}\right)$ in a first step of a prototypical operator splitting approach, where $\boldsymbol{\mu}^{k}$ is given. Next, the new velocity solution can then be used in the algebraic approach to calculate the updated extra-stress tensor $\boldsymbol{\sigma}^{k+1}$ from the discretised either differential or integral constitutive equation, where the latter is based on evolving the discrete Deformation Fields. In a third step, the Diffusion Tensor is updated by a discrete version of Eq. (2.51c) accomplishing a full loop of a useful decoupled solution technique, which can not be established in case of the original problems characterised by Eq. (2.45). In principle, the operator splitting technique may be applied regarding the PDE approach proposed in Eq. (2.57) as well. In doing so, the velocity field $\boldsymbol{u}^{k+1}$ would be directly used to compute the updated Diffusion Tensor $\boldsymbol{\mu}^{k+1}$ from the discrete version of Eq. (2.57b) avoiding the intermediate step of calculating $\boldsymbol{\sigma}^{k+1}$ compared to the algebraic approach. Thus, the novel Tensor Diffusion approach provides an at least conceptually improved scheme for solving the corresponding set of equations numerically in a decoupled manner, even in the non-solvent case.

In summary, a diffusive operator of "natural" type is introduced into non-solvent viscoelastic fluid flow models by means of the novel Tensor Diffusion approach proposed above. In doing so, the numerical treatment of corresponding flow problems is potentially improved in terms of three- or four-field formulations of the Tensor Stokes problem concerning both, monolithic as well as segragated solution techniques. Moreover, direct steady-state solutions involving differential or integral constitutive equations could be successfully computed simply based on a generalised Stokes-like problem. This might be achieved by explicitly expressing the complex rheology of the fluid by means of the Diffusion Tensor, which potentially avoids the need of considering constitutive equations at all. However, a detailed validation and evaluation of the Tensor Diffusion approach is presented in Chpt. 4 – also with respect to deriving analytic representations of the Diffusion Tensor, for which preliminary basic investigations are performed in Chpt. 3.

Chapter 3

Calculating fully developed viscoelastic flow profiles

So far, foundational and prototypical discretisation and solution techniques regarding viscoelastic differential and integral flow models have been discussed. But, no detailed remark is made concerning the applied boundary conditions supplementing the flow model for obtaining a well-posed mathematical problem. Following Ref. [20], the velocity field should be prescribed on the complete boundary of the computational domain for successfully performing numerical simulations in the context of differential constitutive equations, in case no such thing as symmetry boundary conditions needs to be taken into account. For example, a non-vanishing velocity profile is set on the in- and outflow egdes, that is on Γ_{in} and Γ_{out}, when simulating viscoelastic Poiseuille flow as depicted in Fig. 3.1, while $\mathbf{u} = (u, v)^\top \equiv 0$ is fixed on $\Gamma_{\text{no-slip}}$. In fact, this flow configuration can be regarded as a section of length $[0, L]$ of a channel of infinite length with height $[a, b]$, where the flow would not be affected by any in- or outflow boundary values. Consequently, such a flow state might be reproduced in the section of the channel by specifying the in- and outflow data as fully developed flow profiles according to the chosen material model. The corresponding velocity field $\mathbf{u}$ would consist of a non-vanishing y-dependent velocity u in x-direction, while $v \equiv 0$ holds regarding the y-velocity.

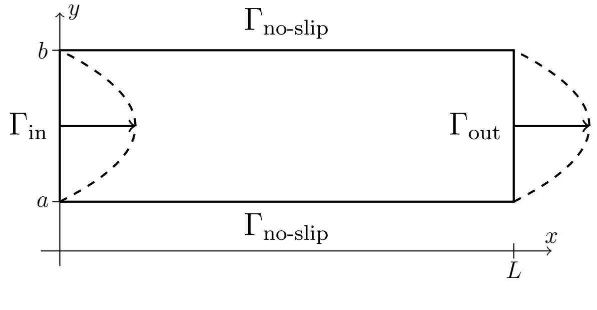
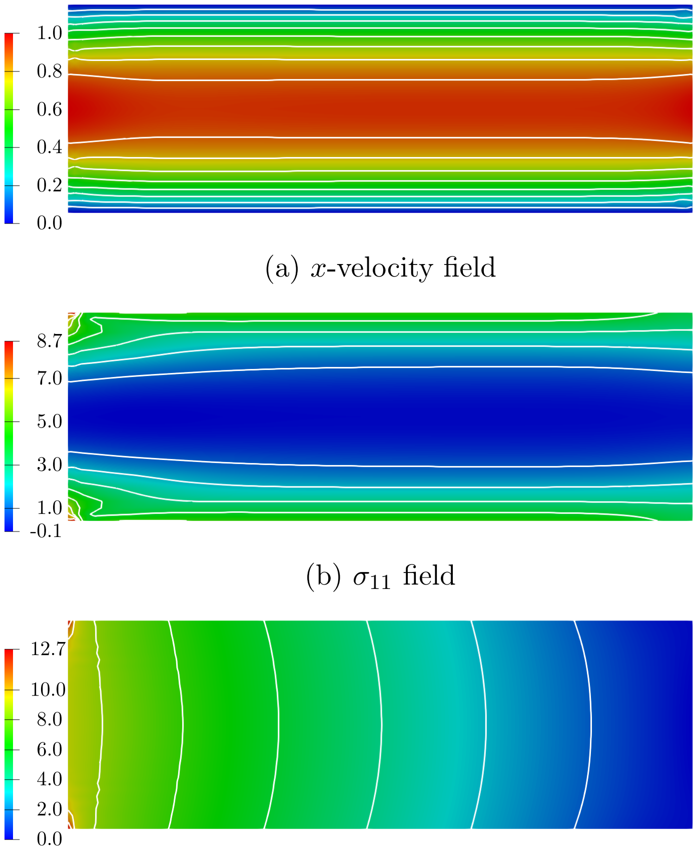


Figure 3.1: Poiseuille flow configuration

In case of one of the simplest models for describing viscoelastic material behaviour, that is the Oldroyd-B or UCM model, the resulting fully developed velocity profile is of parabolic shape (see Ref. [47] or Sec. 3.1.1). Hence, a parabolic velocity profile is a suitable choice regarding boundary values in corresponding viscoelastic fluid flow simulations, which is possibly combined with the according stress profiles applied at the inflow edge [20]. In contrast, considering parabolic in- and outflow profiles in case of simulating viscoelastic Poiseuille flow described by the Giesekus model leads to flow phenomena caused by the prescribed boundary data (see Fig. 3.2). It turns out, that the applied parabolic velocity profile is an inapropriate choice, since the velocity field takes a different shape away from the in- and outflow edges as realised from Fig. 3.2(a). This is contradictory to the numerical results expected for Poiseuille flow, which should show straight contour lines across the whole fluid domain. The effect of inappropriate boundary data becomes even more clear in case of considering the corresponding stress and pressure fields depicted in Figs. 3.2(b) and 3.2(c). In detail, significant oscillations or perturbations are visible particularly close to the inflow edge, which is equipped additionally with the stress profiles resulting from the parabolic velocity field. Consequently, it is highly recommended to apply suitable in- and outflow boundary data according to the considered viscoelastic model.



(a) x-velocity field

(b) σ_{11} field

(c) p field

Figure 3.2: Flow quantities resulting from the Giesekus model with parabolic in- and outflow profiles

Similarly, it makes sense to prescribe fully developed flow profiles as in- and outflow data in numerical simulations of complex flow configurations, since numerical artefacts arising from inappropriate boundary conditions are suppressed. At the same time, it needs to be ensured that the flow at the in- and outflow edges indeed is fully developed. Hence, it should not be affected by flow phenomena arising in the interior of the domain due to obstacles or overall geometrical changes, for example of the cross section. This can be done by sufficiently increasing the length of the up- and downstream channels of the computational domain to provide enough space for the flow to relax to the fully developed state. Thus, flow phenomena or oscillations as well as instabilities are damped or even removed, which would arise in case of a fully developed flow being enforced at the in- and outflow regardless of its actual – possibly not fully developed – shape. In addition, the flow can be considered indeed as fully developed when entering (and leaving) the actual computational domain. This corresponds to an important feature of viscoelastic fluids, where the current flow state is affected by deformations applied in the past. Thus, assuming a fully developed flow at the inlet of the computational domain – also with respect to the stress tensor – ensures an evolved deformed state of the fluid when entering the domain, where no recently applied deformations affect the fluid flow [11].

Unfortunetaly, fully developed flow profiles can not be derived straightforward in an analytical and closed form concerning various nonlinear viscoelastic models (see Refs. [9, 58, 59, 60]). Hence, a numerical technique for determining such viscoelastic flow profiles, which among other things can be applied as suitable in- and outflow data, is presented in the following. But in this work, the numerical framework for treating differential and integral viscoelastic flow models in terms of fully developed channel flows is mainly taken into account for another purpose. In detail, the initial validation of the novel Tensor Diffusion approach proposed in Sec. 2.4 is performed by means of such flow configurations, which furthermore offer the possibility to simplify both types of flow models and allow a deeper insight into the underlying structure and mathematical properties. In order to prepare the initial investigations on the Tensor Diffusion approach, the fully developed flow profiles resulting in case of the Giesekus model are validated and evaluated in Sec. 3.1 with respect to the included material and model parameters. In doing so, the effect of the parameters on the shape of the obtained velocity and stress fields is elaborated to provide the basis for analysing the form of the corresponding Diffusion Tensor in Sec. 4.1.1. Concerning integral constitutive equations, only a selection of the resulting flow profiles is given in Sec. 3.2 for a brief comparison with the Giesekus model. Besides, one of the essential potential benefits of the Tensor Diffusion approach, that is the simplification of (integral) viscoelastic flow models to a generalised Stokes-like problem, is illustrated in this context, as the Diffusion Tensor can be given (semi-)analytically in case of the PSM as well as Wagner model.

3.1 Differential constitutive equations

In this section, the numerical framework for calculating fully developed flow profiles according to viscoelastic fluids is presented and validated by means of differential constitutive equations. Therefore, the Giesekus model is considered in Sec. 3.1.1 in terms of the Poiseuille flow configuration, which results in a one-dimensional system of nonlinear ordinary differential equations (ODE) with respect to the unknown flow quantities. Corresponding numerical results, which are computed based on a Finite Difference discretisation, are discussed in Sec. 3.1.2 taking into account the effect of varying material and model parameters. Finally, the obtained fully developed flow profiles for the Giesekus model are validated in terms of two-dimensional Finite Element simulations, where the suitably interpolated profiles are prescribed as Dirichlet boundary data.

3.1.1 The underlying framework

Obviously, a two-dimensional stationary flow state is obtained by the flow configuration depicted in Fig. 3.1, in case suitable time-independent in- and outflow data is prescribed. Thus, the direct steady-state differential viscoelastic flow model from Eq. (2.16) is considered below in the context of fully developed channel flows. Thereby, the velocity field is assumed to be of the same shape at any cutline over the channel height, that is simply consisting of a non-vanishing y-dependent contribution in x-direction. Similarly, also the components of the extra-stress tensor vary only with respect to the channel height, which is why the flow quantities satisfy

$$\mathbf{u} = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u\,(y) \\ 0 \end{pmatrix}, \quad \mathbf{D}\,(\mathbf{u}) = \frac{1}{2}\begin{pmatrix} 0 & \frac{\partial u}{\partial y} \\ \frac{\partial u}{\partial y} & 0 \end{pmatrix} \quad \frac{\partial}{\partial x}\sigma_{ij} = 0, \; i, j \in \{1, 2\} \tag{3.1}$$

where σ_{ij} denote the components of the extra-stress tensor $\mathbf{\Sigma} \in \mathbb{R}^{2\times 2}$. Applying the properties (3.1) to the steady-state version of the Giesekus model, that is inserting Eq. (2.5) into Eq. (2.16), leads to a one-dimensional nonlinear ODE system reading

$$\begin{aligned}
-\eta_s \frac{\partial^2 u}{\partial y^2} - \frac{\partial \sigma_{12}}{\partial y} + \frac{\partial p}{\partial x} &= 0 && \text{(3.2a)}\\
-\frac{\partial \sigma_{22}}{\partial y} + \frac{\partial p}{\partial y} &= 0 && \text{(3.2b)}\\
-2\sigma_{12}\frac{\partial u}{\partial y} + \frac{1}{\Lambda}\left[\sigma_{11} + \alpha\frac{\Lambda}{\eta_p}\left(\sigma_{11}^2 + \sigma_{12}^2\right)\right] &= 0 && \text{(3.2c)}\\
-\sigma_{22}\frac{\partial u}{\partial y} + \frac{1}{\Lambda}\left[\sigma_{12} + \alpha\frac{\Lambda}{\eta_p}\sigma_{12}\left(\sigma_{11} + \sigma_{22}\right)\right] &= \frac{\eta_p}{\Lambda}\frac{\partial u}{\partial y} && \text{(3.2d)}\\
\frac{1}{\Lambda}\left[\sigma_{22} + \alpha\frac{\Lambda}{\eta_p}\left(\sigma_{12}^2 + \sigma_{22}^2\right)\right] &= 0 && \text{(3.2e)}
\end{aligned}$$

Here, the polymeric viscosity η_p is linked to the solvent viscosity η_s by means of the total viscosity $\eta_0 = \eta_s + \eta_p$, where the amount of solvent contribution is specified via $\beta = \frac{\eta_s}{\eta_0}$. In the following, the unknown quantities $\left(\frac{\partial p}{\partial x}, u, \sigma_{11}, \sigma_{12}, \sigma_{22}\right)$ are determined, such that the resulting flow profiles correspond to a certain viscoelastic flow configuration typically defined by the so-called Weissenberg Number $\mathrm{We} = \Lambda\frac{u_c}{l_c}$. Hence, the flow is specified by means of the characteristic velocity u_c as well as length l_c and the relaxation time Λ. Regarding the flow configuration depicted in Fig. 3.1, l_c is determined by the channel height $b - a$, while u_c is given in terms of the mean value U_{mean} of the velocity. Thus, it makes sense to generate fully developed flow profiles resulting in a prescribed flow rate of $\dot{U} = U_{\text{mean}}\,(b - a)$, which is why the above system of equations is supplemented by the condition

$$\int_a^b u\,(y)\;\mathrm{d}y = \dot{U} \tag{3.2f}$$

In fact, the set of equations (3.2) can be solved analytically in case a Newtonian fluid, that is $\Lambda = 0$, or the Oldroyd-B ($\alpha = 0$, $\beta \in \,]0, 1[$) as well as the UCM model ($\alpha = 0$, $\beta = 0$) for $\Lambda > 0$, resulting in a parabolic velocity profile of the form

$$u\,(y) = \frac{1}{2\eta_0}\frac{\partial p}{\partial x}\left(y^2 - (b + a)\,y + ba\right) \tag{3.3}$$

Thereby, $u\left(a\right) = u\left(b\right) = 0$ holds due to $\mathbf{u} \equiv 0$ on $\Gamma_{\text{no-slip}}$, while the maximum velocity U is obtained on the centre line of the channel. Moreover, the pressure drop in the parabolic velocity profile can be determined easily as $\frac{\partial p}{\partial x} = -2\eta_0 U \left(\frac{b-a}{2}\right)^2$, where Eq. (3.2f) is satisfied due to $\dot{U} = \frac{2}{3}U\left(b-a\right)$. In addition,

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{pmatrix} = \begin{pmatrix} 2\eta_p \Lambda \left(\frac{\partial u}{\partial y}\right)^2 & \eta_p \frac{\partial u}{\partial y} \\ \eta_p \frac{\partial u}{\partial y} & 0 \end{pmatrix} \tag{3.4}$$

holds regarding the components of the extra-stress tensor. Thus, applying a parabolic velocity profile as fully developed in- and outflow profile in numerical simulations regarding the Oldroyd-B or UCM model is justified. However, based on the momentum equations (3.2a) and (3.2b) the pressure field may be written as

$$p\left(x, y\right) = c_1 + c_2 x + f_p\left(y\right) \tag{3.5}$$

where $c_1, c_2 \in \mathbb{R}$ are constants. Furthermore, for a velocity field pointing in positive x-direction the pressure decreases linearly in x, that is $\frac{\partial p}{\partial x} = c_2 \equiv \text{const} < 0$, which is already indicated by means of the Oldroyd-B or UCM model. In addition, the pressure might show a y-dependent behaviour described by the contribution $f_p : \mathbb{R} \to \mathbb{R}$, where $f_p = \sigma_{22}$ is directly implied from Eq. (3.2b). Note, that this equation does not need to be taken into account in the solution procedure described below, as all degrees of freedom are already covered by the remaining equations. Instead, the momentum equation (3.2a) acts as a key component, where the pressure drop is explicitly linked to the velocity field and hence basically to the desired flow rate – at least for $\eta_s > 0$. Moreover, the constitutive equations can be regarded as an auxiliary problem to establish a nonlinear coupling of the pressure drop and the velocity field, even in the non-solvent case.

For treating the set of nonlinear equations presented in Eq. (3.2) numerically, discrete grid points $y_k = a + k\frac{b-a}{n}$, $k = 0, \ldots, n$, are spreaded over the channel height $y \in [a, b]$. The resulting discrete nonlinear system for calculating the fully developed flow profiles is obtained from Eq. (3.2) by applying a Finite Difference approximation of second order regarding the derivatives with respect to y. At the same time, no-slip boundary conditions regarding $\mathbf{u}$ are prescribed, that is $u\left(a = y_0\right) = u\left(b = y_n\right) = 0$, while do-nothing boundary conditions with respect to the components of the extra-stress tensor are chosen. The condition (3.2f) specifying the flow rate is approximated by a suitable quadrature rule, for example the Composite-Trapezoidal Rule. The resulting nonlinear system for the discrete values of $\left(u, \sigma_{11}, \sigma_{12}, \sigma_{22}\right)$ as well as the pressure drop $\frac{\partial p}{\partial x}$ is solved by means of Newton's method, where the discrete Jacobian matrix is calculated analytically. Note, that the number of discrete grid points $n \in \mathbb{N}$ is chosen, such that the profiles show a mesh-converged behaviour in the picture norm, whereas a detailed convergence study may be performed for example based on the convergence of the pressure drop $\frac{\partial p}{\partial x}$ with respect to n. But this is not done here, since the resulting flow profiles are intended to be used for example as boundary data in two-dimensional Finite Element simulations, where the discrete data points are interpolated anyways to generate suitable boundary values according to a probably coarser spatial resolution. Furthermore, the scope of the following analysis is to evaluate the shape of the flow profiles with respect to varying material and model parameters, which is why it is sufficient to consider numerical results of the above framework converged up to a visual "tolerance".

In principle, the problem formulation introduced above by means of the single-mode Giesekus model can be extended to the multi-mode case straightforward. For this purpose, separate constitutive equations (3.2c), (3.2d) and (3.2e) need to be considered for each stress tensor $\boldsymbol{\Sigma}_k$ including separate parameters Λ_k, $\eta_{p,k}$ and α_k, while the extra-stress component in the momentum equation (3.2a) needs to be written as $\sigma_{12} = \sum_{k=1}^{K} \sigma_{12,k}$. Naturally, the numerical effort will increase, but the overall framework can still be applied.

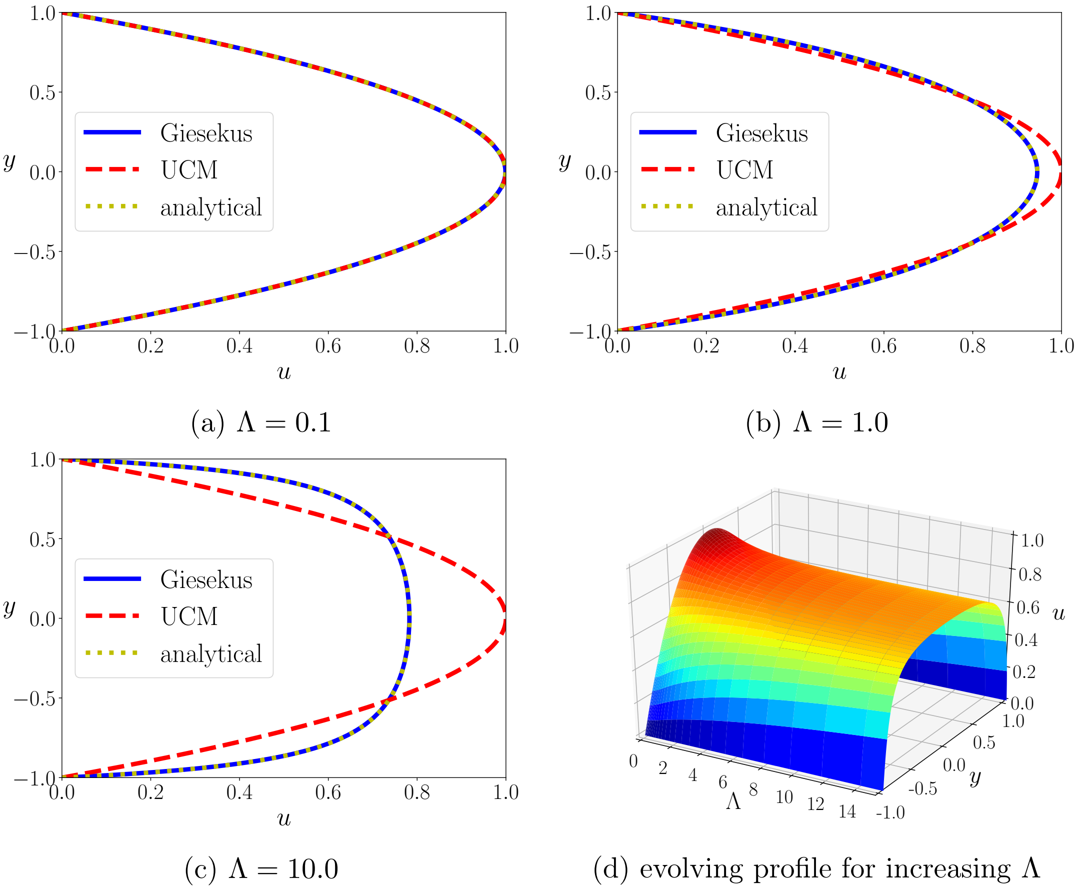
Furthermore, the numerical method is applicable conceptually to any differential viscoelastic model by adapting the constitutive equations (3.2c), (3.2d) and (3.2e). In case of the PTT model presented in Eq. (2.6), the complexity of the numerical problem can even be reduced significantly. Following Ref. [9], the velocity profile in the non-solvent case $\eta_s = 0$ is given analytically depending on the pressure drop as well as the chosen model and material parameters, in detail

$$\begin{aligned} u_{\text{lin}}(y) &= \frac{1}{2\eta_p}\frac{\partial p}{\partial x}\left[\left(1+\kappa\left(\frac{\Lambda}{\eta_p}\frac{\partial p}{\partial x}\left(y-\frac{b+a}{2}\right)\right)^2\right)\left(y-\frac{b+a}{2}\right)^2 \dots\right. \\ &\quad \left.\dots - \left(1+\kappa\left(\frac{\Lambda}{\eta_p}\frac{\partial p}{\partial x}\left(\frac{b-a}{2}\right)\right)^2\right)\left(\frac{b-a}{2}\right)^2\right] \qquad (3.6a) \\ u_{\text{exp}}(y) &= \frac{\eta_p}{4\kappa\Lambda^2\frac{\partial p}{\partial x}}\left[\exp\left(2\kappa\left(\frac{\Lambda}{\eta_p}\frac{\partial p}{\partial x}\left(y-\frac{b+a}{2}\right)\right)^2\right)\dots\right. \\ &\quad \left.\dots - \exp\left(2\kappa\left(\frac{\Lambda}{\eta_p}\frac{\partial p}{\partial x}\left(\frac{b-a}{2}\right)\right)^2\right)\right] \qquad (3.6b) \end{aligned}$$

Thus, the one-dimensional nonlinear set of equations from Eq. (3.2) may be transformed to the problem of determining the pressure drop, such that the velocity profile results in the desired flow rate. But, this problem can not be solved as straightforward as in case of the Oldroyd-B or UCM model including a parabolic velocity profile or a linear dependency on $\frac{\partial p}{\partial x}$. Instead, a corresponding solution might be computed via nonlinear optimisation, where the objective function is defined based on the integral expression from Eq. (3.2f) including the analytical profiles from Eq. (3.6). In fact, the same can be done for the Giesekus model as well, where a corresponding semi-analytical expression of the fully developed velocity profile in the non-solvent case is given in Ref. [9]. But in this work, the semi-analytical solution is used only for validating the one-dimensional numerical framework introduced above, which provides a kind of "black box" tool for calculating fully developed flow profiles regarding in principle any (differential) constitutive law or specific rheology. Furthermore, a numerical solution technique needs to be applied anyways in case of a non-vanishing solvent viscosity, also regarding the PTT model. However, the PTT model will not be considered here in detail, since the corresponding numerical solutions show a behaviour similar to the Giesekus model, which is discussed in the following section. But, the flow profiles according to the PTT model will be applied in Sec. 4.2 in terms of boundary conditions for evaluating the Tensor Diffusion approach in two-dimensional Finite Element simulations.

3.1.2 Numerical results and validation

As mentioned before, no realistic or meaningful viscoelastic material behaviour can be predicted by means of the Oldroyd-B or UCM model. For example, this can be realised based on the corresponding fully developed parabolic velocity profile from Eq. (3.3), which is independent of the relaxation time Λ. Consequently, no modelling of any nonlinear effects such as shear thinning, which occurs for increasing relaxation times, is possible, although this represents a typical viscoelastic phenomenon in terms of channel flow configurations. Thus, results for the Giesekus model computed by means of the numerical framework presented above are discussed to illustrate, that indeed flow profiles deviating from the parabolic shape are obtained. In addition, the flow profiles are evaluated with respect to the effect of the included material and model parameters on the nonlinearity of the solution. Therefore, the calculated velocity profiles are compared against a parabolic profile with a maximum velocity of $U = 1$ over a channel height of $y \in [-1, 1]$, which is why the numerically determined velocity profiles need to result in a flow rate of $\dot{U} = \frac{2}{3} U \left(b - a\right) = \frac{4}{3}$. Furthermore, the corresponding stress profiles are taken into account as well.



(a) $\Lambda = 0.1$

(b) $\Lambda = 1.0$

(c) $\Lambda = 10.0$

(d) evolving profile for increasing Λ

Figure 3.3: x-velocity profiles for the Giesekus model at $\alpha = 0.1$, $\eta_0 = 1.0$, $\beta = 0.0$, $U = 1.0$ for several relaxation times Λ

In a first step, the flow profiles resulting from the non-solvent case, that is $\eta_s = 0$ in Eq. (3.2a), and fixed $\alpha = 0.1$ are considered for various values of Λ. For increasing relaxation times, a typical shear thinning behaviour can be observed regarding the velocity profiles depicted in Fig. 3.3, which is one of the main material properties predicted by the Giesekus model. In case of shear thinning, the velocity profile deviates from the parabolic profile, such that a large velocity gradient arises close to the walls, while a plateau-like behaviour is observed in the middle of the channel. Furthermore, a very good agreement to the semi-analytical reference solution given in Ref. [9] is observed on a visual basis, where the numerical results are obtained for a spatial discretisation consisting of $n = 513$ one-dimensional grid points. In particular, the large velocity gradients close to the channel wall, which are caused by the pronounced shear thinning effect, are captured precisely. In Fig. 3.4, the error between the computational result and the reference solution regarding the velocity profile is plotted over n, where the latter is evaluated for the numerically determined pressure drop $\frac{\partial p}{\partial x}$. It turns out, that a resolution of $n = 513$ already provides a quite accurate approximation of the semi-analytical expression of the velocity field even for the considered – quite extreme – parameter configuration. Moreover, the approximation quality is sufficiently high regarding the application of these (suitably interpolated) fully developed velocity profiles as boundary data in Finite Element simulations. Probably, corresponding two-dimensional configurations consist of a much coarser spatial resolution of the corresponding in- and outflow egdes, which is why the approximation error in the one-dimensional setting does not need to be decreased too much. Consequently, a spatial resolution of $n = 513$ is applied in the one-dimensional framework to also generate the numerical results presented in the following.

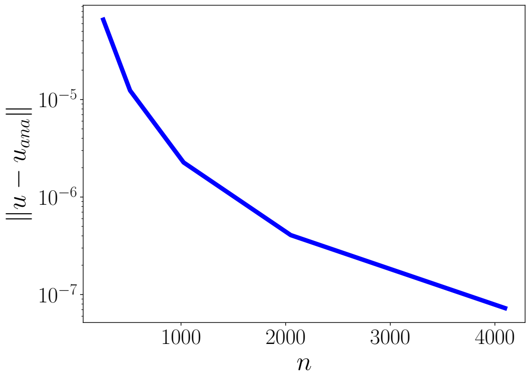
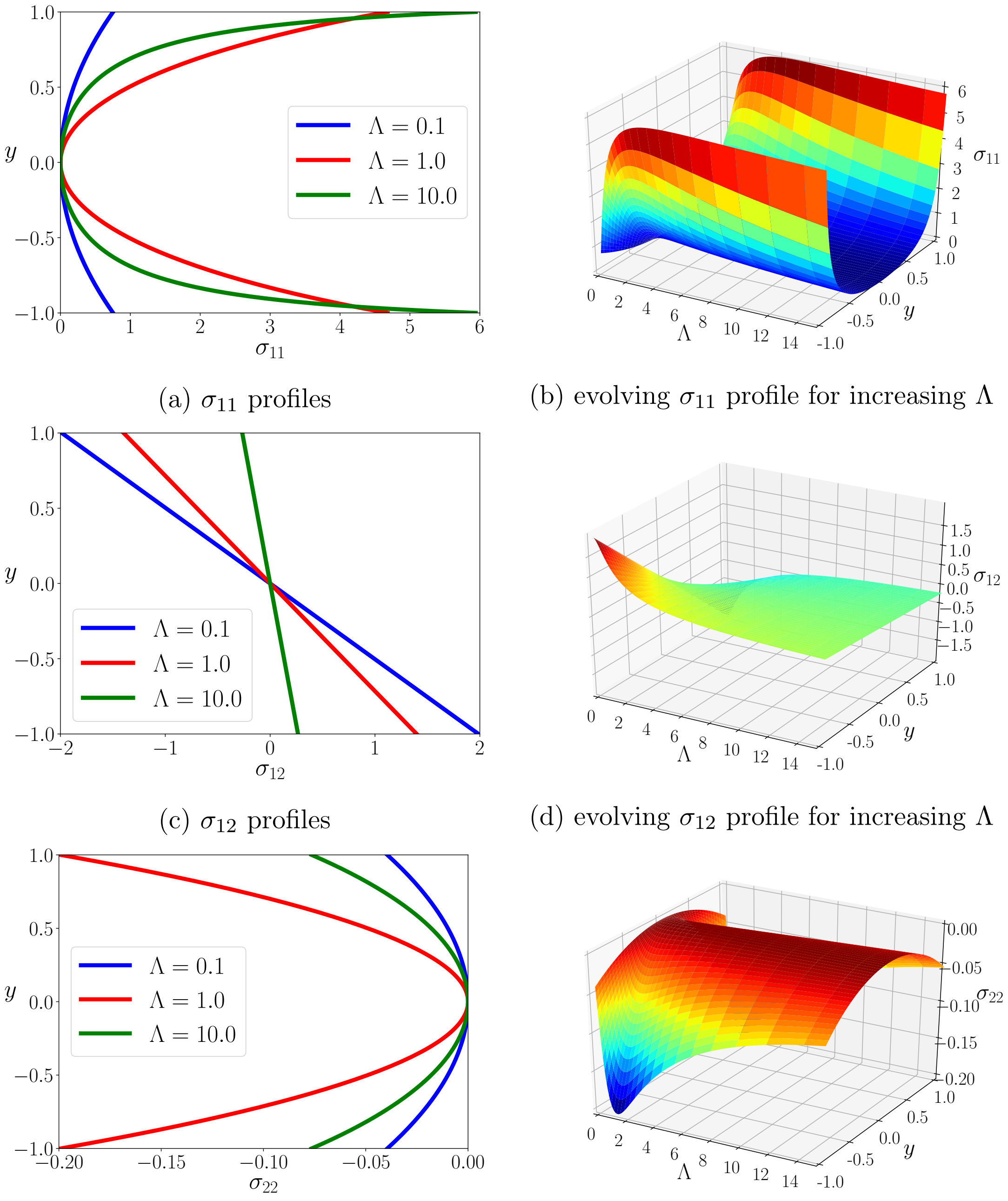


Figure 3.4: Error of calculated x-velocity profile to the analytical solution for the Giesekus model at $\Lambda = 10.0, \alpha = 0.1, \eta_0 = 1.0, \beta = 0.0$, $U = 1.0$ for increasing spatial resolution

To further outline the difference in the solutions of the nonlinear Giesekus model compared to the linear UCM model, the stress profiles are presented in Fig. 3.5 in addition to the velocity profiles discussed above. For $\eta_s = 0$ and $\alpha = 0.1$, a rapid increase of σ_{11} is observed for smaller relaxation times, which is kind of consistent to the UCM model, where the corresponding $\mathbf{\Sigma}$-component depends linearly on Λ. Thereby, the Giesekus model for small Λ indeed represents a "small perturbation" of the UCM model, which can be realised by means of Eqs. (3.2c), (3.2d) and (3.2e). However, a maximum value of σ_{11} at the channel walls is obtained at $\Lambda \approx 3$ followed by a slow descent, which illustrates the stabilizing character of the additional quadratic stress contribution in the constitutive equation compared to the UCM model [4, 6]. Recall, that σ_{11} grows unboundedly with repect to Λ in case of the UCM model, but stays parabolic with respect to y. Instead, the slope of σ_{11} obtained from the Giesekus model increases monotonously close to the channel walls according to the velocity field or the amount of shear thinning.



(a) σ_{11} profiles

(b) evolving σ_{11} profile for increasing Λ

(c) σ_{12} profiles

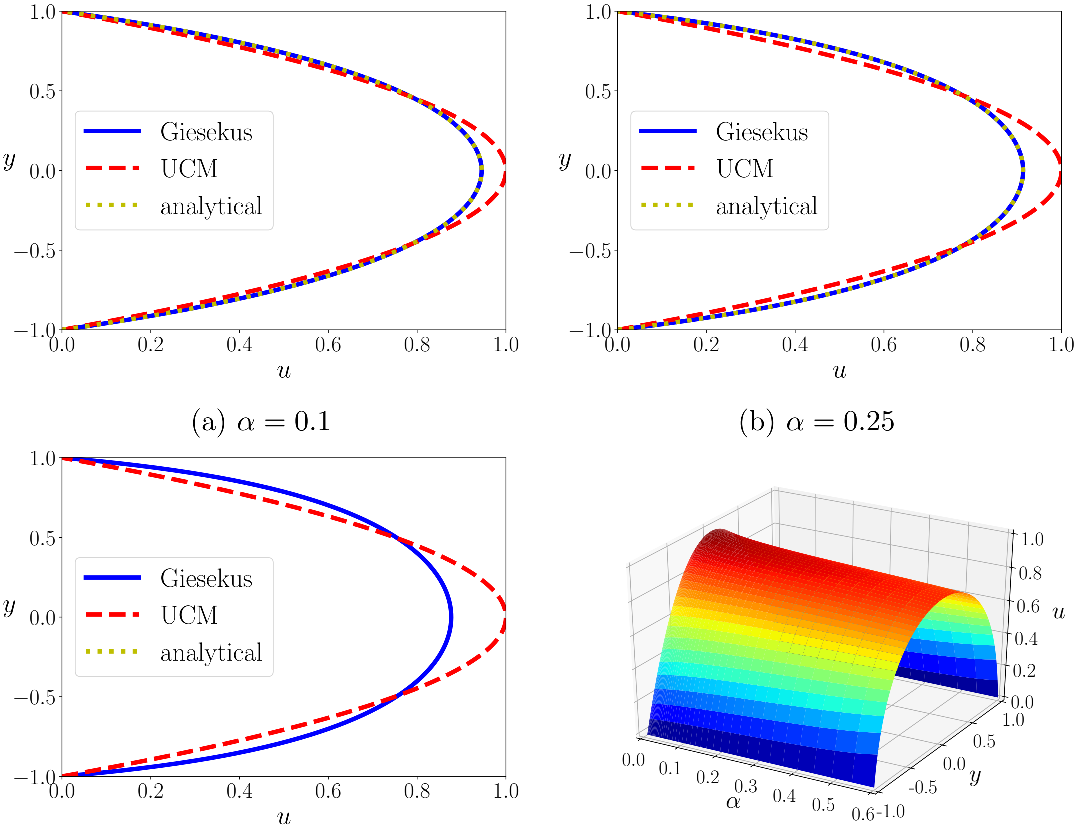
(d) evolving σ_{12} profile for increasing Λ

(e) σ_{22} profiles

(f) evolving σ_{22} profile for increasing Λ

Figure 3.5: Stress profiles for the Giesekus model at $\alpha = 0.1$, $\eta_0 = 1.0$, $\beta = 0.0$, $U = 1.0$ for several relaxation times Λ

Concerning the σ_{22}-profile, a maximum value with respect to magnitude is reached at $y = \pm 1$ as well, in detail at $\Lambda \approx 2$, while the slope as well as the magnitude decrease for larger relaxation times. In contrast, $\sigma_{22} = 0$ holds in case of the UCM model. A monotonous behaviour with respect to the relaxation time Λ is observed only for σ_{12}, where the magnitude as well as the slope of the corresponding profiles decrease for increasing Λ. In case of the UCM model, σ_{12} does not depend on the relaxation time at all. Overall, the stabilising character of the additional quadratic contribution concerning the extra-stress tensor $\mathbf{\Sigma}$ is observed. In particular, the component σ_{11} seems to approach a finite limit value for increasing relaxation times or Weissenberg numbers after a significant build-up of stress for lower or moderate Λ. But, also the remaining flow quantities resulting from the Giesekus model clearly deviate in a nonlinear manner from the (linear) UCM-solutions.



(a) $\alpha = 0.1$

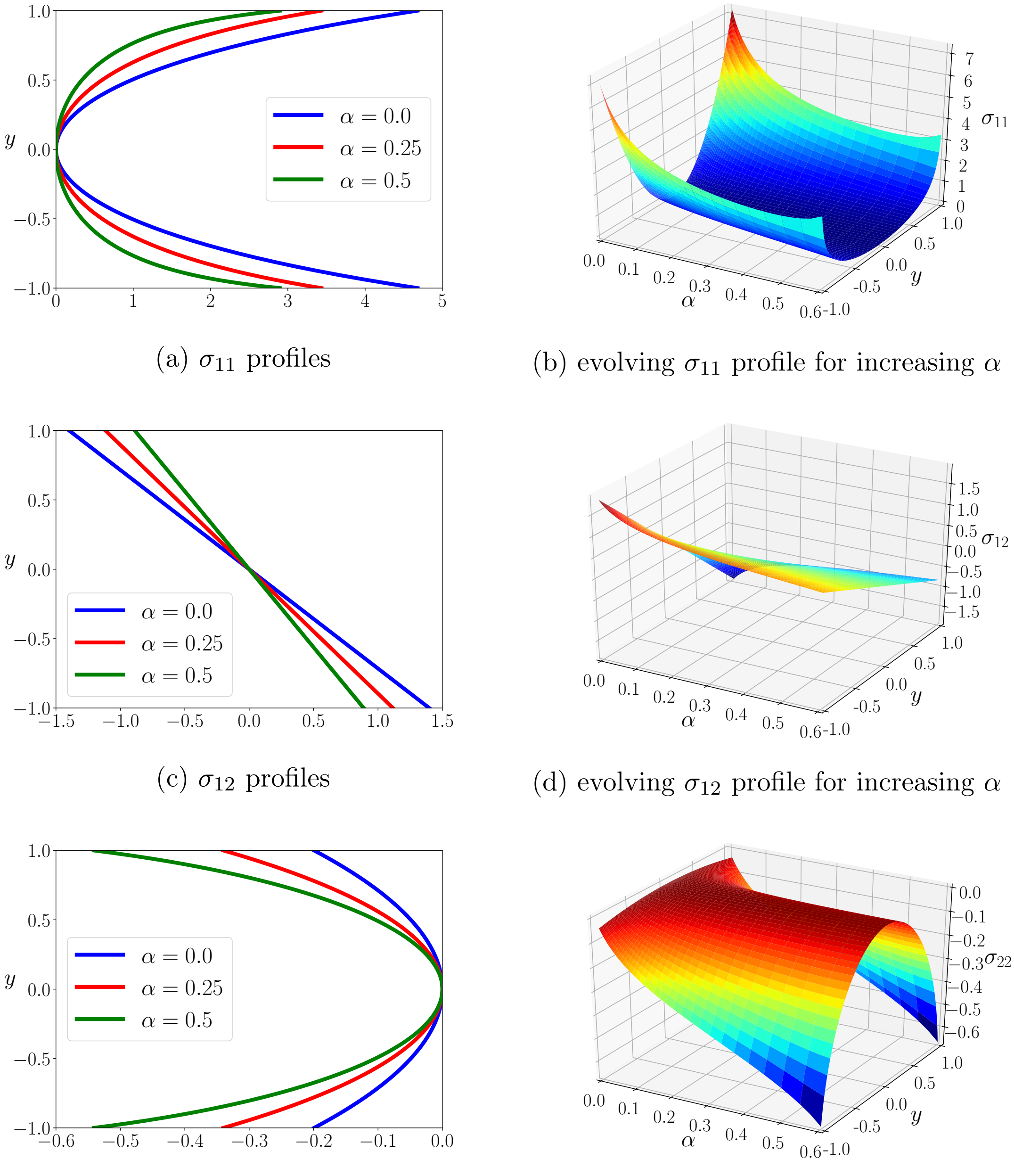
(b) $\alpha = 0.25$

(c) $\alpha = 0.5$

(d) evolving profile for increasing α

Figure 3.6: x-velocity profiles for the Giesekus model at $\Lambda = 1.0$, $\eta_0 = 1.0$, $\beta = 0.0$, $U = 1.0$ for several mobility factors α

After having analysed the behaviour of the flow profiles regarding the relaxation time Λ, as a next step the effect of a varying Giesekus parameter or mobility factor α is investigated. Therefore, again a vanishing solvent viscosity is considered and additionally the relaxation time is set to $\Lambda = 1.0$, which results in the flow profiles depicted in Figs. 3.6 and 3.7 for several values of α. It turns out, that increasing the mobility factor has in principle a similar effect on the velocity field as considering higher relaxation times, since an increasing shear thinning effect is observed. Similarly, also the stress component σ_{11} clearly deviates from the parabolic shape with respect to the channel height, which is obtained in case of the UCM model, that is $\alpha = 0$. In detail, its magnitude decreases, but the slope close to the channel wall increases according to shear thinning. Also regarding the σ_{22}-component, the deviations to the UCM-solution $\sigma_{22} = 0$ are intensified, since its magnitude grows for increasing α. Similar to larger Λ, σ_{12} preserves a linear form for higher α, which is also provided by the UCM model, although its magnitude decreases. In fact, this behaviour can be realised from Eq. (3.2a) giving $\sigma_{12} = \frac{\partial p}{\partial x} y$, since the pressure drop $\frac{\partial p}{\partial x} < 0$ increases according to α as well as Λ or the amount of shear thinning.



(a) σ_{11} profiles

(b) evolving σ_{11} profile for increasing α

(c) σ_{12} profiles

(d) evolving σ_{12} profile for increasing α

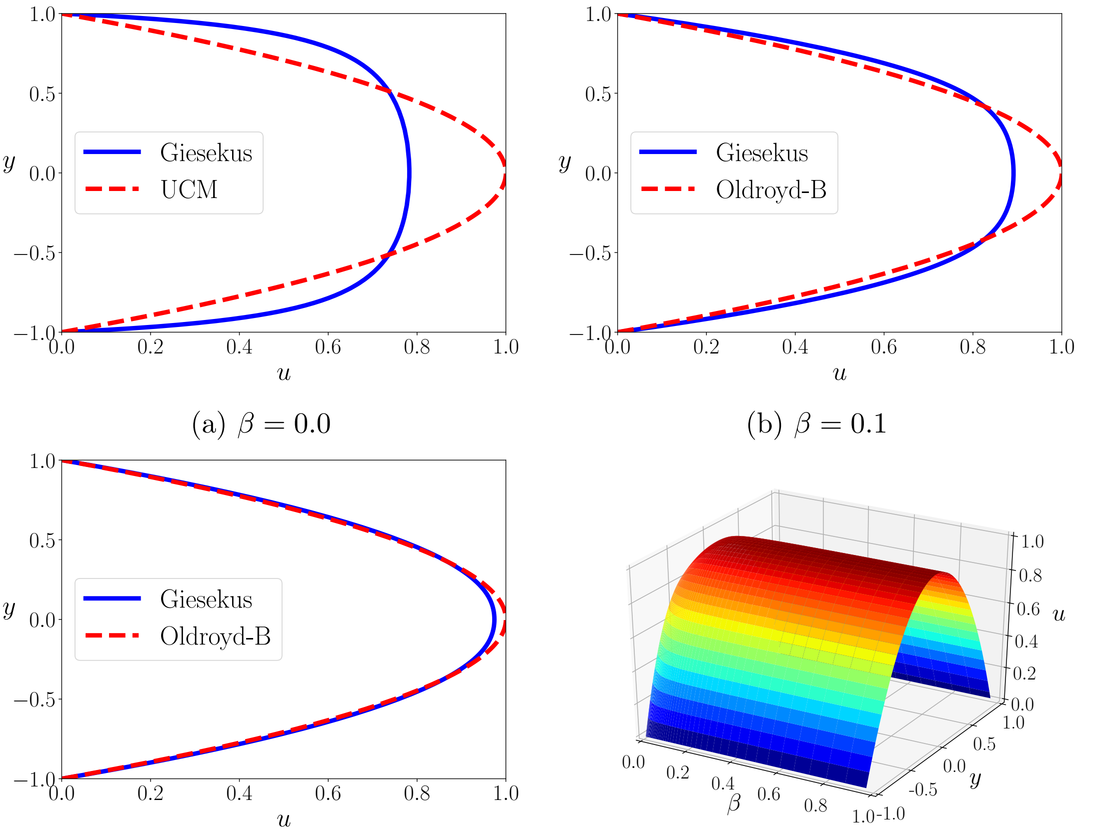
(e) σ_{22} profiles

(f) evolving σ_{22} profile for increasing α

Figure 3.7: Stress profiles for the Giesekus model at $\Lambda = 1.0$, $\eta_0 = 1.0$, $\beta = 0.0$, $U = 1.0$ for several mobility factors α

However, σ_{11} seems to grow again close to the channel walls for $\alpha > 0.5$, but no solutions corresponding to significantly larger α are computable due to the structural limits of the Giesekus model outlined at the end of Sec. 3.2.2. Thus, the change in the behaviour of σ_{11} can not be pursued. However, in contrast to increasing the relaxation time, no pronounced build-up of the stress is observed for small values of α, as a more or less monotonous behaviour of $\mathbf{\Sigma}$ with respect to the mobility factor is observed.

Up to now, the non-solvent case is considered only, that is $\eta_s = 0$ is set in the momentum equation (3.2a). Hence, the effect of a present viscosity on the fully developed flow profiles is investigated in the following, where corresponding results are depicted in Figs. 3.8 and 3.9.



(a) $\beta = 0.0$

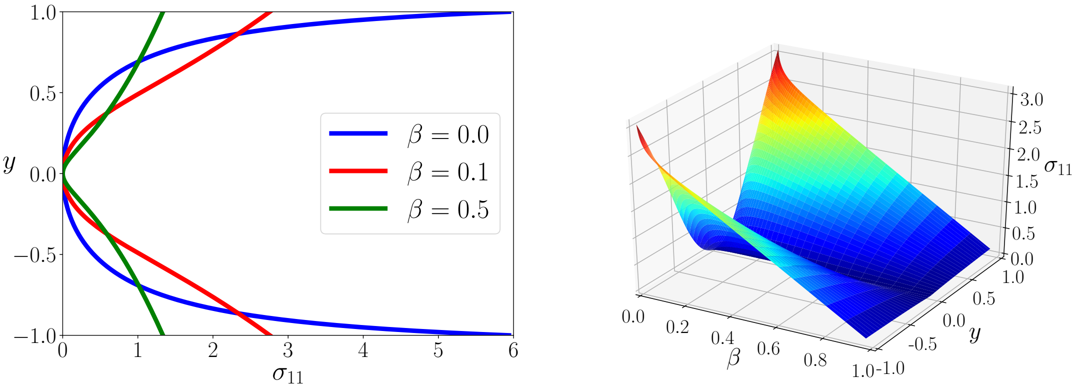
(b) $\beta = 0.1$

(c) $\beta = 0.5$

(d) evolving profile for increasing β

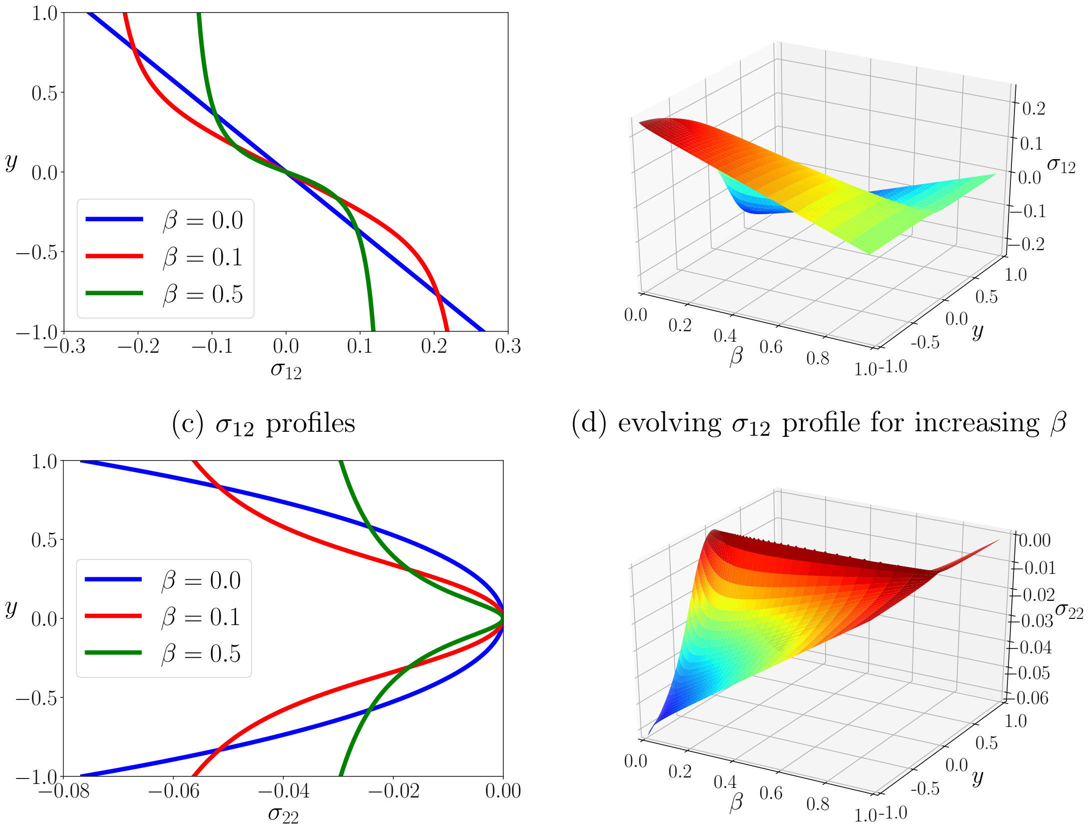
Figure 3.8: x-velocity profiles for the Giesekus model at $\Lambda = 10.0$, $\eta_0 = 1.0$, $\alpha = 0.1$, $U = 1.0$ for several amounts β of solvent viscosity

As expected, the velocity profiles for $\beta = \frac{\eta_s}{\eta_0} \to 1$ reform incrementally to a parabolic shape, which correlates to the velocity profile of a Newtonian fluid. Thereby, a pronounced change of the solutions is observed for small β, while a nearly parabolic profile is already recovered at $\beta \approx 0.5$.



(a) σ_{11} profiles

(b) evolving σ_{11} profile for increasing β



(c) σ_{12} profiles

(d) evolving σ_{12} profile for increasing β

(e) σ_{22} profiles

(f) evolving σ_{22} profile for increasing β

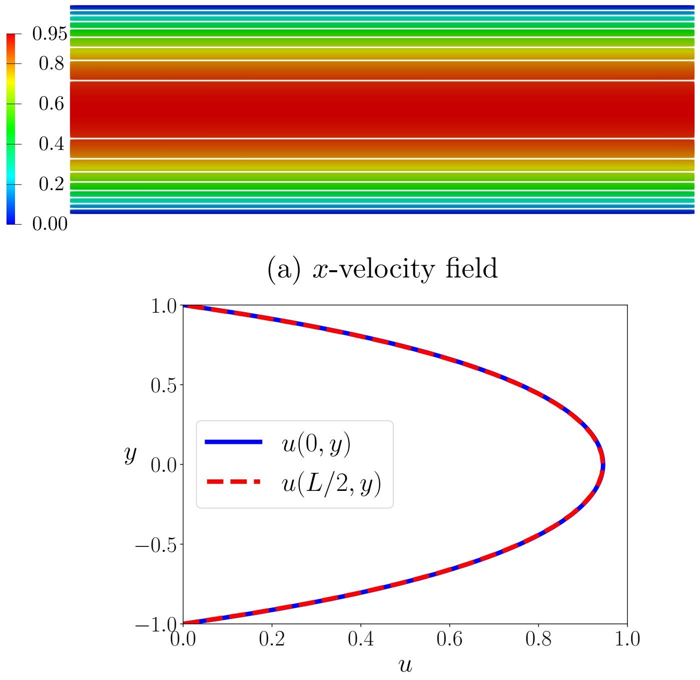
Figure 3.9: Stress profiles for the Giesekus model at $\Lambda = 10.0$, $\eta_0 = 1.0$, $\alpha = 0.1$, $U = 1.0$ for several amounts β of solvent viscosity

Naturally, also the stress profiles approach the Newtonian limit $\mathbf{\Sigma} = 0$ for $\beta \to 1$, which can be realised by means of the nonlinear set of equations from Eq. (3.2) as well. In addition, the profiles resulting in case of $\beta > 0$ deviate significantly from the solutions obtained for pure melts, since a large stress gradient is observed close to the centre line of the channel at $y = 0$ (see Fig. 3.9). But at the same time, the overall magnitude is decreased and a plateau-like behaviour is formed towards the channel walls. Obviously, the underlying shear thinning effect is damped by introducing a solvent contribution to the viscosity, which causes the varying shape of the flow profiles due to the fluid becoming more Newtonian. Consequently, the velocity profile recovers the "linear" solution, but the stress profiles for $\beta \to 1$ still show nonlinear effects close to the centre line of the channel.

Overall, the flow profiles calculated for the Giesekus model in terms of fully developed channel flow configurations show a reasonable behaviour under variation of the present parameters, that is the relaxation time Λ, the mobility factor α and the amount of solvent contribution β. It turns out, that the velocity profiles consist of a distinct shear thinning effect for increasing Λ as well as α, which typically is predicted by means of the Giesekus model [4, 6]. In addition, introducing a present solvent contribution pushes the flow profiles towards the Newtonian case.

In the following, the validity of the fully developed flow profiles determined above is illustrated by means of Finite Element simulations of Poiseuille flow of a viscoelastic fluid described by the Giesekus model. In doing so, Dirichlet boundary conditions with respect to the velocity are prescribed at every boundary segment of the computational domain, including fully developed velocity profiles applied as in- and outflow data at Γ_{in} and Γ_{out}. These profiles are determined by means of Finite Element interpolants of the discrete values of the Finite Difference discretisation proposed in Sec. 3.1.1. Thereby, the required spatial resolution for determining the fully developed flow profiles in the one-dimensional framework needs to be sufficiently fine, for example to resolve large gradients of the velocity profile in case of a pronounced shear thinning effect. In contrast, the resolution of the in- and outflow edges in two-dimensional Finite Element simulations typically will be much coarser. Hence, the data points used for interpolation are chosen from the Finite Difference approximation according to the degrees of freedom of the Finite Element discretisation to establish accurate boundary data. However, the upper and lower walls of the channel are set to be no-slip boundaries, that means $\mathbf{u} \equiv 0$ is prescribed on $\Gamma_{\text{no-slip}}$. At the inlet Γ_{in}, the fully developed stress profiles are prescribed as well, but a do-nothing boundary condition with respect to $\mathbf{\Sigma}$ is applied on $\Gamma_{\text{no-slip}}$ as well as Γ_{out}.

In this context, the prescribed boundary values can be considered as the actual fully developed flow profiles according to the Giesekus model, if the contour lines of the x-component u of the velocity field $\mathbf{u}$ as well as the contour lines of the components of the extra-stress tensor $\mathbf{\Sigma}$ obey a purely straight behaviour in the complete channel. This straight shape implies, that the same profile of the associated quantity is obtained on every cutline over the channel height – including the prescribed in- and outflow edge. Thus, there is no change in the flow with respect to x and the profiles arising from the two-dimensional Finite Element simulation coincide with the profiles applied at the boundary. In this case, the results of the fully developed channel flow configuration can be regarded as validated.



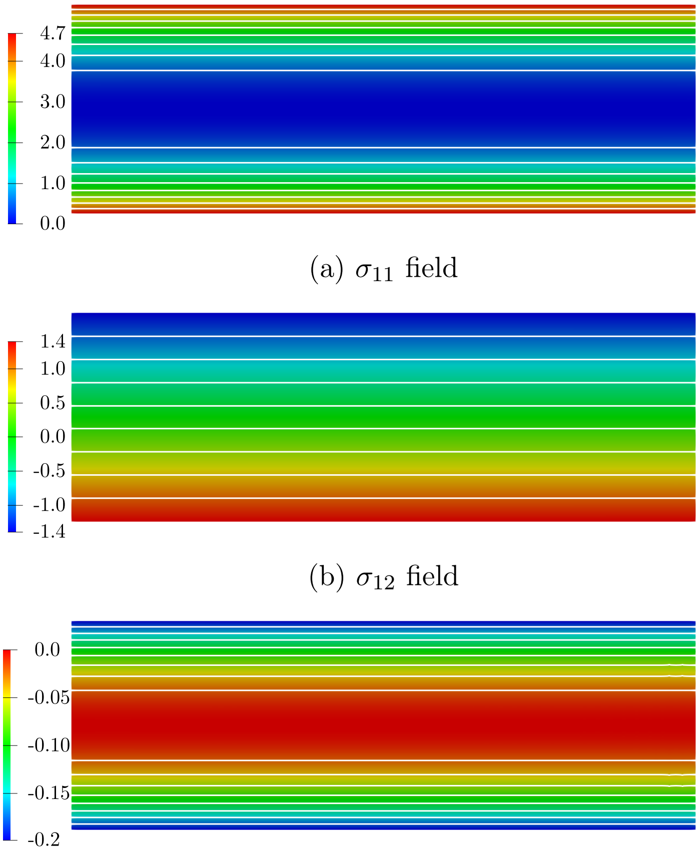
(a) x-velocity field

(b) x-velocity profile

Figure 3.10: Velocity solution for the Giesekus model at $\Lambda = 1.0$, $\alpha = 0.1$, $\eta_0 = 1.0$, $\beta = 0.0$, $U = 1.0$

As representative study, non-solvent viscoelastic fluids described by the

Giesekus model at fixed $\Lambda = 1.0$ and $\alpha = 0.1$ are simulated. The computational domain is defined based on Fig. 3.1 as a rectangular channel of height $y \in [-1, 1]$ and length $x \in [0, 6]$, where the actual computational mesh is build by a successive refinement of a coarse mesh consisting of three squares. Moreover, the full set of variables $(\mathbf{u}, \mathbf{\Sigma}, p)$ is discretised according to the techniques described in Sec. 2.2.3, where suitable fully developed flow profiles presented in Figs. 3.3 and 3.5 are applied as in- and outflow profiles. Corresponding simulation results regarding the velocity field can be found in Fig. 3.10, where the contour lines indeed show a perfectly straight behaviour and the profile in the middle of the channel matches the one-dimensional result prescribed at the in- and outflow edges. The same can be observed regarding the numerical solution of the stress variables depicted in Fig. 3.11, where the profiles set as Dirichlet data at Γ_{in} represent the fully developed stress profiles. Hence, the flow profiles determined in the one-dimensional framework proposed above coincide with the actual fully developed profiles arising in two-dimensional simulations.



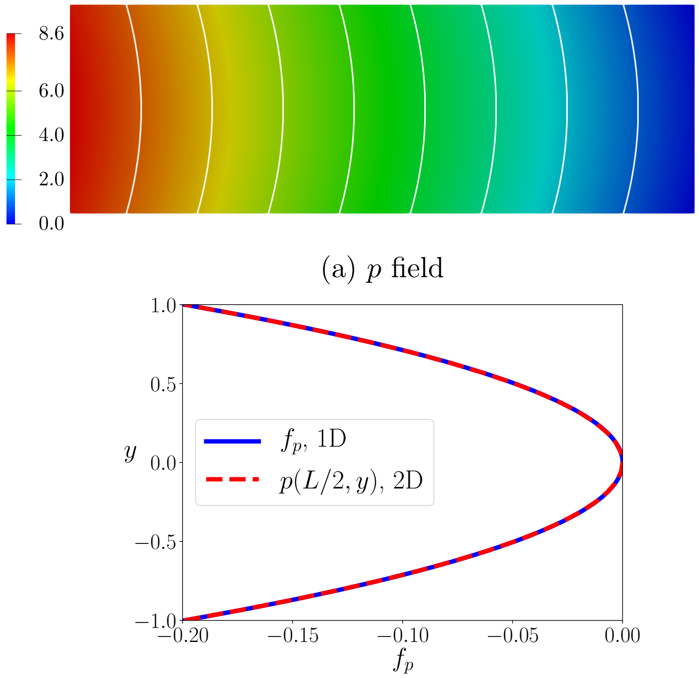
(a) σ_{11} field

(b) σ_{12} field

(c) σ_{22} field

Figure 3.11: Stress solution for the Giesekus model at $\Lambda = 1.0$, $\alpha = 0.1$, $\eta_0 = 1.0$, $\beta = 0.0$, $U = 1.0$

Besides the shear thinning effect, a typical observation in terms of the Giesekus model is the fact, that the pressure shows a y-dependent behaviour. Actually, the corresponding contour lines over the channel height take a curved shape, in contrast to the Oldroyd-B or UCM model, where the pressure consists of straight contour lines with respect to y. This issue is already addressed by the pressure function presented in Eq. (3.5), where the y-dependent part is characterised by $f_p(y) = \sigma_{22}(y)$. It turns out, that this pressure characteristic actually is recovered in the two-dimensional simulations of the Poiseuille flow (see Fig. 3.12(a)). In fact, the pressure contour lines match the profile of the stress component σ_{22} up to an additive contribution depending on the position x over the channel length, where the pressure profile depicted in Fig. 3.12(b) is already shifted to coincide with σ_{22}. Thus, the desired y-dependent behaviour of the pressure field is obtained by terms of fully developed channel flow configurations as well.



(a) p field

(b) (shifted) p profile

Figure 3.12: Pressure solution for Giesekus model at $\Lambda = 1.0$, $\alpha = 0.1$, $\eta_0 = 1.0$, $\beta = 0.0$, $U = 1.0$

In terms of further investigations it is worked out, that the velocity profiles resulting from one- and two-dimensional configurations show a very good agreement also regarding higher relaxation times or an increased mobility factor. In these cases, also the stress fields consist of straight contour lines over the full channel, while the contour lines of the pressure field again take a curved shape. Similarly, the numerically calculated flow profiles for $\beta > 0$ coincide with the profiles determined in two-dimensional channel flow simulations, since purely straight contour lines for the velocity as well as stress fields are obtained. In summary, the flow profiles determined from the one-dimensional nonlinear system (3.2) coincide with the flow profiles resulting from two-dimensional simulations of Poiseuille flow of a viscoelastic fluid desribed by the Giesekus model. This implies, that the numerically determined flow profiles provide a meaningful choice as Dirichlet boundary conditions even for more complex flows. Moreover, the one-dimensional framework proposed in Sec. 3.1.1 can be used for analysing the properties of flow models in a simplified setting, since the corresponding results are validated in terms of actual two-dimensional simulations. In fact, this is done in Sec. 4.1.1 regarding the initial validation and evaluation of the Tensor Diffusion approach introduced in Sec. 2.4. Furthermore, the corresponding procedure can be extended to covering integral constitutive equations, which is discussed in the next section.

3.2 Integral material laws

In the following, viscoelastic fluid flow models including an integral constitutive equation are considered within the framework of fully developed channel flows to determine corresponding flow profiles similar to the differential case. For deriving an according one-dimensional system of equations, the properties listed in Eq. (3.1) are applied to the stationary integral viscoelastic flow model from Eq. (2.43) including the damping functions given in Eq. (2.15). In contrast to differential models, the Finger tensor $\mathbf{B}$ represents the primal flow variable instead of the extra-stress tensor $\mathbf{\Sigma}$, which is why accordingly $\frac{\partial}{\partial x} B_{ij} = 0,\ i, j \in \{1, 2\}$ holds for the components of the Finger tensor. When considering the evolution equation (2.43d) of the Finger tensor $\mathbf{B}$ for fully developed channel flows, the one-dimensional set of equations

$$\frac{\partial}{\partial s} B_{11}(s) = 2B_{12}(s)\frac{\partial u}{\partial y}, \quad \frac{\partial}{\partial s} B_{12}(s) = B_{22}(s)\frac{\partial u}{\partial y}, \quad \frac{\partial}{\partial s} B_{22}(s) = 0$$

for $s \in [0, \infty[$ is obtained. Together with the initial condition $\mathbf{B}(0) = \mathbf{I}$, the above equations result in analytical expressions for the components of the Finger tensor, in detail

$$B_{22}(s) = 1, \quad B_{12}(s) = s\frac{\partial u}{\partial y}, \quad B_{11}(s) = s^2\left(\frac{\partial u}{\partial y}\right)^2 + 1 \quad \forall s \in [0, \infty[\tag{3.7}$$

Since $\mathbf{B}^{-1}$ can be computed analytically based on the components of $\mathbf{B}$ as well as $\det(\mathbf{B}) \equiv 1$ due to incompressibility [12], inserting Eq. (3.7) into the stress integral from Eq. (2.43b) gives the one-dimensional set of equations

$$\begin{aligned}
0 &= -\eta_s \frac{\partial^2 u}{\partial y^2} - \frac{\partial \sigma_{12}}{\partial y} + \frac{\partial p}{\partial x} && (3.8a)\\
0 &= -\frac{\partial \sigma_{22}}{\partial y} + \frac{\partial p}{\partial y} && (3.8b)\\
\sigma_{11} &= \int_0^\infty m(s)\left(\phi_1(\mathrm{tr}(\mathbf{B}))\left(s^2\left(\frac{\partial u}{\partial y}\right)^2 + 1\right) + \phi_2(\mathrm{tr}(\mathbf{B}))\right)\mathrm{d}s && (3.8c)\\
\sigma_{12} &= \int_0^\infty m(s)\left(\phi_1(\mathrm{tr}(\mathbf{B})) - \phi_2(\mathrm{tr}(\mathbf{B}))\right) s\frac{\partial u}{\partial y}\,\mathrm{d}s && (3.8d)\\
\sigma_{22} &= \int_0^\infty m(s)\left(\phi_1(\mathrm{tr}(\mathbf{B})) + \phi_2(\mathrm{tr}(\mathbf{B}))\left(s^2\left(\frac{\partial u}{\partial y}\right)^2 + 1\right)\right)\mathrm{d}s && (3.8e)
\end{aligned}$$

based the original stationary integral model. Furthermore, the trace of the Finger tensor evolves to

$$\operatorname{tr}(\mathbf{B}) = B_{11} + B_{22} = s^2 \left(\frac{\partial u}{\partial y}\right)^2 + 2 \tag{3.9}$$

which is why $\phi_{1,2} = \phi_{1,2}(\operatorname{tr}(\mathbf{B})) = \phi_{1,2}\left(s, \frac{\partial u}{\partial y}\right)$ in Eq. (3.8c), (3.8d) and (3.8e). Thus, from Eq. (3.8) it is realised, that the components of the extra-stress tensor $\mathbf{\Sigma}$ are written explicitly depending on $\frac{\partial u}{\partial y}$ and do not have to be considered as numerical variables. Furthermore, the fully developed flow profiles are mainly determined based on Eq. (3.8a) according to the differential case, which provides a nonlinear relation of the velocity field and the pressure drop over the channel length. Hence, by inserting Eq. (3.8d) into the momentum equation (3.8a) and taking $\frac{\partial u}{\partial y}$ out of the time integral, the fully developed velocity profile regarding integral viscoelastic models is determined by

$$\begin{aligned} & \frac{\partial p}{\partial x} - \eta_s \frac{\partial^2 u}{\partial y^2} \cdots \\ \cdots - \frac{\partial}{\partial y}\left(\left[\int_0^\infty m(s)\left(\phi_1\left(s, \frac{\partial u}{\partial y}\right) - \phi_2\left(s, \frac{\partial u}{\partial y}\right)\right) s \, \mathrm{d}s\right] \frac{\partial u}{\partial y}\right) &= 0 \end{aligned} \tag{3.10}$$

supplemented with Eq. (3.2f) specifying the desired flow rate. Thus, the integral viscoelastic flow model is reduced to a nonlinear problem in the unknowns $\left(\frac{\partial u}{\partial y}, \frac{\partial p}{\partial x}\right)$, where the stress profiles are calculated in terms of post-processing only.

Note, that many difficulties arising in the context of the original Deformation Fields Method, which are indicated in Sec. 2.3, are irrelevant or not present in the given setting. For example, the positive definiteness as well as incompressibility of the fields are satisfied automatically by considering fully developed channel flows. Moreover, no accumulating error regarding the time-approximation of $\mathbf{B}$ needs to be taken into account, since the corresponding analytical solution is known. However, it is confirmed by means Eq. (3.7), that the Deformation Fields $\mathbf{B}$ indeed do not reach a stationary state even in case of a velocity field independent of the time variable as mentioned in Ref. [13]. Nevertheless, the extra-stress tensor $\mathbf{\Sigma}$ from the integral expression (2.43b) reaches a finite value, since $\mathbf{B}$ only grows quadratically in s. At the same time, the damping functions from Eqs. (3.14) and (3.20) obtained for the invariant in Eq. (3.9) as well as the memory function $m(s) = \frac{\eta_p}{\Lambda^2}\exp\left(-\frac{s}{\Lambda}\right)$ decay with increasing time $s \in [0, \infty[$, the latter even exponentially. In addition, determining fully developed flow profiles regarding integral constitutive equations in case of the multi-mode approach is conceptually even simpler than compared to differential models. In detail, multiple modes are considered by inserting the memory function m from Eq. (2.10) into Eq. (3.10). Thus, only the evaluation of the former stress integral is affected when considering multiple modes, while the overall problem formulation stays unchanged.

In the following, fully developed velocity profiles are calculated based on Eq. (3.10) according to the UCM, PSM, Wagner and Wagner-Demarmels model. Therefore, the specific damping functions from Eq. (2.15) are inserted into the governing equation (3.10), which results in conceptually varying solution approaches for each model. At first, flow profiles regarding the UCM model are considered, which can be derived analytically according to the differential case. In this regard, $\phi_1 \equiv 1$, $\phi_2 \equiv 0$ and $m(s) = \frac{\eta_p}{\Lambda^2} \exp\left(-\frac{s}{\Lambda}\right)$ are applied in Eq. (3.10), which gives

$$\begin{aligned} 0 &= \frac{\partial p}{\partial x} - \eta_s \frac{\partial^2 u}{\partial y^2} - \frac{\partial}{\partial y}\left(\left[\int_0^\infty \frac{\eta_p}{\Lambda^2} \exp\left(-\frac{s}{\Lambda}\right) s \, \mathrm{d}s\right] \frac{\partial u}{\partial y}\right) \\ &= \frac{\partial p}{\partial x} - \eta_s \frac{\partial^2 u}{\partial y^2} - \frac{\partial}{\partial y}\left(\eta_p \frac{\partial u}{\partial y}\right) = \frac{\partial p}{\partial x} - \eta_0 \frac{\partial^2 u}{\partial y^2} \end{aligned} \tag{3.11}$$

including $\sigma_{12} = \eta_p \frac{\partial u}{\partial y}$. In addition, the same parabolic velocity profile as in Eq. (3.3) is obtained for a channel height of $y \in [a, b]$ and $u(a) = (b) = 0$. Actually, the Finger tensor $\mathbf{B}$ is often replaced by $\mathbf{B} - \mathbf{I}$ in the stress integral (2.43b) when referring to the UCM model, which does not affect the nonlinear problem in Eq. (3.10) or the solution of Eq. (3.11). But, the components σ_{11} and σ_{22} from Eq. (3.8c) and (3.8e) result in

$$\sigma_{11} = \int_0^\infty \frac{\eta_p}{\Lambda^2} \exp\left(-\frac{s}{\Lambda}\right) s^2 \, \mathrm{d}s \left(\frac{\partial u}{\partial y}\right)^2, \quad \sigma_{22} = 0 \tag{3.12}$$

Hence, $\sigma_{11} = 2\eta_p \Lambda \left(\frac{\partial u}{\partial y}\right)^2$ is directly obtained, which completes the results of the differential version of the UCM model from Eq. (3.4). Moreover, considering $\mathbf{B} - \mathbf{I}$ instead of $\mathbf{B}$ causes a vanishing pressure function f_p, which is introduced in terms of differential constitutive equations. Accordingly, $f_p = \sigma_{22}$ is satisfied also based on the one-dimensional form of the general integral viscoelastic model when taking into account Eq. (3.8b). But, a y-dependent behaviour of the pressure is introduced in case of non-trivial damping functions $\phi_{1,2}$, for example regarding the PSM or Wagner model as outlined below. Furthermore, the two components σ_{11} and σ_{22} of the extra-stress tensor are related in this context according to

$$\sigma_{11} = \int_0^\infty m(s)\, \phi_1\left(s, \frac{\partial u}{\partial y}\right) s^2 \left(\frac{\partial u}{\partial y}\right)^2 \mathrm{d}s + \sigma_{22} \tag{3.13a}$$

$$\sigma_{22} = \int_0^\infty m(s)\, \phi_1\left(s, \frac{\partial u}{\partial y}\right) \mathrm{d}s \tag{3.13b}$$

based on Eqs. (3.8c) and (3.8e). Consequently, the extra-stress tensor is shifted compared to the UCM-solution by a diagonal matrix with entries f_p varying over the channel height. Hence, the stress components are presented below in terms of the PSM or Wagner model excluding the contribution of σ_{22} or f_p from Eq. (3.13). By doing so, a reasonable comparison of the results regarding the UCM model is drawn, since especially $\sigma_{11} = 0$ is obtained on the centre line of the channel. For completeness, the profile of the pressure function f_p is depicted below as well to illustrate the y-dependent contribution in the pressure field. However, an analytical form of the velocity profile – and thus also of the stresses – resulting from more complex models like the PSM or Wagner(-Demarmels) model can not be derived. Consequently, the (nonlinear) equation (3.10) has to be solved numerically, which is discussed in the following.

3.2.1 The PSM model

Compared to the UCM model, more complex integral constitutive equations are considered by choosing non-trivial damping functions $\phi_{1,2}$ in the stress integral from Eq. (2.43b). For example, the damping function in case of the PSM model evolves to

$$\phi_1 = \frac{1}{1 + \gamma s^2 \left(\frac{\partial u}{\partial y}\right)^2} \tag{3.14}$$

when inserting Eq. (3.9), that is the trace of the Finger tensor resulting in two-dimensional fully developed channel flows, into Eq. (2.15a). Consequently, the basic equation (3.10) for calculating fully developed integral flow profiles gives

$$\frac{\partial p}{\partial x} - \eta_s \frac{\partial^2 u}{\partial y^2} - \frac{\partial}{\partial y} \left(\int_0^\infty \frac{\eta_p}{\Lambda^2} \exp\left(-\frac{s}{\Lambda}\right) \frac{s}{1 + \gamma s^2 \left(\frac{\partial u}{\partial y}\right)^2} \mathrm{d}s \frac{\partial u}{\partial y} \right) = 0 \tag{3.15}$$

which is a nonlinear equation with respect to $\frac{\partial u}{\partial y}$. Note, that considering the Wagner-Demarmels model, that is the damping functions from Eq. (2.15c), for fully developed channel flows leads to Eq. (3.15) as well. Thus, the same velocity profile as in case of the PSM model is obtained, while the corresponding stress profiles will be of different shape. Recall, that these quantities are determined in pure post-processing fashion, which is why the Wagner-Demarmels model will be not investigated here. However, the integral expression occurring in Eq. (3.15) can not be given in closed form, which is why a corresponding approximation needs to be taken into account within the numerical approach to solve for the unknown quantities $\frac{\partial p}{\partial x}$ and $\frac{\partial u}{\partial y}$ or u. Therefore, the viscosity

$$\mu^{\mathrm{PSM}} \left(\frac{\partial u}{\partial y}\right) := \int_0^\infty \frac{\eta_p}{\Lambda^2} \exp\left(-\frac{s}{\Lambda}\right) \frac{s}{1 + \gamma s^2 \left(\frac{\partial u}{\partial y}\right)^2} \mathrm{d}s \tag{3.16}$$

is defined, whereby the original equation (3.15) for calculating the fully developed velocity profile for the PSM model may be rewritten as

$$\frac{\partial p}{\partial x} - \eta_s \frac{\partial^2 u}{\partial y^2} - \frac{\partial}{\partial y} \left[\mu^{\mathrm{PSM}} \left(\frac{\partial u}{\partial y}\right) \frac{\partial u}{\partial y} \right] = 0 \tag{3.17}$$

Obviously, Eq. (3.17) is of similar type as (symmetric) Sturm-Liouville problems, which is why an according discretisation technique in terms of a Finite Difference approximation may be applied here. Similar to the differential case, discrete grid points $y_k = a + k\frac{b-a}{n}$, $k = 0, \ldots, n$, are spreaded over the channel height $y \in [a, b]$, based on which the derivatives occurring in Eq. (3.17) are discretised. By defining the operator

$$\Delta_h v\left(y\right) = \frac{v\left(y + h\right) - v\left(y - h\right)}{2h}$$

regarding central Finite Differencing, the derivative in Eq. (3.17) involving the newly defined viscosity μ^{PSM} is approximated by

$$\begin{aligned}
\frac{\partial}{\partial y}\left[\mu^{\mathrm{PSM}}\frac{\partial u}{\partial y}\right] &\approx \Delta_{\frac{h}{2}}\left[\mu_k^{\mathrm{PSM}}\Delta_{\frac{h}{2}}u_k\right] \\
&= \frac{1}{h^2}\left[\mu_{k+\frac{1}{2}}^{\mathrm{PSM}}u_{k+1} - \left(\mu_{k+\frac{1}{2}}^{\mathrm{PSM}} + \mu_{k-\frac{1}{2}}^{\mathrm{PSM}}\right)u_k + \mu_{k-\frac{1}{2}}^{\mathrm{PSM}}u_{k-1}\right] \quad (3.18)
\end{aligned}$$

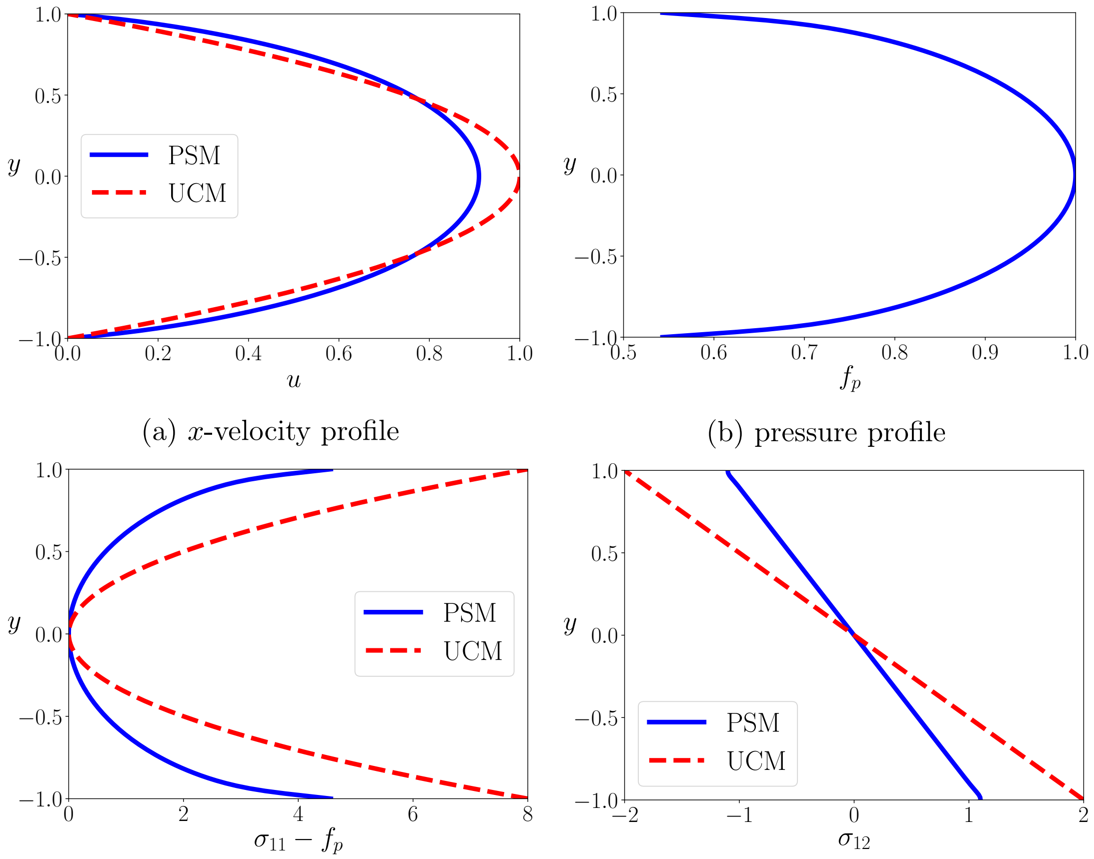
where $u_k := u\left(y_k\right)$, $k = 0, \ldots, n$, denote the discrete unknowns regarding the velocity field u, while the discrete values of the viscosity are defined via

$$\mu_{k\pm\frac{1}{2}}^{\mathrm{PSM}} \approx \frac{1}{2}\left(\mu_k^{\mathrm{PSM}} + \mu_{k\pm1}^{\mathrm{PSM}}\right), \quad \mu_k^{\mathrm{PSM}} = \mu^{\mathrm{PSM}}\left(\frac{u_{k+1} - u_{k-1}}{2h}\right) \tag{3.19}$$

Hence, the viscosity in an intermediate grid point $y_{k\pm\frac{1}{2}}$ is set according to the average value regarding the two neighbouring actual grid points. Finally, a nonlinear system with respect to the unknowns u_k is obtained from Eq. (3.17) by applying the discretisation proposed in Eq. (3.18) together with a Finite Difference approximation of second order concerning the solvent contribution.

Since evaluating the viscosity μ^{PSM} involves "infinite" numerical integration, it makes sense to decouple this step from actually solving the discrete version of Eq. (3.17). Therefore, the viscosity μ^{PSM} from Eq. (3.16) is evaluated for given discrete values u_k according to Eq. (3.19) in the first step of the decoupled solution approach. Thereby, μ^{PSM} is integrated numerically in the actual implementation as long as corresponding contributions are larger than a specified tolerance. In a second step, the discrete versions of Eq. (3.17) as well as (3.2f) regarding the flow rate are solved for an update solution of u_k, $k = 0, \ldots, n$, and $\frac{\partial p}{\partial x}$ for given μ_k^{PSM}. Basically, the resulting numerical solution approach consists of alternately evaluating μ^{PSM} from Eq. (3.16) according to Eq. (3.19) and solving Eq. (3.17) for the discrete values of u until the (relative) changes in the velocity field as well as the viscosity fall below a certain tolerance. In fact, this problem formulation or solution approach can be interpreted as a one-dimensional equivalent of the Tensor Diffusion proposed in Sec. 2.4, where the Diffusion Tensor is determined in an intermediate step of a decoupled scheme. In contrast, a fully coupled treatment of the discrete unknowns of the velocity field for solving Eq. (3.10) can be established in terms of the Wagner model (see Sec. 3.2.2).

Numerical results concerning the quantities u, σ_{11} and σ_{12} for the non-solvent case at $\Lambda = 1.0$ and $\gamma = 0.1$ are given in Fig. 3.13. In principle, the flow profiles show a similar behaviour as the results obtained for the Giesekus model, that is a shear thinning effect is observed regarding the velocity profile depicted in Fig. 3.13(a) and the stress profiles deviate accordingly from the solutions of the UCM model. In addition, a curved pressure contour line is obtained, which is indicated by means of the pressure function f_p in Fig. 3.13(b). However, the numerical method described above does not yield a solution for significantly larger relaxation times than $\Lambda = 1$ due to a breakdown of the calculations, which is investigated further below in case of the Wagner model. The corresponding studies give the impression, that conceptual limitations (also) of the PSM model are responsible for the failure of the numerical simulations. Probably due to the same reason, flow profiles can not be obtained for higher model parameters γ as well. Nevertheless, varying the model parameter γ occurring in the damping function from Eq. (3.14) has the same effect as the corresponding variation of the mobility factor α in the Giesekus model. In detail, a higher value of γ intensifies the developed shear thinning effect for a given relaxation time Λ.



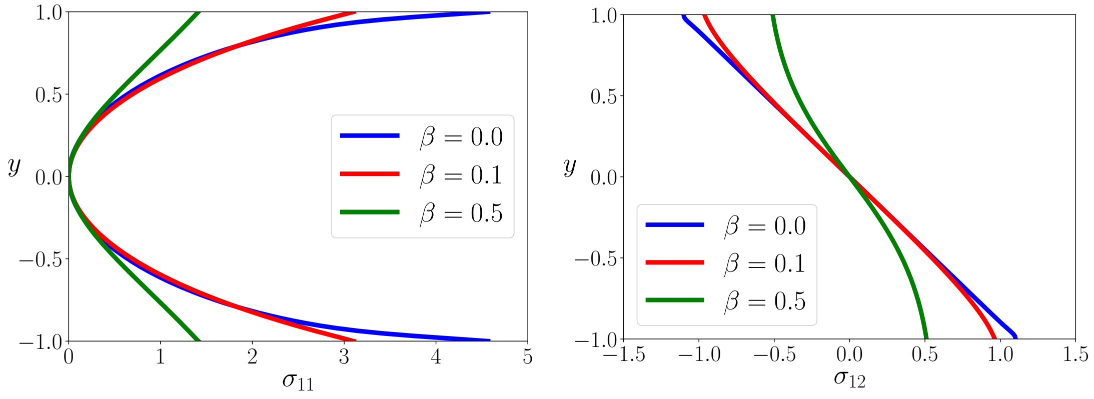
(a) x-velocity profile

(b) pressure profile

(c) profile of $\sigma_{11} - f_p$

(d) σ_{12} profile

Figure 3.13: Flow profiles for the PSM model at $\Lambda = 1.0$, $\gamma = 0.1$, $\eta_0 = 1.0$, $\beta = 0.0$, $U = 1.0$



(a) σ_{11} profiles

(b) σ_{12} profiles

Figure 3.14: Stress profiles for the PSM model at $\Lambda = 1.0$, $\gamma = 0.1$, $\eta_0 = 1.0$, $U = 1.0$ for several amounts β of solvent viscosity

Also in case of a present solvent contribution, the velocity profiles show a behaviour similar to the Giesekus model, since a parabolic shape is recovered for $\beta \to 1$. But in contrast to the results depicted in Fig. 3.9, the stress profiles corresponding to the PSM model do not consist of such a large gradient close to the centre line of the channel. Actually, a present solvent contribution damps the growth of the stress variable causing a much smoother transition to the Newtonian case (see Fig. 3.14(a) and 3.14(b)), while the stress profiles regarding the Giesekus model seem to converge to a nearly discontinuous behaviour in $y = 0$ for $\beta \to 1$.

In summary, the fully developed flow profiles determined in case of the PSM model behave similar to the (differential) Giesekus model under a variation of the material and model parameters, which indicates that reasonable results are obtained for this integral constitutive equation as well. But, no validation in terms of two-dimensional simulations can be performed, since a corresponding numerical framework for simulating integral viscoelastic fluid flows is not realised yet. Thus, the calculated fully developed flow profiles can not be prescibed as boundary data in actual two-dimensional configurations. But, according investigations are performed in terms of the initial validation of the Tensor Diffusion approach in Sec. 4.1.1, where the Wagner model discussed next is taken into account as well. In fact, it is aimed at realising the numerical treatment of integral viscoelastic models by means of the Tensor Diffusion approach as part of future research work, particularly with respect to complex two-dimensional configurations.

3.2.2 The Wagner model

In the following, the framework of fully developed channel flows is transferred to the Wagner model, which allows a further investigation on the general properties of this integral constitutive equation. In doing so, the non-trivial damping function

$$\phi_1 = f \exp\left(-sn_1\sqrt{\left(\frac{\partial u}{\partial y}\right)^2}\right) + (1-f)\exp\left(-sn_2\sqrt{\left(\frac{\partial u}{\partial y}\right)^2}\right) \tag{3.20}$$

is obtained by inserting the trace of the Finger tensor $\mathbf{B}$ from Eq. (3.9) into the damping function given in Eq. (2.15b). For simplicity, $f = 1$ is considered in the following theoretical remarks, as results for the convex combination in Eq. (3.20) follow directly. In contrast, numerical examples are usually presented based on the nonlinear material parameters given in Ref. [17], including $f \in]0,1[$.

Similar to the PSM model, in a first step the damping function from Eq. (3.20) is inserted into the basic equation (3.10) for calculating fully developed flow profiles regarding integral viscoelastic models, which gives

$$\begin{aligned}\frac{\partial p}{\partial x} - \eta_s\frac{\partial^2 u}{\partial y^2} &= \frac{\partial}{\partial y}\left(\int_0^\infty m(s)\exp\left(-sn_1\sqrt{\left(\frac{\partial u}{\partial y}\right)^2}\right) s\,\mathrm{d}s\frac{\partial u}{\partial y}\right)\\ &= \frac{\partial}{\partial y}\left(\int_0^\infty \frac{\eta_p}{\Lambda^2}\exp\left(-s\left(\frac{1}{\Lambda} + n_1\sqrt{\left(\frac{\partial u}{\partial y}\right)^2}\right)\right) s\,\mathrm{d}s\frac{\partial u}{\partial y}\right)\end{aligned}$$

since $m(s) = \frac{\eta_p}{\Lambda^2} \exp\left(-\frac{s}{\Lambda}\right)$. In contrast to the PSM model, the nonlinear viscosity function arising for the Wagner model can be given in closed form, in detail

$$\begin{aligned}\mu^{\text{Wag}} &= \int_0^\infty \frac{\eta_p}{\Lambda^2} \exp\left(-s\left(\frac{1}{\Lambda} + n_1\sqrt{\left(\frac{\partial u}{\partial y}\right)^2}\right)\right) s\,\mathrm{d}s \\ &= \eta_p\left(1 + n_1\Lambda\sqrt{\left(\frac{\partial u}{\partial y}\right)^2}\right)^{-2} \qquad (3.21)\end{aligned}$$

Consequently, Eq. (3.10) evolves to the nonlinear equation

$$\frac{\partial p}{\partial x} - \eta_s\frac{\partial^2 u}{\partial y^2} - \eta_p\frac{\partial}{\partial y}\left(\left(1 + n_1\Lambda\sqrt{\left(\frac{\partial u}{\partial y}\right)^2}\right)^{-2}\frac{\partial u}{\partial y}\right) = 0 \qquad (3.22)$$

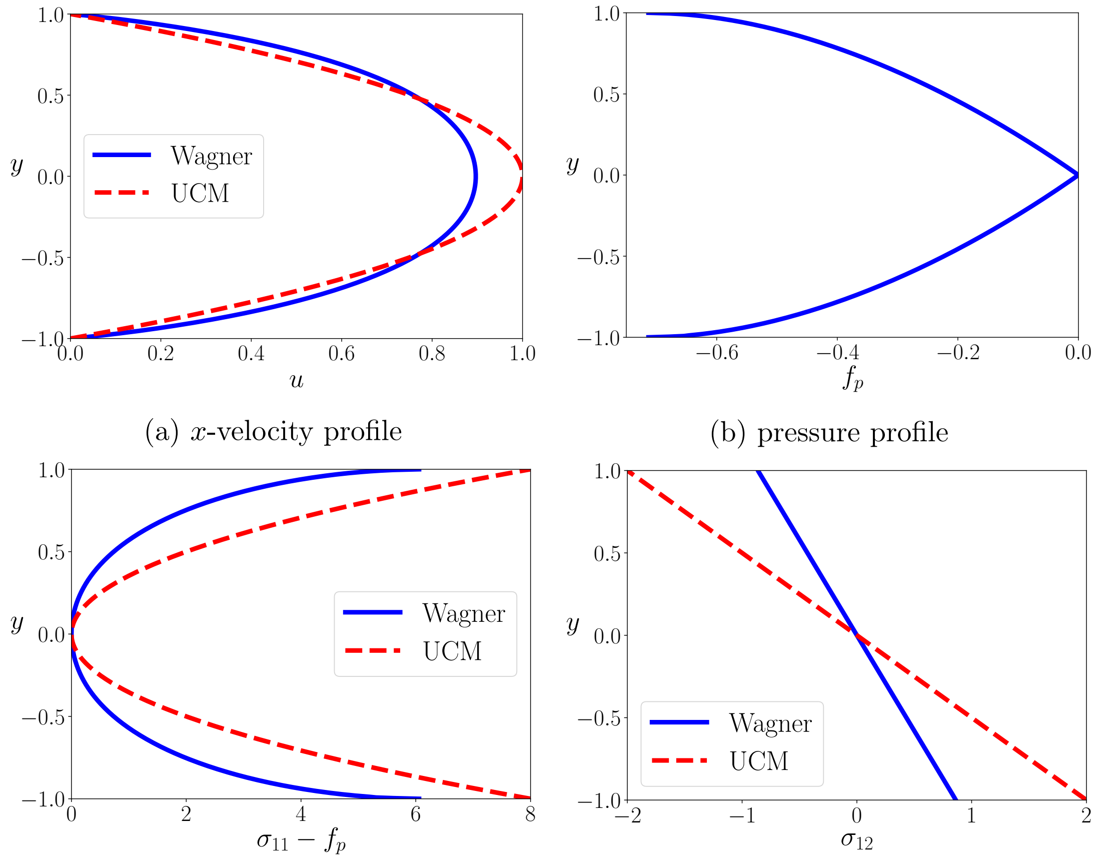
in case of the Wagner model at $f = 1$. Note, that the corresponding equation resulting for $f \in]0,1[$ is obtained by replacing the viscosity contribution in Eq. (3.22) by a convex combination of viscosities of the type presented in Eq. (3.21) according to Eq. (3.20). Moreover, the above problem formulation can be regarded as a one-dimensional analog of the pure Tensor Stokes problem from Eq. (2.49), where the integral constitutive equation is indeed replaced by an explicit expression of the Diffusion Tensor, which is investigated further in Sec. 4.1.1. In addition, the multi-mode approach might be included by defining the viscosity μ^{Wag} as a sum of terms of the form presented in Eq. (3.21) for separate parameters $\eta_{p,k}$ and Λ_k.

In principle, Eq. (3.22) (or the version corresponding to $f \in]0,1[$) can be treated numerically in a similar way as the PSM model, that is based on a Finite Difference discretisation according to Eq. (3.18). But in contrast to the PSM model, the nonlinear system arising from Eqs. (3.22) and (3.2f) regarding the pressure drop $\frac{\partial p}{\partial x}$ and the discrete unknowns u_k of the velocity field is solved monolithically. In detail, the intermediate decoupled step of evaluating the viscosity may be omitted in case of the Wagner model by explicitly considering the approximation

$$\mu_k^{\text{Wag}} := \mu^{\text{Wag}}\left(\Delta_{\frac{h}{2}} u_k\right) = \left(1 + n_1\Lambda\sqrt{\left(\frac{u_{k+\frac{1}{2}} - u_{k-\frac{1}{2}}}{h}\right)^2}\right)^{-2} \qquad (3.23)$$

in the discrete version of Eq. (3.22). Thus, the nonlinear character of Eq. (3.22) is kept in the discrete setting, since an actual nonlinear equation regarding the discrete unknowns u_k of the velocity field is derived when taking into account Eq. (3.18) with respect to μ_k^{Wag} from Eq. (3.23) instead of μ_k^{PSM}.

Numerical results concerning the Wagner model for the parameters $f = 0.57$, $n_1 = 0.31$ and $n_2 = 0.106$ given in Ref. [17] as well as $\beta = 0$ are presented in Fig. 3.15. In principle, the flow profiles behave similarly to the PSM model at $\Lambda = 1.0$ and $\gamma = 0.1$ depicted in Fig. 3.13. In detail, the velocity profile shows a clear shear thinning effect and also the stress profiles deviate significantly from the solution regarding the UCM model.



(a) x-velocity profile

(b) pressure profile

(c) profile of $\sigma_{11} - f_p$

(d) σ_{12} profile

Figure 3.15: Flow profiles for the Wagner model at $\Lambda \approx 1.6$, $f = 0.57$, $n_1 = 0.31$, $n_2 = 0.106$, $\eta_0 = 1.0$, $\beta = 0.0$, $U = 1.0$

But apparently, the quantity f_p depicted in Fig. 3.15(b) characterising the y-dependent behaviour of the pressure field shows an obvious difference to the corresponding quantity arising from the PSM model (see Fig. 3.13(b)). In fact, f_p is not smooth in $y = 0$ in case of the Wagner model, which can realised by computing $f_p = \sigma_{22}$ based on Eq. (3.8e) for the damping function from Eq. (3.20), which gives

$$\sigma_{22}^{\text{Wag}} = \frac{\eta_p}{\Lambda}\left(f \left(1 + n_1 \Lambda \sqrt{\left(\frac{\partial u}{\partial y}\right)^2} \right)^{-1} + (1-f) \left(1 + n_2 \Lambda \sqrt{\left(\frac{\partial u}{\partial y}\right)^2} \right)^{-1} \right)$$

Thus, σ_{22} or f_p includes the absolute value of $\frac{\partial u}{\partial y}$, which causes the non-smoothness in $y = 0$. However, in case of a present solvent viscosity the velocity profile evolves again towards a parabolic shape similar to the PSM as well as Giesekus model, which is realised by means of further numerical studies. Also the stress profiles arising from the Wagner model at $\beta > 0$ look similar to the previous results, as the magnitude of the stresses decreases for increasing β. But, the Newtonian limit $\mathbf{\Sigma} = 0$ is not approached as smooth as in case of the PSM model, since the component σ_{22} of the extra-stress tensor consists of a kink at $y = 0$ for all $\beta \in [0, 1[$, similar to f_p from Fig. 3.15(b) .

As already mentioned in terms of the PSM model, for the given setting it is not possible to compute flow profiles regarding the Wagner model at significantly larger relaxation times than $\Lambda = 1.5$. Especially concerning the simulation of industrial applications, successful numercial computations for significantly larger relaxation times are of intensified interest due to the properties of the processed materials, for example polymer melts. Consequently, the source of the failure in obtaining corresponding numerical results is investigated deeper in terms of the present – quite simple – setting. A typical guess would be, that the discretisation is not able to capture the large gradient of the velocity or stress fields occurring due to a pronounced shear thinning effect. But instead, the flow profiles depicted in Fig. 3.15 for the maximum relaxation time of $\Lambda_{\text{crit}} \approx 1.6$ only show a moderate growth close to the channel walls, which is probably resolved sufficiently accurate by means of the applied spatial resolution. Moreover, a higher mesh refinement even leads to a decreasing maximum relaxation time for successfully solving the nonlinear problem from Eq. (3.22). Thereby, the relaxation time is increased incrementally, where the previous solution is used as starting value for computing the next higher Λ, which should provide suitable initial solutions in terms of Newton's method. But, the nonlinear solver is not converging even for very small increments of Λ. In addition, performing damped Newton steps, which would lead to a globalised convergence behaviour by including an (adaptive) step length control when updating the solution vector, does not result in a converging numerical scheme either. The same holds in case of applying a simple fixed-point iteration. Similarly, modifying the discretisation approach by taking into account a Finite Difference approximation of higher order does not resolve this issue, which is furthermore observed in two-dimensional Finite Element simulations as well (see Sec. 4.1.1). Thus, the Wagner model seems to consist of underlying mathematical properties, which prevent successful computations at higher Λ, since adapting the solution as well as discretisation techniques does not improve the numerical scheme. As expected, inserting diffusion into the system by setting $\eta_s > 0$ easily leads to solutions for higher Λ, whereas the non-solvent case is of intensified interest, since it provides the most challenging configuration.

Luckily, the nonlinear problem corresponding to the Wagner model is given in closed form, which is why a more detailed analysis can be performed compared to the PSM model. In fact, the relation

$$\frac{\partial u}{\partial y} = \frac{\partial p}{\partial x} \frac{y\left(1 + n_1\Lambda\sqrt{\left(\frac{\partial u}{\partial y}\right)^2}\right)^2\left(1 + n_2\Lambda\sqrt{\left(\frac{\partial u}{\partial y}\right)^2}\right)^2}{\eta_p\left(f\left(1 + n_2\Lambda\sqrt{\left(\frac{\partial u}{\partial y}\right)^2}\right)^2 + (1-f)\left(1 + n_1\Lambda\sqrt{\left(\frac{\partial u}{\partial y}\right)^2}\right)^2\right)} \tag{3.24}$$

concerning the derivative of the velocity field can be derived by integrating Eq. (3.22) for $f \in]0,1[$ and $\eta_s = 0$ with respect to $y \in [-1,1]$. Note, that the constant arising during integration vanishes in the considered setting, since $\frac{\partial u}{\partial y} = 0$ for $y = 0$. Naturally, considering $y \in [a,b]$ for arbitrary $a, b \in \mathbb{R}$, where $a < b$, will not affect the following remarks, since only slight adaptions are necessary for a corresponding generalisation. Thereby, the y-term in the enumerator of Eq. (3.24) is replaced by $y - \frac{b+a}{2}$, again giving $\frac{\partial u}{\partial y} = 0$ in the centre of the channel, which does not lead to major changes in the structure of the problem. However, solving

Eq. (3.24) for $\frac{\partial u}{\partial y}$ can be interpreted as finding the roots of a quadratic polynomial when again considering $f = 1$. Thus, the condition

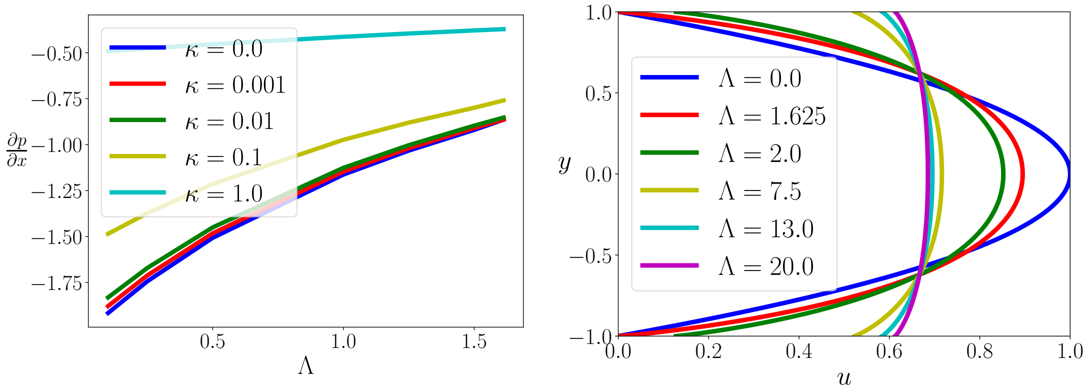
$$1 - 4\frac{\partial p}{\partial x} n_1 \frac{\Lambda}{\eta_p} \operatorname{sgn}\left(\frac{\partial u}{\partial y}\right) y \geq 1 - 4\left|\frac{\partial p}{\partial x}\right| n_1 \frac{\Lambda}{\eta_p} \geq 0 \tag{3.25}$$

can be derived to guarantee the existence of real polynomial roots, that is real solutions concerning $\frac{\partial u}{\partial y}$. Thereby, $\frac{\partial p}{\partial x} \equiv \text{const} < 0$ and $\operatorname{sgn}\left(\frac{\partial u}{\partial y}\right) = -\operatorname{sgn}(y)$ is taken into account in the above inequality. Moreover, a similar condition can be derived regarding $f \in]0,1[$, which gives a polynomial of degree four in $\frac{\partial u}{\partial y}$. Consequently, the original nonlinear equation (3.22) for calculating fully developed flow profiles in case of the Wagner model at $f = 1$ can be solved only, if the chosen parameter set satisfies Eq. (3.25), which is violated at $\Lambda > \Lambda_{\text{crit}}$ for the parameters given above. Furthermore, Eq. (3.25) explains the descreasing maximum relaxation time for $\eta_s = 0$ in case of increasing mesh refinement: In numerical experiments it is observed, that the magnitude of $\frac{\partial p}{\partial x}$ at fixed Λ increases according to the spatial resolution until convergence is reached. Hence, Eq. (3.25) is satisfied in case of increasing mesh refinement only by considering accordingly decreasing relaxation times Λ.

However, introducing $\eta_s > 0$ yields root-finding of a polynomial of degree three for $f = 1$ or degree five for $f \in]0,1[$ in $\frac{\partial u}{\partial y}$, which thus guarantees at least one real solution of Eq. (3.22). Hence, computing fully developed flow profiles regarding the Wagner model should be possible for any parameter set $(f, n_1, n_2, \Lambda, \eta_0)$ including $\eta_s > 0$. In contrast, the limits of the non-solvent Wagner model in the current setting can be weakened, when so-called slip boundary conditions are applied with respect to the velocity field on the channel walls. In doing so, the absolute value of the pressure drop – for example resulting with respect to Poiseuille flow – is reduced compared to the no-slip setting, which allows to consider larger relaxation times for satisfying Eq. (3.25). Following Ref. [9], the simple linear Navier slip law reads

$$u = \kappa \frac{\partial u}{\partial y} \tag{3.26}$$

where the parameter $\kappa \in [0, \infty[$ is related to the amount of slip prescribed on the boundary. For $\kappa = 0$, the typical no-slip boundary condition is recovered, while $\kappa \to \infty$ leads to the full-slip or Neumann boundary condition. In the numerical framework presented above, the slip boundary condition from Eq. (3.26) can be implemented within the discretisation of the nonlinear problem from Eq. (3.22) by making use of a suitable forward Finite Differencing in the grid points on the channel wall. It turns out, that $\kappa > 0$ indeed allows the computation of flow profiles corresponding to relaxation times larger than the critical value of $\Lambda \approx 1.6$ obtained for $\kappa = 0$. As realised from Fig. 3.16(a), increasing the slip parameter κ for a fixed relaxation time Λ leads to a proportionally reduced pressure drop in the channel. Thus, in principle “arbitrary” high relaxation times can be computed by means of the Wagner model for accordingly increasing κ, where velocity profiles for increasing Λ (and thus also κ) are depicted in Fig. 3.16(b). Obviously, a constant velocity profile is obtained for $\Lambda, \kappa \to \infty$, which would correspond to infinite shear thinning including full slip.



(a) pressure drop over Λ (and κ)

(b) x-velocity profiles

Figure 3.16: Flow quantities for the Wagner model at $f = 0.57$, $n_1 = 0.31$, $n_2 = 0.106$, $\eta_0 = 1.0$, $\beta = 0.0$, $U = 1.0$ for increasing relaxation times Λ and slip parameters κ

Following Ref. [9], the Giesekus model discussed in Sec. 3.1 consists of structural limits similar to the Wagner model. This is realised by means of the analytical expression

$$\frac{\partial u}{\partial y} = 2\alpha \frac{\Lambda}{\eta_p} \frac{\partial p}{\partial x} y \frac{1 \pm (2\alpha - 1)\sqrt{1 - 4\alpha^2 \frac{\Lambda^2}{\eta_p^2} \left(\frac{\partial p}{\partial x} y\right)^2}}{\left(2\alpha - 1 \pm \sqrt{1 - 4\alpha^2 \frac{\Lambda^2}{\eta_p^2} \left(\frac{\partial p}{\partial x} y\right)^2}\right)^2} \tag{3.27}$$

regarding the derivative of the velocity field, which can be derived from Eq. (3.2) in case of $\eta_s = 0$. Obviously, the relation in Eq. (3.27) is not well-defined, once the radicand in the denominator becomes negative or the whole denominator vanishes, while the former turns out to be more critical. Consequently, the inequality

$$1 - 4\alpha^2 \frac{\Lambda^2}{\eta_p^2} \left(\frac{\partial p}{\partial x}\right)^2 \geq 0 \tag{3.28}$$

needs to be satisfied by the chosen parameter set in case of the Giesekus model, which is a condition similar to Eq. (3.25) for the Wagner model. However, a "typical" mobility factor of $\alpha = 0.1$ is considered in Sec. 3.1 when analysing the flow profiles of the Giesekus model for increasing relaxation times Λ. In addition, the pressure drop $\frac{\partial p}{\partial x}$ decreases for increasing Λ, which is why the radicand will be close to 1 for a wide relaxation time spectrum. But, no solutions corresponding to $\Lambda = 1.0$ and mobility factors significantly larger than $\alpha \approx 0.5$ can be obtained (see Fig. 3.6), while choosing a mobility factor of $\alpha = 1$ leads to a non-solvable Giesekus model already at $\Lambda_{\text{crit}} \approx 0.325$. Again, the observed shear thinning effect is not very pronounced, but calculating the flow profiles including no-slip boundary conditions for higher relaxation times still fails, probably due to violating Eq. (3.28). A remedy for computing flow profiles regarding higher relaxation times for the non-solvent case of the Giesekus model is provided by introducing slip boundary conditions. Similar to the Wagner model, the slip parameter κ needs to be increased proportionally to the relaxation time for obtaining solutions from the one-dimensional system from Eq. (3.2) at higher Λ.

Instead of introducing slip, inserting a solvent contribution into the Giesekus model modifies the inequality from Eq. (3.28), such that

$$1 - 4\alpha^2 \frac{\Lambda^2}{\eta_p^2} \left(\left| \frac{\partial p}{\partial x} \right| |y| - \eta_s \left| \frac{\partial u}{\partial y} \right| \right)^2 \geq 0 \tag{3.29}$$

needs to be satisfied for successfully calculating the derivative of the velocity field. Thus, the criterion regarding the parameter set $(\alpha, \Lambda, \eta_p)$ is weakened by considering $\eta_s > 0$ similar to the Wagner model, which is why higher relaxation times can be considered. When only a small amount of solvent contribution is taken into account, that is $0 < \beta \ll 1$, where $\eta_s = \beta\eta_0$ and $\eta_p = (1-\beta)\,\eta_0$, no solution for relaxation times Λ significantly larger than the critical value Λ_{crit} can be computed, since the inequality from Eq. (3.28) is weakened only a little. Thus, the condition from Eq. (3.29) can be violated during the start-up of inserting $\eta_s > 0$. In contrast, the velocity profile converges towards the parabolic shape in case of $\beta \to 1$ leading to $\eta_s \left| \frac{\partial u}{\partial y} \right| \to \left| \frac{\partial p}{\partial x} \right| |y|$, which is obtained faster than $\eta_p \to 0$ due to the velocity profile being nearly parabolic already at $\beta \approx 0.5$ (see Fig. 3.8(c)). Consequently, the left-hand side of Eq. (3.29) converges to 1, which is why fully developed flow profiles for arbitrarily high relaxation times can be computed, in case of a sufficiently large amount of solvent contribution is considered.

The above results regarding slip boundary conditions are obtained in preliminary numerical tests only, while a more detailed analysis is part of future work. Nevertheless, based on the inequalities from Eqs. (3.25) and (3.28) it becomes clear, why introducing slip (potentially) improves the numerical simulation of (realistic) viscoelastic fluid flows: First of all, the magnitude of the pressure drop $\frac{\partial p}{\partial x}$ is decreased for increasing κ, which in turn allows higher values of Λ for satisfying the inequalities arising from the non-solvent Wagner or Giesekus model. In particular, this might be relevant in case of considering realistic material and model parameters within the constitutive equations – for example concerning polymer melts, which typically involve large relaxation times, but no solvent contribution to the viscosity. Secondly, higher relaxation times lead to an intensified shear thinning effect or a plateau-like shape of the velocity field even in case of wall slip. At the same time, large gradients of the flow fields are avoided, in contrast to results including no-slip boundary conditions. This might help the numerical scheme when simulating viscoelastic fluids for higher relaxation times or Weissenberg numbers, since for example a critical growth of the variables is suppressed. Hence, applying techniques such as LCR or adaptive grid refinement for resolving large flow gradients might be unnecessary. However, the optimal choice of a suitable slip parameter for a given setting still needs to be exploited, where the above results regarding the Wagner model are computed based on a – in some sense – minimal slip length. But it is not clear, whether that is a meaningful choice for reproducing experimental results, which is why a detailed validation and evaluation of modelling wall slip is required.

As an alternative approach to introducing slip boundary conditions, well-defined constitutive equations may be taken into account when simulating realistic viscoelastic fluids, which do not consist of such underlying mathematical properties as the Giesekus or Wagner model. In detail, a suitable candidate is provided by the linear or exponential PTT model, where the corresponding fully developed velocity profiles are computable for any parameter set. In particular, the corresponding analytical expressions from Eq. (3.6) can be evaluated for any (reasonable) choice of the pressure drop as well as the relaxation time, that is $\frac{\partial p}{\partial x} < 0$ and $\Lambda > 0$. Thus, realistic viscoelastic material parameters might be successfully considered in corresponding fluid flow simulations by taking into account the PTT model instead of the Giesekus or Wagner model. Recall, that also the PSM model seems to consist of structural limits similar to the Wagner model as indicated in Sec. 3.2.1. But, an according analysis can not be performed straightforward, since the arising nonlinear viscosity is not given in closed form. Naturally, a well-defined integral model is of interest as well, but such an “analog” to the PTT model is not known to the author.

In summary, fully developed flow profiles resulting from the nonlinear integral viscoelastic models considered above resemble the behaviour already observed in case of the differential constitutive equations in Sec. 3.1. As an important feature of both types of constitutive laws, a pronounced shear thinning effect is obtained for increasing relaxation times, while varying the remaining material and model parameters shapes the flow profiles in a reasonable way. But, a validation of the determined flow profiles is performed in terms of two-dimensional channel flow configurations only with respect to differential models. Nevertheless, the corresponding framework may be used for further investigations, in particular concerning the initial validation and evaluation of the Tensor Diffusion approach in Sec. 4.1.1 regarding differential as well as integral viscoelastic flow models. Moreover, a selection of the fully developed flow profiles covered above are prescribed in Secs. 4.1.2 and 4.2 as in- and outflow profiles concerning two-dimensional Finite Element simulations in terms of the Tensor Diffusion approach.

Chapter 4

Validating and evaluating the Tensor Diffusion approach

In Chpt. 2, state-of-the-art approaches regarding Finite Element techniques for simulating viscoelastic fluid flows are presented taking into account differential as well as integral material models. In both cases, several difficulties and challenges arise, for example the hyperbolic character of the differential constitutive law or the evolution equation of the Finger tensor requires certain attention. Moreover, the multi-mode approach causes huge numerical effort in terms of differential models concerning the prediction of realistic viscoelastic fluids. At the same time, the High Weissenberg Number Problem may lead to a breakdown of calculations. Furthermore, integral models at realistic and thus large relaxation times involve high computational costs due to evolving the Deformation Fields over a wide time interval – even in case of stationary flows. In particular, major difficulties arise when treating both types of constitutive equations numerically in case of a vanishing solvent viscosity in the momentum equation of the Stokes part of the flow model. In detail, the properties of the mathematical problem are downgraded as the regularity of the Stokes subproblem gets lost, where the stress tensor is considered as a force term. Hence, certain issues occur when considering fully coupled solution schemes in the non-solvent case, for example regarding the applicability of Newton-multigrid techniques, which turn out to behave non-robust. In addition, the possibility of applying decoupled solution approaches especially concerning direct steady-state configurations is removed, since no meaningful relation of the velocity and stress fields is established in the momentum equation. Unfortunately, such a segregated scheme is highly demanded in case of solving integral or multi-mode differential models, where a monolithic treatment is not practical or leads to high numerical effort.

Concerning this matter, the Tensor Diffusion approach is proposed in Sec. 2.4, where the extra-stress tensor is decomposed into a tensor-valued viscosity multiplied with the strain-rate tensor. By inserting this decomposition into the momentum equation of the Stokes subproblem, a diffusive operator is recovered even in the case of no solvent, which conceptually offers the possibility to resolve some of the difficulties addressed above. However, the underlying assumption, that a corresponding decomposition of the extra-stress exists in general, is not yet verified. In this regard, the Tensor Diffusion approach is validated in the following section by reproducing viscoelastic flow characteristics usually resulting from differential or integral material models in terms of one as well as two-dimensional flow states. In detail, channel flow configurations as well as the Flow around cylinder benchmark are considered. Afterwards, further investigations on the Tensor Diffusion approach to simulate viscoelastic fluid flows are carried out in Sec. 4.2 based on the flow through a rounded contraction. In doing so, possible improvements of the original method, but also certain limitations of the novel approach are evaluated, especially concerning numerical solution schemes.

4.1 Proof of concept

In this section, the novel Tensor Diffusion approach is validated in terms of quasi one-dimensional as well as actual two-dimensional flow configurations. The initial validation is performed by means of Poiseuille-like flows in Sec. 4.1.1, where the considered tensor-valued viscosity is derived in terms of fully developed channel flows. By inserting the arising Diffusion Tensor into the pure Tensor Stokes problem, the according fully developed flow profiles, especially regarding the velocity field, are intended to be recovered within two-dimensional Finite Element simulations. In addition, it turns out that the viscoelastic rheology indeed might be completely characterised by a suitable choice of the Diffusion Tensor. Thus, the possibility is highlighted to potentially remove the constitutive equation of differential or integral type from the viscoelastic flow model, which would significantly improve the corresponding numerical approach as outlined in Sec. 2.4. However, the validation of the Tensor Diffusion approach is complemented in Sec. 4.1.2 by considering two-dimensional Finite Element simulations of the well-known Flow around cylinder benchmark [47, 46, 61]. In this regard, the drag values as well as stress profiles resulting from the four-field formulation of the Tensor Stokes problem are compared to reference results to evaluate the corresponding solution quality.

4.1.1 Poiseuille-like flow configurations

In the following, channel flow configurations are considered for validating the Tensor Diffusion approach. By considering the framework of fully developed channel flows from Chpt. 3, the original two-dimensional differential or integral viscoelastic model can be reduced to a more or less simple set of equations. In this context, the components of the Diffusion Tensor are derived either numerically or (semi-)analytically depending on the specific constitutive law. In a second step, the resulting tensor-valued viscosity is applied in the pure Tensor Stokes problem to investigate certain properties of this novel approach by means of two-dimensional Finite Element simulations. For example, the need of considering the symmetrised stress decomposition in the momentum equation or the possibility to reduce the full viscoelastic flow model to a Stokes-like problem is demonstrated.

The UCM model

Similar to determining fully developed flow profiles, it is a reasonable approach starting off the study on the Tensor Diffusion approach by considering the Upper-Convected Maxwell model. In this case, the resulting extra-stress and strain-rate tensors can be given analytically as

$$\mathbf{D}(\mathbf{u}) = \frac{1}{2}\begin{pmatrix} 0 & \frac{\partial u}{\partial y} \\ \frac{\partial u}{\partial y} & 0 \end{pmatrix}, \quad \mathbf{\Sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{pmatrix} = \begin{pmatrix} 2\eta_p \Lambda \left(\frac{\partial u}{\partial y}\right)^2 & \eta_p \frac{\partial u}{\partial y} \\ \eta_p \frac{\partial u}{\partial y} & 0 \end{pmatrix} \tag{4.1}$$

where the extra-stress tensor can be deduced from both, the differential as well as integral version of the UCM model (see Chpt. 3). It is realised from Eq. (4.1), that indeed a matrix- or tensor-valued quantity can be derived – even analytically – relating $\mathbf{\Sigma}$ and $\mathbf{D}$ according to Eq. (2.47), that means $\mathbf{\Sigma} = \mathbf{M} \cdot \mathbf{D}(\mathbf{u})$, where

$$\mathbf{M} = \begin{pmatrix} \mu_{11} & \mu_{12} \\ \mu_{21} & \mu_{22} \end{pmatrix} = 2\eta_p \begin{pmatrix} 1 & 2\Lambda\frac{\partial u}{\partial y} \\ 0 & 1 \end{pmatrix} \tag{4.2}$$

Obviously, the resulting (nonsymmetric) Diffusion Tensor consists of constant diagonal entries, while the upper right entry depends linearly on the relaxation time Λ as well as the derivative of the x-component u of the velocity field. Note, that a diagonal matrix with entries $2\eta_p$ is obtained in the Newtonian limit $\Lambda \to 0$, which correlates to the Newtonian *scalar* viscosity when being inserted into the (symmetrised) Tensor Stokes problem (2.49), hence giving the well-known Stokes equations.

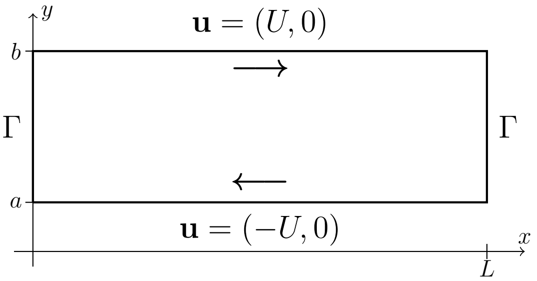


Figure 4.1: Shear flow configuration

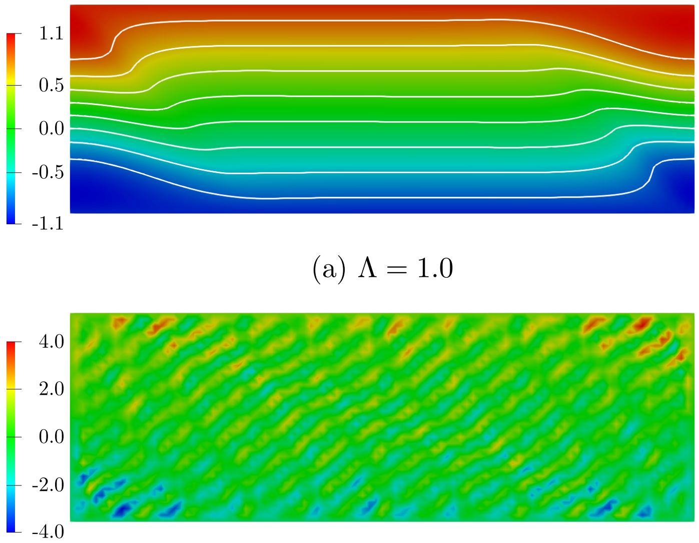
Based on the above Diffusion Tensor, the need of considering the symmetrised Tensor Stokes problem introduced in Sec. 2.4 is demonstrated in terms of (two-dimensional) numerical simulations. Therefore, a so-called shear flow in a channel of height $y \in [-1, 1]$ and length $x \in [0, 6]$ is regarded in the following, where the velocity on the upper and lower walls of the channel points in positive and negative x-direction, respectively (see Fig. 4.1). For this flow configuration, the ODE system from Eq. (3.2) in case of the UCM model, that is $\eta_s = 0$ and $\alpha = 0$, gives a vanishing pressure drop as well as a linear velocity profile, which results in $\frac{\partial u}{\partial y} = U$. Consequently, a globally constant, but nonsymmetric, Diffusion Tensor is obtained from Eq. (4.2), which is applied in the original Tensor Stokes problem

$$-\nabla \cdot (\mathbf{M} \cdot \mathbf{D}(\mathbf{u})) + \nabla p = \mathbf{0}, \qquad \nabla \cdot \mathbf{u} = 0 \tag{4.3}$$

In terms of a numerical treatment, the Finite Element framework proposed in Sec. 2.2.3 is applied, but without considering the extra-stress tensor as a flow variable, since the above Stokes-like problem depends only on the velocity and pressure fields. However, since a quite simple solution will be obtained from the shear flow configuration even by means of the Tensor Stokes problem, the complexity of the flow is increased by incorporating actual two-dimensional effects. Therefore, the velocity profile

$$\tilde{u}\left(y\right) = Uy\left(1 + \gamma\left(1 - y^2\right)\right) \tag{4.4}$$

is set on the left and right edge Γ of the channel, which corresponds to a modification of the actual solution by a polynomial of degree three weighted with the factor $\gamma > 0$. By doing so, a “wrong” boundary condition concerning a fully developed flow is enforced, which still satisfies the continuity equation and thus provides a “consistent” perturbation of the analytical solution. At the same time, the analytical Diffusion Tensor according to the fully developed flow is prescribed globally in the computational domain, which is why the corresponding linear velocity profile should be recovered inside the channel – in case the Tensor Diffusion approach is defined properly.



(a) $\Lambda = 1.0$

(b) $\Lambda = 5.0$

Figure 4.2: x-velocity field resulting from the Tensor Stokes problem (4.3) for the UCM model at $\eta_0 = 1.0$, $U = 1.0$, $\gamma = 1.0$ for several relaxation times

For a moderate relaxation time of $\Lambda = 1.0$, the resulting flow shows a more or less appropriate behaviour. In particular, the desired linear velocity profile is obtained in the middle of the channel, since the contour lines depicted in Fig. 4.2(a) are equidistant. But at the same time, a nonsymmetric velocity distribution results from the – except for the applied Diffusion Tensor – symmetric flow configration. Moreover, the flow breaks down when Λ is increased (see Fig. 4.2(b)) probably again due to the prescribed tensor-valued viscosity from Eq. (4.2) being nonsymmetric. This presumption is confirmed by solving the symmetrised Tensor Stokes problem presented in Eq. (2.49) instead of the original version from Eq. (4.3), which results in the velocity distribution depicted in Fig. 4.3. Towards both, left as well as right edge of computational domain, the velocity profile from Eq. (4.4) is approached in a symmetric way, whereas the fully developed analytical linear profile is recovered in the channel centre. Overall, a symmetric flow field is generated in contrast to the orignal Tensor Stokes problem, together with a smoother transition from left and right edge to the channel centre.

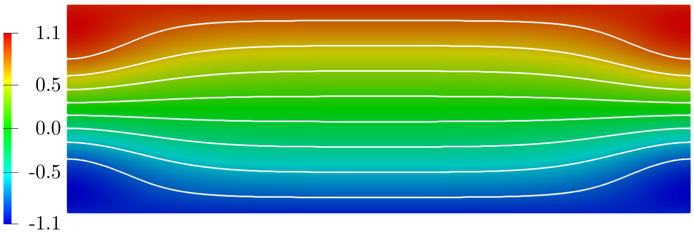


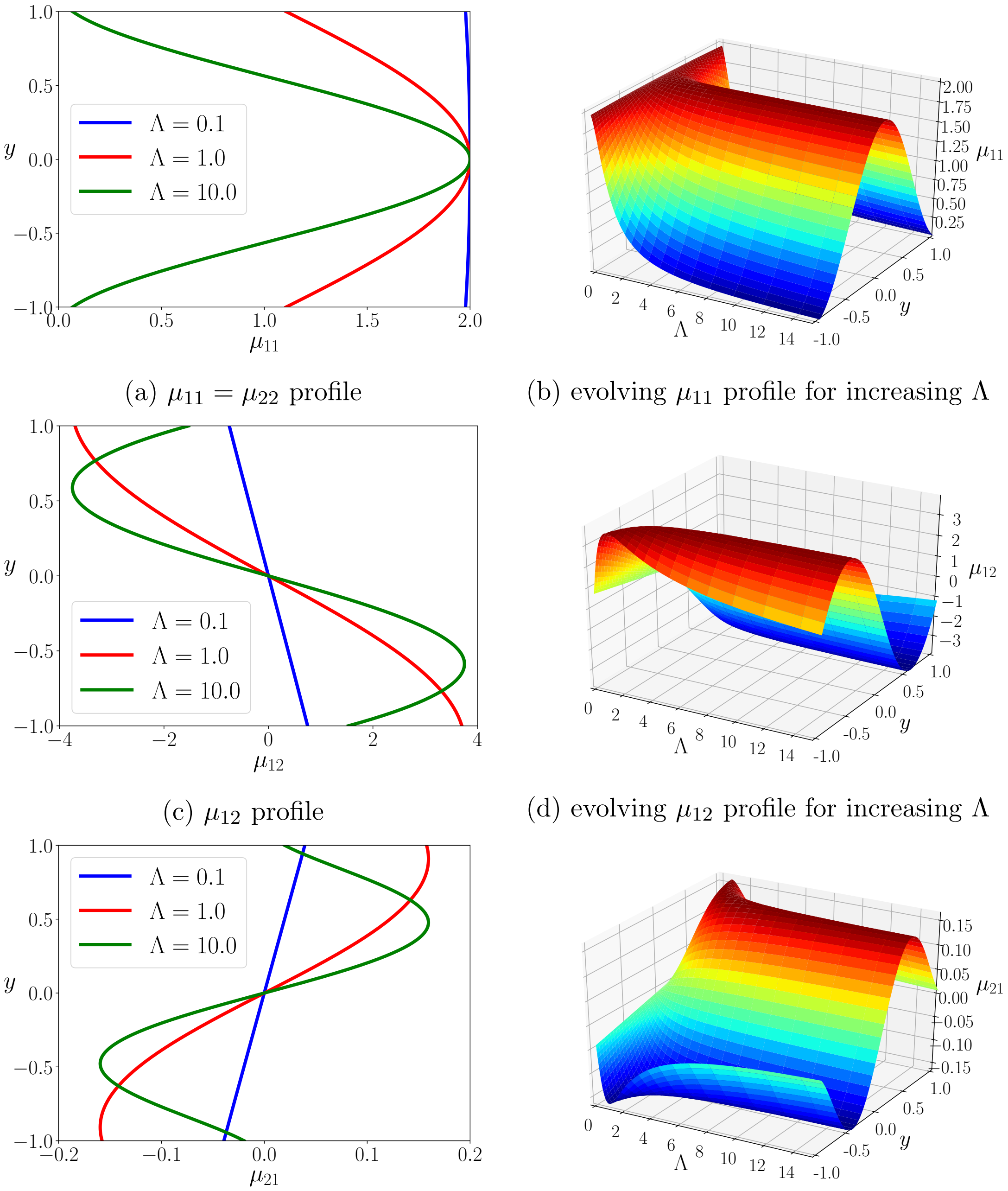
Figure 4.3: x-velocity field resulting from the symmetrised Tensor Stokes problem for the UCM model at $\Lambda = 5.0$, $\eta_0 = 1.0$, $U = 1.0$, $\gamma = 1.0$

Consequently, the Tensor Diffusion approach gives meaningful results for the considered shear flow configuration, at least when applied in terms of the symmetrised Tensor Stokes problem. Thus, the corresponding symmetric operator will be considered in the momentum equation of the flow model in terms of the subsequent investigations as already indicated in Sec. 2.4. But still, the Diffusion Tensor needs to be determined analytically or numerically based on the nonsymmetric stress decomposition, since the symmetrised version leads to a singular system with respect to the components of $\mathbf{M}$. Recall, that the symmetrised stress decomposition is directly obtained from the original nonsymmetric case due to Eq. (2.48), both including the same tensor-valued viscosity.

Poiseuille-like flow for the Giesekus model

As shown above, a tensor-valued viscosity relating the extra-stress tensor $\boldsymbol{\Sigma}$ to the strain-rate tensor $\mathbf{D}(\mathbf{u})$ can be derived analytically from the differential as well as integral version of the UCM model. Furthermore, reasonable results for the shear flow configuration depicted in Fig. 4.1 are obtained by solving an according symmetrised Tensor Stokes problem. In a similar way, the Tensor Diffusion approach is validated below for more complex configurations by considering nonlinear (differential) constitutive equations, represented by the Giesekus model, in terms of Poiseuille-like flows. Since the velocity as well as stress profiles can not be given analytically in this case, the profiles of the Diffusion Tensor $\mathbf{M}$ are determined numerically by solving $\boldsymbol{\Sigma} = \mathbf{M} \cdot \mathbf{D}(\mathbf{u})$ for the components of $\mathbf{M}$ based on the flow profiles determined in Sec. 3.1. Accordingly, a parameter study regarding the effect of Λ, α and β on the components of the Diffusion Tensor is performed to evaluate the shape of $\mathbf{M}$. In a second step, the resulting Diffusion Tensor is applied in the pure Tensor Stokes problem, which is considered within two-dimensional Finite Element simulations of Poiseuille-like flow configurations. In doing so, the shear thinning effect typically predicted by means of the Giesekus model is desired to be reproduced in terms of a simple Stokes-like problem, which includes a suitable tensor-valued viscosity. For increasing Λ, the deviations from the UCM-solutions increase and the profiles show a highly nonlinear behaviour, where μ_{21} seems to behave qualitatively axial symmetric compared to μ_{12}, but consisting of a different magnitude.

First of all, the components of the Diffusion Tensor $\mathbf{M}$ depicted in Fig. 4.4 for varying Λ at fixed $\alpha = 0.1$ and $\beta = 0.0$ are analysed. Naturally, the Diffusion Tensor obtained from the Giesekus model at small relaxation times is close to the solution regarding the UCM model, that means $\mu_{11} = \mu_{22}$ is nearly constant and μ_{12} behaves linearly, while μ_{21} vanishes.



(a) $\mu_{11} = \mu_{22}$ profile

(b) evolving μ_{11} profile for increasing Λ

(c) μ_{12} profile

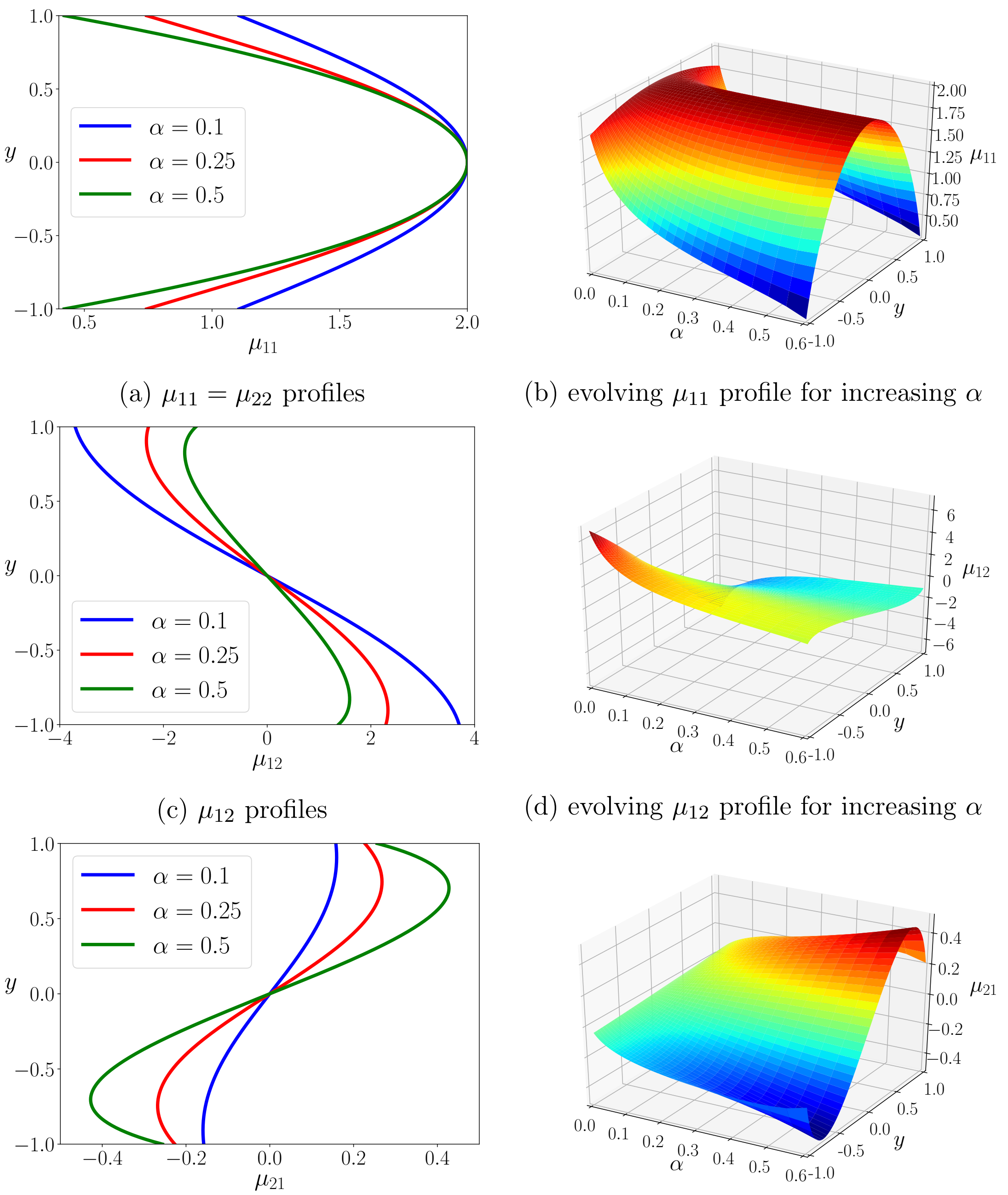
(d) evolving μ_{12} profile for increasing Λ

(e) μ_{21} profile

(f) evolving μ_{21} profile for increasing Λ

Figure 4.4: Diffusion Tensor profiles for the Giesekus model at $\alpha = 0.1$, $\eta_0 = 1.0$, $\beta = 0.0$, $U = 1.0$ for several relaxation times Λ

Similar to the extra-stress tensor, also the Diffusion Tensor seems to be bounded for higher relaxation times in contrast to the UCM model, where the magnitude of μ_{12} is proportional to Λ. In detail, μ_{11} takes the maximum value of $2\eta_p$ on the centre line, while μ_{12} and μ_{21} attain the maximum and minimum values for increasing Λ in between centre line and walls of the channel. Note, that the Diffusion Tensor approaches the Newtonian form at $y = 0$, as μ_{12} and μ_{21} vanish. Moreover, the off-diagonal components of $\mathbf{M}$ show a rapid increase for smaller relaxation times and take extreme values at $y = \pm 1$ for $\Lambda \approx 2$, similar to the diagonal components of the extra-stress tensor (see Fig. 3.5). In contrast, μ_{11} behaves monotonically with respect to Λ at the channel walls.



(a) $\mu_{11} = \mu_{22}$ profiles

(b) evolving μ_{11} profile for increasing α

(c) μ_{12} profiles

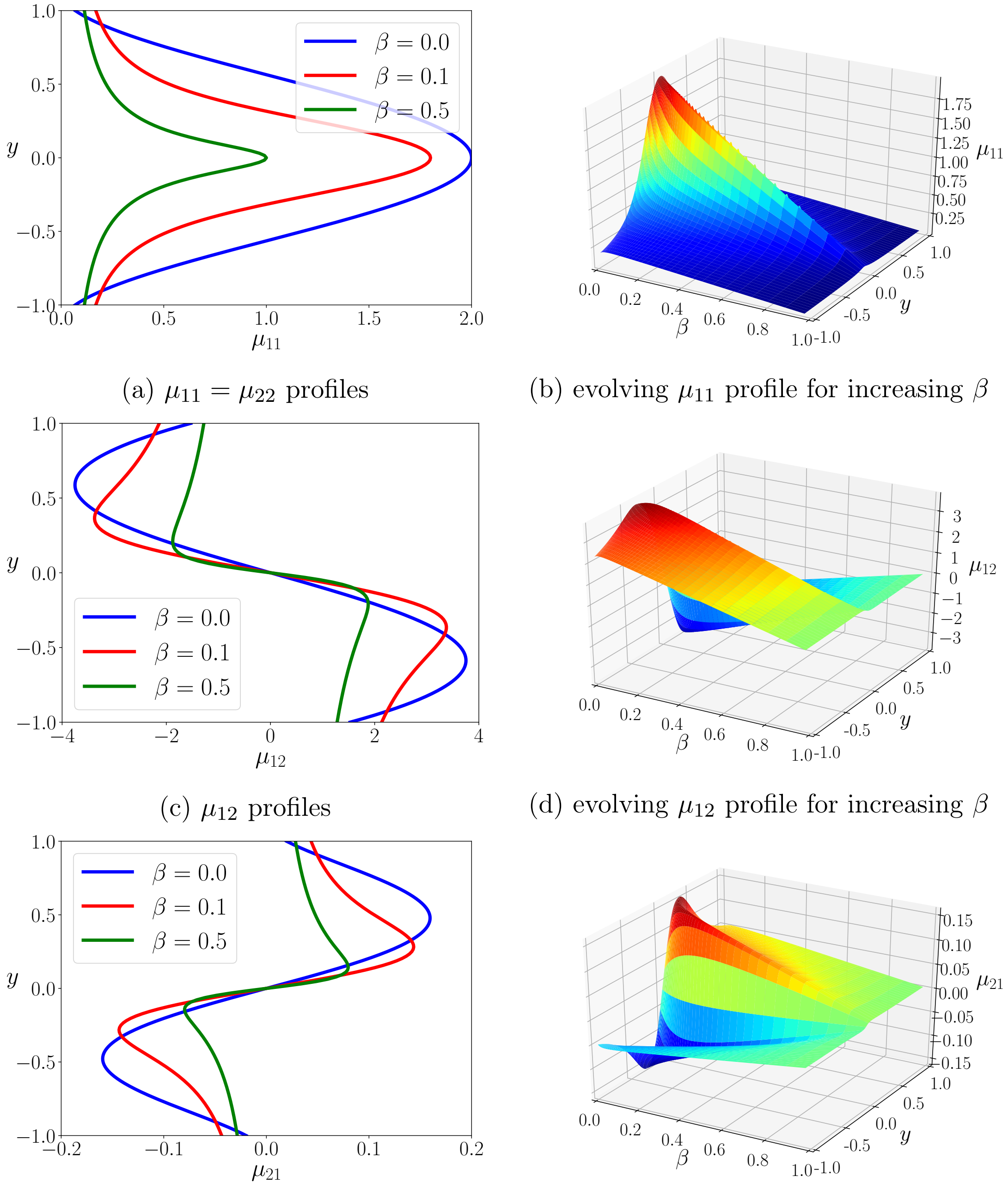
(d) evolving μ_{12} profile for increasing α

(e) μ_{21} profiles

(f) evolving μ_{21} profile for increasing α

Figure 4.5: Diffusion Tensor profiles for the Giesekus model at $\Lambda = 1.0$, $\eta_0 = 1.0$, $\beta = 0.0$, $U = 1.0$ for several mobility factors α

When increasing the mobility factor α at fixed $\Lambda = 1.0$ and $\beta = 0.0$, the components of the Diffusion Tensor $\mathbf{M}$ behave similar to increasing Λ – according to the results observed for the velocity and stress profiles. Thus, μ_{11} again takes the maximum value at $y = 0$ and μ_{12} as well as μ_{21} develop minimum and maximum values for increasing α in between centre line and channel walls (see Fig. 4.5). In some sense, the shape of μ_{11} for higher Λ and α can be interpreted as a shear thinning viscosity, since the magnitude decreases towards the channel walls, that is for higher "shear rates" $\frac{\partial u}{\partial y}$, but a maximum value is obtained on the centre line, that is at zero shear. Again, the components of $\mathbf{M}$ behave mainly monotonically with respect to α, similar to the components of $\mathbf{\Sigma}$. But, the magnitude of the off-diagonal component μ_{21} seems to decrease close to the channel walls for higher α, where σ_{11} increases in similar way. Furthermore, μ_{11} (or μ_{22}) looks similar to σ_{22} and μ_{12} recovers qualitatively the behaviour of σ_{12}.



(a) $\mu_{11} = \mu_{22}$ profiles

(b) evolving μ_{11} profile for increasing β

(c) μ_{12} profiles

(d) evolving μ_{12} profile for increasing β

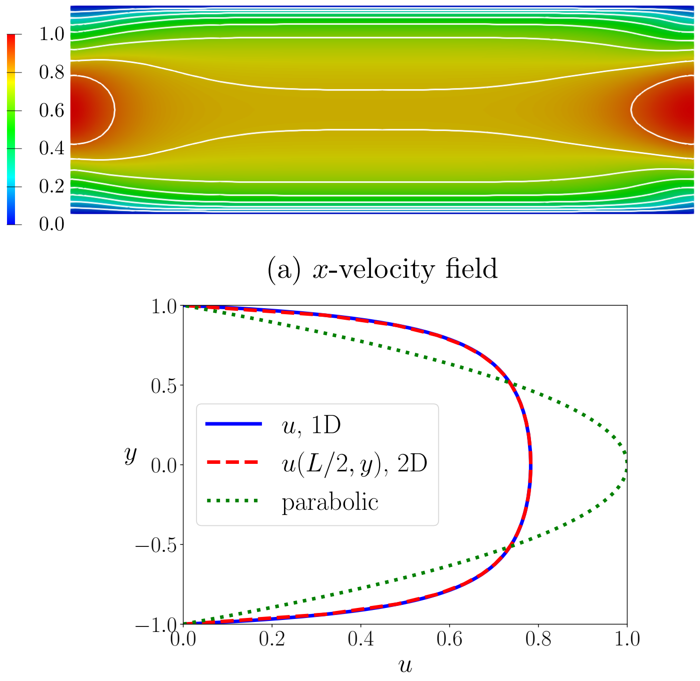
(e) μ_{21} profiles

(f) evolving μ_{21} profile for increasing β

Figure 4.6: Diffusion Tensor profiles for the Giesekus model at $\Lambda = 10.0$, $\alpha = 0.1$, $\eta_0 = 1.0$, $U = 1.0$ for several amounts β of solvent viscosity

In case of a present solvent contribution $\beta = \frac{\eta_s}{\eta_0} > 0$, where the corresponding profiles of $\mathbf{M}$ are depicted in Fig. 4.6, the somewhat strange behaviour of the extra-stress tensor $\mathbf{\Sigma}$ is recovered by the Diffusion Tensor (see Fig. 3.9). In detail, the components of $\mathbf{M}$ seem to show a large growth in the centre of the channel, that is at $y = 0$, while the overall magnitude decreases. At least the latter makes sense, since $\mathbf{M} \to 0$ is expected for $\beta \to 1$ leading to the Newtonian case of a vanishing extra-stress tensor $\mathbf{\Sigma}$, which is realised from Eq. (2.47) or (2.48). However, the reason of an increasing gradient of $\mathbf{M}$ or $\mathbf{\Sigma}$ close to $y = 0$ is still unclear. As before, the maximum value of $\mu_{11} = 2\eta_p$ on the centre line of the channel as well as the axial symmetry of μ_{12} and μ_{21} is observed.

Summarising the observations in terms of varying Λ, α and β, the Diffusion Tensor $\mathbf{M}$ reflects the behaviour of the extra-stress tensor recorded in Sec. 3.1. Thus, the material behaviour indeed might be expressed by means of the Diffusion Tensor instead of the extra-stress tensor or a constitutive equation. This is confirmed in the following by solving the symmetrised Tensor Stokes problem (2.49) for channel flow configurations by means of two-dimensional Finite Element simulations, similar to the investigations in Chpt. 3. In detail, a modified Poiseuille flow is considered in the following, such that a parabolic velocity profile is strongly enforced as in- and outflow data in terms of non-homogeneous Dirichlet boundary conditions [21]. At the same time, the Diffusion Tensor $\mathbf{M}$, which is determined above according to the fully developed nonlinear viscoelastic flow characterised by the Giesekus model, is prescribed globally in the computational domain. Since the nonlinear material behaviour is intended to be described by the Diffusion Tensor instead of the constitutive equation, the corresponding fully developed velocity profile should be recovered in the two-dimensional domain away from the in- and outflow edges. Thereby, the Diffusion Tensor is inserted into the Finite Element framework described in Sec. 2.2.3 by defining element-wise constant representations of $\mathbf{M}$ based on the discrete values resulting from the one-dimensional configuration. Consequently, an approximation of $\mathbf{M}$ according to the discretisation proposed in Sec. 2.4 in terms of the four-field formulation of the Tensor Stokes problem is obtained.



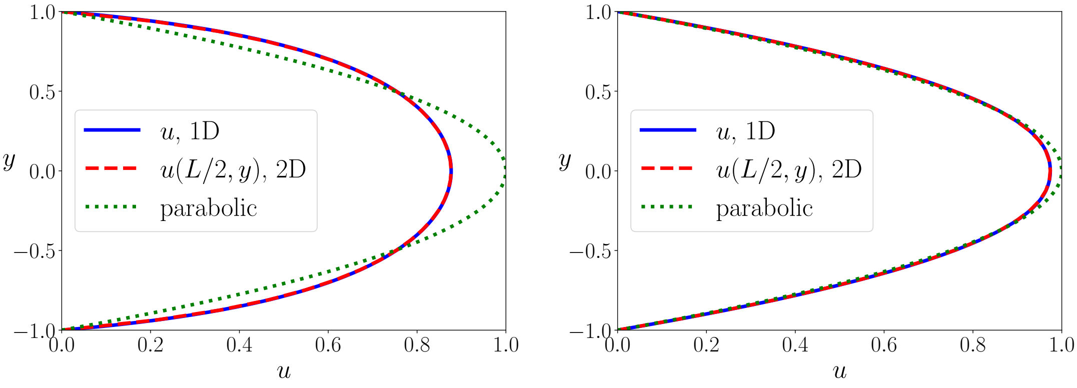
(a) x-velocity field

(b) x-velocity profile

Figure 4.7: Velocity solution resulting from the symmetrised Tensor Stokes problem for the Giesekus model at $\Lambda = 10.0$, $\alpha = 0.1$, $\eta_0 = 1.0$, $\beta = 0.0$, $U = 1.0$

Indeed, the velocity distribution depicted in Fig. 4.7(a) resulting from the Tensor

Stokes problem in case of the Giesekus model shows the expected pronounced shear thinning effect for the quite extreme parameter set consisting of $\Lambda = 10.0$, $\alpha = 0.1$ and $\beta = 0.0$. Furthermore, the velocity profile in the middle of the channel behaves almost perfectly according to the fully developed velocity profile computed from the one-dimensional setting (see Fig. 4.7(b)). In similar way, the Poiseuille flow with parabolic in- and outflow profiles gives meaningful results regarding the Tensor Diffusion approach applied for the parameter sets $(\Lambda = 1.0, \alpha = 0.5, \beta = 0.0)$ as well as $(\Lambda = 10.0, \alpha = 0.1, \beta = 0.5)$ depicted in Fig. 4.8. In detail, the shear thinning effect is reproduced quite well in the solution obtained from the Tensor Stokes problem (2.49) for a higher value of the mobility factor α, where the viscosity depicted in Fig. 4.5 is applied. Furthermore, the nearly parabolic fully developed velocity profile is recovered inside the channel in case of a non-vanishing solvent viscosity as well.



(a) $\Lambda = 1.0$, $\alpha = 0.5$, $\beta = 0.0$ (b) $\Lambda = 10.0$, $\alpha = 0.1$, $\beta = 0.5$

Figure 4.8: x-velocity profile resulting from the symmetrised Tensor Stokes problem for the Giesekus model at $\eta_0 = 1.0$, $U = 1.0$

Thus, the nonlinear solution corresponding to the Giesekus model is reproduced in this two-dimensional channel flow configuration simply by solving the linear Stokes-like problem (2.49) for a suitable choice of $\mathbf{M}$. Consequently, the nonlinear rheology of a viscoelastic fluid – described here by the differential Giesekus model – is reconstructed within a pure $(\mathbf{u}, p)$-problem by defining an appropriate tensor-valued diffusion operator. Hence, a first evidance is provided, that the (symmetrised) Tensor Diffusion approach actually is a reasonable concept.

Poiseuille-like flow for the PSM model

After having validated the novel Tensor Diffusion approach in terms of two-dimensional channel flow configurations regarding differential constitutive equations, nonlinear integral viscoelastic models are considered as a next step. Based on the findings from Sec. 3.2.1, the extra-stress tensor arising from the PSM model in fully developed channel flows reads

$$\boldsymbol{\Sigma} = \frac{\eta_p}{\Lambda^2} \int_0^\infty \exp\left(-\frac{s}{\Lambda}\right) \frac{1}{1 + \gamma s^2 \left(\frac{\partial u}{\partial y}\right)^2} \left[\begin{pmatrix} s^2 \left(\frac{\partial u}{\partial y}\right)^2 & s\frac{\partial u}{\partial y} \\ s\frac{\partial u}{\partial y} & 0 \end{pmatrix} + \mathbf{I} \right] \mathrm{d}s \qquad (4.5)$$

Hence, the above stress integral can be decomposed according to

$$\boldsymbol{\Sigma} = \left\{ 2\frac{\eta_p}{\Lambda^2} \int_0^\infty \exp\left(-\frac{s}{\Lambda}\right) \frac{1}{1+\gamma s^2 \left(\frac{\partial u}{\partial y}\right)^2} \begin{pmatrix} s & s^2 \frac{\partial u}{\partial y} \\ 0 & s \end{pmatrix} \mathrm{d}s \right\} \mathbf{D}\left(\mathbf{u}\right) + \mathbf{N} \tag{4.6}$$

where the strain-rate tensor $\mathbf{D}\left(\mathbf{u}\right)$ takes the form presented in Eq. (4.1). By separating the isotropic contribution supplied by the identity $\mathbf{I}$ in Eq. (4.5), a diagonal matrix $\mathbf{N} \in \mathbb{R}^{2\times 2}$ with entries

$$\nu := \nu_{11} = \nu_{22} = \frac{\eta_p}{\Lambda^2} \int_0^\infty \exp\left(-\frac{s}{\Lambda}\right) \frac{1}{1+\gamma s^2 \left(\frac{\partial u}{\partial y}\right)^2} \mathrm{d}s, \quad \nu_{12} = \nu_{21} = 0 \tag{4.7}$$

is obtained, which consist of the infinite integral over the single-mode memory function multiplied with the damping function from Eq. (3.14). Consequently, a generalised stress decomposition compared to Eq. (2.47) of the form

$$\boldsymbol{\Sigma} = \mathbf{M} \cdot \mathbf{D}\left(\mathbf{u}\right) + \mathbf{N} \tag{4.8}$$

is derived, which gives

$$-\frac{1}{2} \nabla \cdot \left(\mathbf{M} \cdot \mathbf{D}\left(\mathbf{u}\right) + \mathbf{D}\left(\mathbf{u}\right) \cdot \mathbf{M}^\top \right) - \nabla \cdot \mathbf{N} + \nabla p = 0, \quad \nabla \cdot \mathbf{u} = 0 \tag{4.9}$$

after being incorporated into the momentum equation of the symmetrised Tensor Stokes problem (2.49). Since ν from Eq. (4.7) in the current setting is in principle a function of $\frac{\partial u}{\partial y}$ and thus y itself, the additional quantity $\nabla \cdot \mathbf{N}$ in Eq. (4.9) can be regarded as $\nabla \nu$. Consequently, again a (symmetrised) version of the Tensor Stokes problem is obtained similar to Eq. (2.49) by considering the modified pressure $P = p - \nu$ instead of p. Hence, the isotropic part $\mathbf{N}$ of the stress decomposition from Eq. (4.8) does not affect the solution regarding the velocity field, but the pressure field as well as the extra-stress tensor. However, the latter is of minor interest in this context, since the stresses are calculated in pure post-processing fashion only for considering the pure Tensor Stokes problem (4.9). Since the main focus of the current investigations is the evaluation of the Diffusion Tensor $\mathbf{M}$, which is determined based on the velocity field, the modification of the pressure is not taken into account here. Thus, the original symmetrised Tensor Stokes problem from Eq. (2.49) is considered in the following, where $\mathbf{M}$ is calculated from Eq. (4.6). In contrast to the nonlinear differential Giesekus model, the components of the Diffusion Tensor $\mathbf{M}$ (and also the tensor $\mathbf{N}$) resulting from the PSM model can be written explicitly as functions of $\frac{\partial u}{\partial y}$, in detail

$$\mu_{11} = \mu_{22} = 2\frac{\eta_p}{\Lambda^2} \int_0^\infty \exp\left(-\frac{s}{\Lambda}\right) \frac{s}{1+\gamma s^2 \left(\frac{\partial u}{\partial y}\right)^2} ds \tag{4.10a}$$

$$\mu_{12} = 2\frac{\eta_p}{\Lambda^2} \int_0^\infty \exp\left(-\frac{s}{\Lambda}\right) \frac{s^2 \frac{\partial u}{\partial y}}{1+\gamma s^2 \left(\frac{\partial u}{\partial y}\right)^2} ds \tag{4.10b}$$

$$\mu_{21} = 0 \tag{4.10c}$$

Hence, the complex rheology arising from the stress integral over a time interval of infinite length is explicitly expressed by means of the tensor-valued viscosity $\mathbf{M}$. Consequently, the full integral viscoelastic model is transformed in case of fully developed channel flows to a generalised Stokes-like problem of the form

$$-\frac{1}{2}\nabla\cdot\left(\mathbf{M}\left(\frac{\partial u}{\partial y}\right)\cdot\mathbf{D}\left(\mathbf{u}\right)+\mathbf{D}\left(\mathbf{u}\right)\cdot\mathbf{M}^{\top}\left(\frac{\partial u}{\partial y}\right)\right)+\nabla p=0,\quad \nabla\cdot\mathbf{u}=0 \quad (4.11)$$

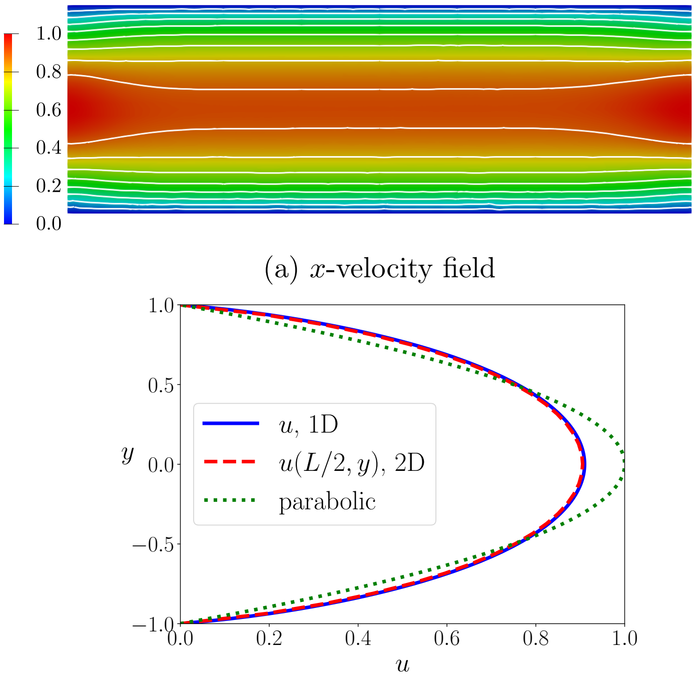
Thereby, simply $(\mathbf{u},p)$ (or the x-component of $\mathbf{u}$ in this case) represent the remaining unknowns for simulating a nonlinear two-dimensional viscoelastic Poiseuille flow originally described by the PSM model consisting of Eq. (2.15a) and (2.43). Unfortunately, integrals over infinite time still have to be computed in Eq. (4.10), since the components of $\mathbf{M}$ are hardly given in closed form. But, the extra-stress or Finger tensors do not need to be considered as numerical unknowns anymore.

Note, that $\mu_{21}=0$ holds in case of the PSM model (and also the Wagner model, see below) in contrast to the Giesekus model due to the separated isotropic contribution denoted by $\mathbf{N}$. This directly correlates to a vanishing component σ_{22} of the extra-stress tensor $\mathbf{\Sigma}$, since $\sigma_{22}=\mu_{21}D_{12}$ is case of fully developed channel flows. Accordingly, a vanishing y-dependent behaviour of the pressure function f_p is obtained, which is introduced in terms of calculating fully developed flow profiles in case of differential as well as integral constitutive equations in Chpt. 3. For recovering a non-vanishing μ_{21} according to the Giesekus model, the generalised stress decomposition from Eq. (4.8) can be written in terms of the original decomposition from Eq. (2.47) by considering $\tilde{\mathbf{M}}=\mathbf{M}+\mathbf{N}\cdot\mathbf{D}^{-1}\left(\mathbf{u}\right)$ assuming that the strain-rate tensor is invertible. In turn, μ_{21} being non-vanishing in case of the Giesekus model suggests, that the corresponding extra-stress tensor actually consists of a generalised decomposition according to Eq. (4.8) as well.

However, there are several ways to calculate the Diffusion Tensor arising from the PSM model when considering Finite Element simulations of two-dimensional Poiseuille-like flows by means of the Tensor Diffusion approach. Similar to the Giesekus model, the components of $\mathbf{M}$ can be calculated from Eq. (4.8) based on the one-dimensional fully developed flow profiles from Sec. 3.2.1, that is in the discrete one-dimensional grid points. Alternatively, the integral expressions from Eq. (4.10) can be evaluated during pre-processing to build up the element-wise constant representation of the Diffusion Tensor. Therefore, the fully developed velocity profile corresponding to the PSM model would need to be provided in terms of suitable Finite Element interpolants. Furthermore, calculating the components of $\mathbf{M}$ from Eq. (4.10) can be considered as an intermediate step when solving the (nonlinear) problem from Eq. (4.11) in a two-dimensional Finite Element framework. In this context, an “infinite” integral needs to be computed in every cubature point within each decoupled step of solving Eq. (4.11), actually leading to a huge numerical effort. Nevertheless, the latter option is chosen to evaluate the Tensor Diffusion approach in case of the PSM model, since relatively “small” computations have to be performed within the Poiseuille-like flow configuration. In addition, a nonlinear problem in $(\mathbf{u},p)$ is obtained resulting in a conceptually different approach than applied above for the Giesekus model. Obviously, this solution approach is already utilised in Sec. 3.2 for calculating fully developed flow profiles in the context of the PSM model. In fact, the quantity μ^{PSM}

(or μ^{Wag}) coincides with the component μ_{11} of the Diffusion Tensor $\mathbf{M} \in \mathbb{R}^{2\times 2}$, since the stress component $\sigma_{12} = \mu_{11} D_{12}$ is considered in the corresponding basic equation (3.10).

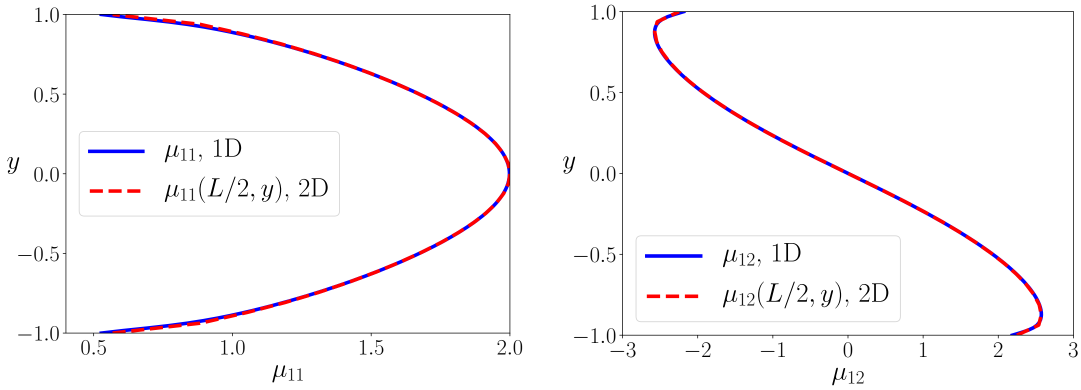
Below, simulation results regarding the PSM model solved by means of the generalised Tensor Stokes problem (4.11) in the context of Poiseuille-like flows are presented, where the corresponding Diffusion Tensor is defined according to Eq. (4.10). Again, a parabolic velocity profile is prescribed on the in- and outflow edges, while the considered Diffusion Tensor corresponds to the fully developed viscoelastic flow. Thus, the according fully developed velocity profile, especially consisting of shear thinning, should be recovered inside the channel.



(a) x-velocity field

(b) x-velocity profile

Figure 4.9: Velocity solution resulting from the symmetrised Tensor Stokes problem for the PSM model at $\Lambda = 1.0$, $\gamma = 0.1$, $\eta_0 = 1.0$, $\beta = 0.0$, $U = 1.0$



(a) $\mu_{11} = \mu_{22}$ profile

(b) μ_{12} profile

Figure 4.10: Diffusion Tensor profiles for the PSM model at $\Lambda = 1.0$, $\gamma = 0.1$, $\eta_0 = 1.0$, $\beta = 0.0$, $U = 1.0$

It turns out, that horizontal contour lines actually occur away from the in- and outflow edges in the velocity field depicted in Fig. 4.9(a), which results from the Tensor Stokes problem (4.11). At the same time, the velocity profile obtained from the two-dimensional simulations matches the fully developed profile computed from the one-dimensional setting almost perfectly (see Fig. 4.9(b)). To further validate the Tensor Diffusion approach in this context, the corresponding profiles of $\mathbf{M}$ obtained from the two-dimensional Tensor Stokes problem are compared with the resulting quantity calculated in the one-dimensional grid points based on the fully developed velocity and stress profiles (see Fig. 4.10). Obviously, these profiles show a very good agreement, similar to the velocity field. Hence, both flow configurations lead to the same solution, which indicates, that the Tensor Diffusion approach is a reasonable approach for treating integral viscoelastic models in terms of numerical simulations – at least concerning channel flow configurations.

It has to be emphasised, that the above flow fields, which are originally described by an integral viscoelastic flow model, are computed by means of a pure Stokes-like problem including a nonlinear tensor-valued viscosity explicitly expressed depending on the velocity field. Thus, the Tensor Diffusion approach offers the possibility to reduce the full viscoelastic flow model to an actual generalised Tensor Stokes problem in the unknowns $(\mathbf{u}, p)$. But neither in case of the Giesekus or PSM model, the corresponding tensor-valued viscosity can be given in closed form: The Diffusion Tensor has to be calculated numerically when considering the Giesekus model, while it can be given analytically in context of the PSM model, but not in closed form. Consequently, there is still the need to determine the Diffusion Tensor during pre-processing or in a decoupled step of the solution scheme. But, at least the original problem formulation is simplified by removing the stress field from the flow model.

Poiseuille-like flow for the Wagner model

As indicated above, there will be a huge improvement in the Tensor Diffusion approach by expressing $\mathbf{M}$ in terms of $\mathbf{u}$ in closed form, since an actual nonlinear tensor-valued viscosity function is derived from the original viscoelastic material law. Fortunately, this is the case for the Wagner model, where the damping function in the stress integral (2.43b) is chosen from Eq. (3.20), which results in

$$
\begin{aligned}
\boldsymbol{\Sigma} \quad = \quad & \left\{ 2\frac{\eta_p}{\Lambda^2} \int_0^\infty \exp\left(-s \left(\frac{1}{\Lambda} + n_1 \sqrt{\left(\frac{\partial u}{\partial y} \right)^2} \right) \right) \begin{pmatrix} s & s^2 \frac{\partial u}{\partial y} \\ 0 & s \end{pmatrix} \mathrm{d}s \right\} \mathbf{D}\left(\mathbf{u}\right) \\
& \cdots + \frac{\eta_p}{\Lambda^2} \int_0^\infty \exp\left(-s \left(\frac{1}{\Lambda} + n_1 \sqrt{\left(\frac{\partial u}{\partial y} \right)^2} \right) \right) \mathbf{I} \, \mathrm{d}s \qquad (4.12)
\end{aligned}
$$

again for $f = 1$ as in Sec. 3.2.2. Consequently, a generalised stress decomposition according to Eq. (4.8) may be derived, where the components of the Diffusion Tensor $\mathbf{M}$ as well as the isotropic contribution $\mathbf{N}$ can be given in closed form reading

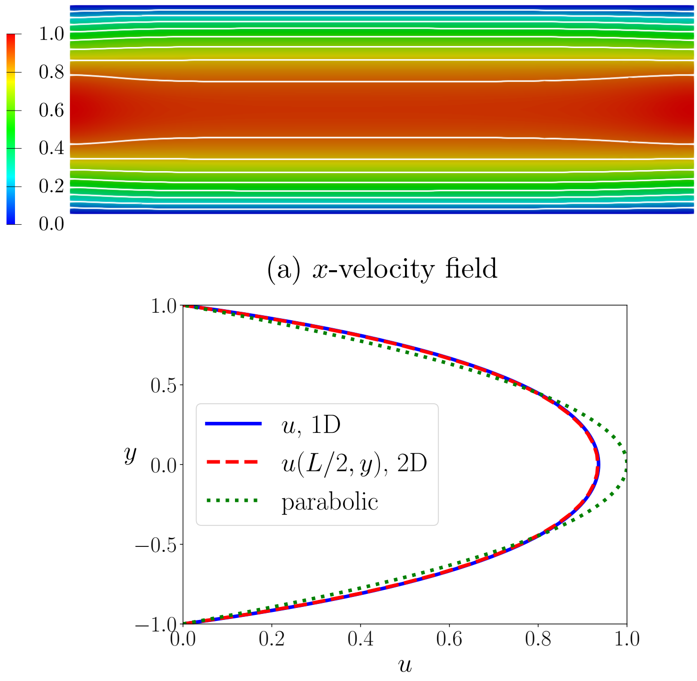
$$\mu_{11} = 2\eta_p \left(1 + n_1 \Lambda \sqrt{\left(\frac{\partial u}{\partial y}\right)^2}\right)^{-2} \tag{4.13a}$$

$$\mu_{12} = 4\eta_p \Lambda \frac{\partial u}{\partial y} \left(1 + n_1 \Lambda \sqrt{\left(\frac{\partial u}{\partial y}\right)^2}\right)^{-3} \tag{4.13b}$$

$$\nu = \frac{\eta_p}{\Lambda} \left(1 + n_1 \Lambda \sqrt{\left(\frac{\partial u}{\partial y}\right)^2}\right)^{-1} \tag{4.13c}$$

besides $\mu_{22} = \mu_{11}$ and $\mu_{21} = 0$. Hence, the original integro-differential flow model from Eq. (2.43) including the damping function proposed in Refs. [17, 18] can be transformed to a generalised Tensor Stokes problem in the unknowns $(\mathbf{u}, p)$ including a shear-rate dependent *tensor-valued* viscosity. In particular, this problem formulation can be interpreted as an extension of classical generalised Stokes equations involving a shear-rate dependent *scalar-valued* viscosity, which are considered for example in Ref. [3]. But, the Tensor Stokes problem (4.11) could be able to predict actual viscoelastic material behaviour, while generalised Stokes equations typically describe a shear thinning (or thickening) effect, only. However, the quantities $\mathbf{M}$ and $\mathbf{N}$ for $f \in \,]0,1[$ arise by a convex combination of terms from Eq. (4.13) according to the damping function from Eq. (3.20). Moreover, the terms in front of the brackets in Eq. (4.13) coincide with the entries of the Diffusion Tensor $\mathbf{M}$ obtained from the UCM model (see Eq. (4.2)). Similar holds for the isotropic contribution denoted by ν in Eq. (4.13c), which arises in case of the integral version of the UCM model by replacing $\mathbf{B} - \mathbf{I}$ with $\mathbf{B}$. Thus, the Diffusion Tensor regarding nonlinear (integral) models possibly results from the UCM model by a suitable scaling, for example with a fractional function as above in case of the Wagner model, which might help to derive a closed form of the components of $\mathbf{M}$ in the case of the PSM model.

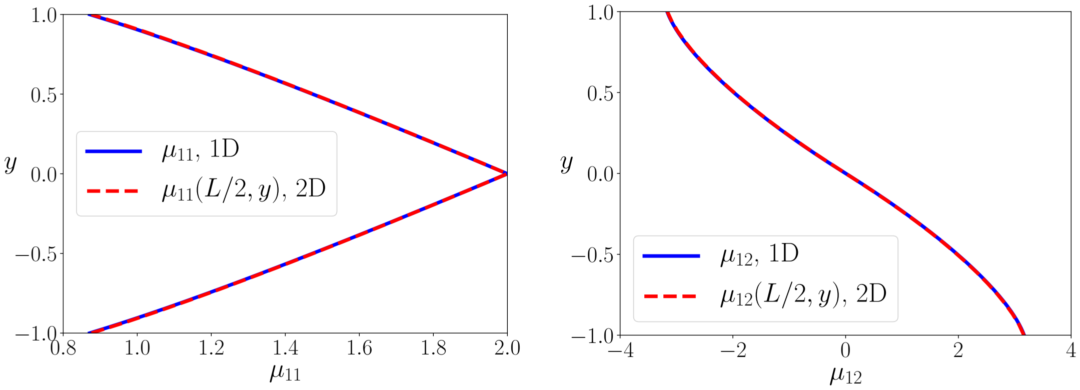
In the following, the generalised Tensor Stokes problem (4.11) is solved for the analytical form of the Diffusion Tensor $\mathbf{M}$ resulting from the Wagner model given in Eq. (4.13). Hence, a nonlinear problem in $(\mathbf{u}, p)$ is solved for the modified Poiseuille flow, where the velocity on the in- and outflow edges is again set to take a parabolic shape. As before, the flow should evolve to its fully developed shape away from the in- and outflow boundaries, since the Diffusion Tensor corresponding to a fully developed channel flow is prescribed globally in the computational domain. In principle, the flow obtained from the Wagner model for the material parameters $f = 0.57$, $n_1 = 0.31$ and $n_2 = 0.106$ given in Ref. [17] shows a similar behaviour as in case of the PSM model depicted in Fig. 4.9. In detail, again a shear thinning behaviour inside of the channel is observed and at the same time, the fully developed velocity profile is recovered (see Fig. 4.11). Thus, viscoelastic channel flow characteristics can be predicted by means of an actual generalised (Tensor) Stokes problem including a nonlinear tensor-valued viscosity function. Recall, that the same flow is originally described by a nonlinear integro-differential model including the damping function from Eq. (3.20). However, the component μ_{11} depicted in Fig. 4.12(a) of the Diffusion Tensor $\mathbf{M}$ is not differentiable in the centre line of the channel, which is observed accordingly for the pressure function f_p (see for example Fig. 3.15(b)). Obviously, this shape occurs due the absolute value of the derivative of the velocity field being present in the denominator of the corresponding analytical expressions of the Diffusion Tensor in Eq. (4.13). In contrast, the components of the Diffusion Tensor resulting from the Giesekus as well as PSM model show a much smoother behaviour as depicted in Fig. 4.4 and 4.10, respectively. But, μ_{12} is smoothed due to $\frac{\partial u}{\partial y}$ being present also in the enumerator of the expression from Eq. (4.13b), while ν should show a similar kink as μ_{11}.



(a) x-velocity field

(b) x-velocity profile

Figure 4.11: Velocity solution resulting from the symmetrised Tensor Stokes problem for the Wagner model at $\Lambda = 1.0$, $f = 0.57$, $n_1 = 0.31$, $n_2 = 0.106$, $\eta_0 = 1.0$, $\beta = 0.0$, $U = 1.0$

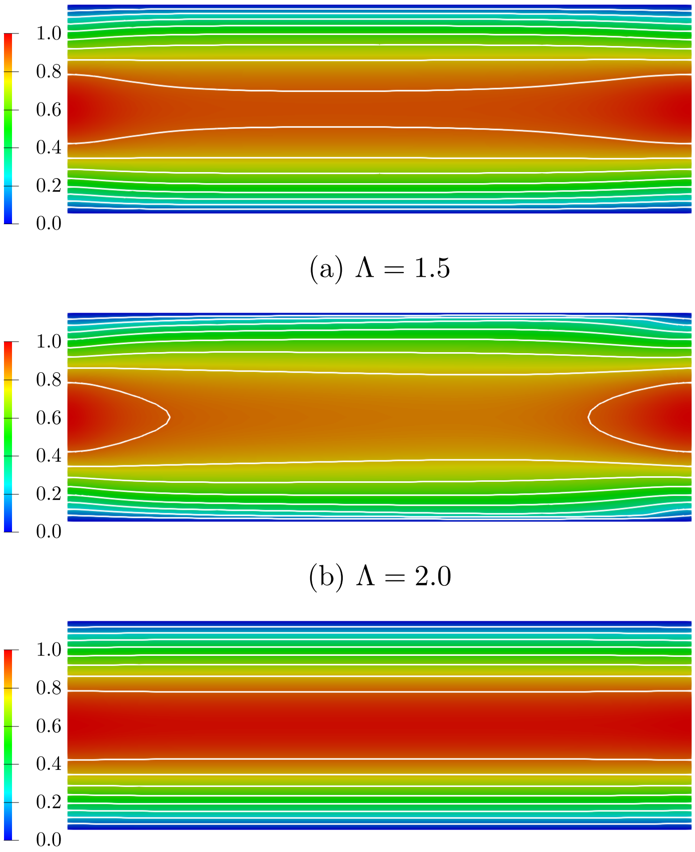


(a) $\mu_{11} = \mu_{22}$ profile

(b) μ_{12} profile

Figure 4.12: Diffusion Tensor profiles for the Wagner model at $\Lambda = 1.0$, $f = 0.57$, $n_1 = 0.31$, $n_2 = 0.106$, $\eta_0 = 1.0$, $\beta = 0.0$, $U = 1.0$

After having validated the Tensor Diffusion approach with respect to the Wagner model in terms of channel flow configurations, an issue investigated in Sec. 3.2.2 is taken up in the following. In detail, no flow profiles at relaxation times significantly larger than $\Lambda = 1$ can be computed for the Wagner model – at least in the framework of fully developed channel flows, which is traced back to the underlying mathematical properties of the rheological model. Below it is illustrated, that calculations fail at a similar critical relaxation time in terms of two-dimensional Finite Element simulations of Poiseuille-like flow configurations. Hence, it is confirmed, that indeed no numerical artefacts give rise to the observed difficulties, but the underlying properties of the model, since a different discretisation as well as solution approach leads to more or less the same limitations.



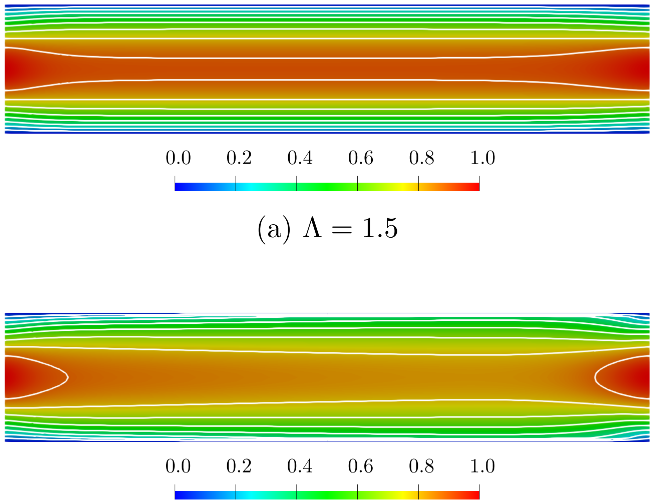
(a) $\Lambda = 1.5$

(b) $\Lambda = 2.0$

(c) $\Lambda = 2.25$

Figure 4.13: x-velocity field resulting from the symmetrised Tensor Stokes problem in a channel of length $L = 6$ for the Wagner model at $f = 0.57$, $n_1 = 0.31$, $n_2 = 0.106$, $\eta_0 = 1.0$, $\beta = 0.0$, $U = 1.0$ for several relaxation times Λ

Recall, that a maximum relaxation time of $\Lambda_{\text{crit}} \approx 1.6$ is reached in the one-dimensional framework regarding the Wagner model at $f = 0.57$, $n_1 = 0.31$ and $n_2 = 0.106$. When simulating the above two-dimensional Poiseuille-like flow by means of the Tensor Diffusion approach for the same parameter set, numerical solutions regarding $\Lambda \approx 2$ actually can be obtained, but the corresponding results show a wrong behaviour as realised from Fig. 4.13. In fact, the flow hardly evolves to its fully developed shape for $\Lambda = 2.0$ and at the same time, the expected shear thinning behaviour, visible for smaller relaxation times, is much weaker in case of $\Lambda > 2$. Thus, the corresponding solutions can be regarded as non-physical, since the shear thinning effect should increase for increasing Λ. An obvious assumption would be, that this wrong flow results due to the relatively short computational domain considered so far. Recall, that the fluid consists of a larger "memory" for higher relaxation times, which might be exemplarily realised from the stress integral in Eq. (2.43b). In detail, the (single- or multi-mode) memory function from Eq. (2.10) decays less rapidly for larger Λ, which causes deformations applied in the past to affect the fluid behaviour over a longer time frame. Consequently, the deformation imposed by the parabolic inflow profile might need a larger channel distance to relax in the context of this Poiseuille-like flow. More precisely, the residence time required for full relaxation of the flow might not be provided by the channel length considered so far, which is why it would make sense to simulate the flow in a longer channel. This is done by extending the channel length from $x \in [0, 6]$ to $x \in [0, 10]$ by adding two square elements to the coarsest mesh level. But even in this case, it is difficult to obtain numerical solutions for significantly larger relaxation times as considered above. Simulation results for the longer channel are depicted in Fig. 4.14, where again the fully developed velocity profile is recovered for $\Lambda = 1.5$ according to the short channel. But even in the longer channel, the velocity field for $\Lambda = 2.0$ does not take a fully developed shape, which seemingly indicates that the channel is still not long enough for reasonably treating the given flow configuration. However, a further extension of the channel length to $x \in [0, 20]$ still does not allow for obtaining a corresponding numerical solution.



(a) $\Lambda = 1.5$

(b) $\Lambda = 2.0$

Figure 4.14: x-velocity field resulting from the symmetrised Tensor Stokes problem in a channel of length $L = 10$ for the Wagner model at $f = 0.57$, $n_1 = 0.31$, $n_2 = 0.106$, $\eta_0 = 1.0$, $\beta = 0.0$, $U = 1.0$ for several relaxation times Λ

In Fig. 4.15, the velocity profile resulting from the Wagner model for $\Lambda = 2.0$ is depicted concerning a varying channel length l, where the profile is taken at the x-position consisting of the "most fully developed" state, that is nearly horizontal contour lines. It turns out, that a longer channel indeed "improves" the obtained shear thinning effect, since an increasing plateau-like behaviour is observed close to the centre line of the channel at $y = 0$. But at the same time, the solutions show a jump in the last element at the channels walls to satisfy the prescribed no-slip boundary condition. Thus, the resolution seems to be not fine enough to capture the change in the velocity field. As the jump in the velocity profile even increases with the channel length, further extending the channel does not seem to resolve the numerical difficulties.

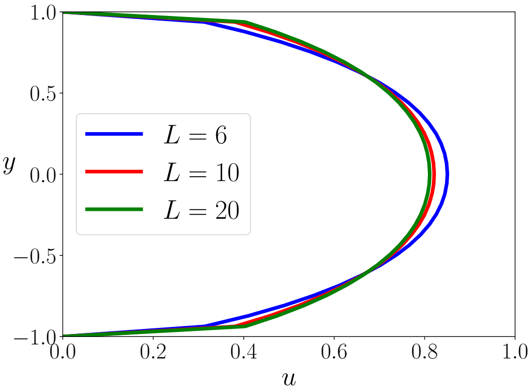
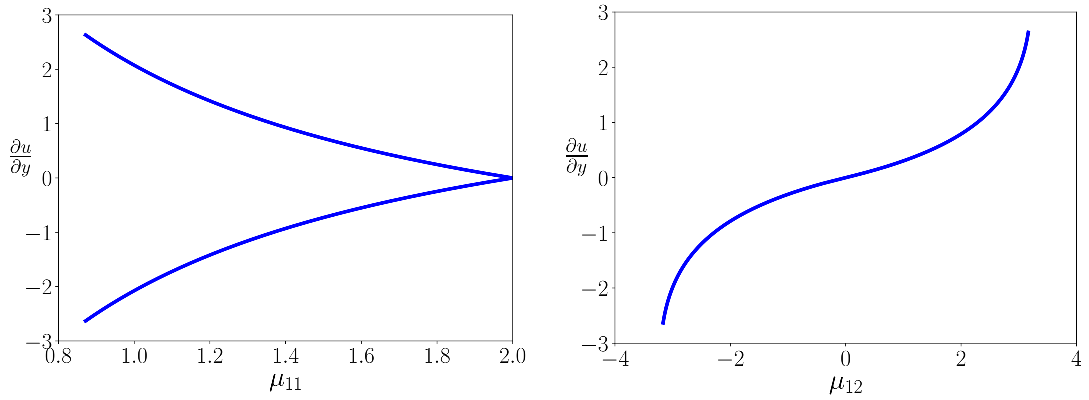


Figure 4.15: x-velocity profiles resulting from the symmetrised Tensor Stokes problem for Wagner model at $\Lambda = 2.0$, $f = 0.57$, $n_1 = 0.31$, $n_2 = 0.106$, $\eta_0 = 1.0$, $\beta = 0.0$, $U = 1.0$ on level 5 for a varying channel length L

Instead, it might make sense to apply local mesh refinement close to the upper and lower walls of the channel to accurately capture the growth of velocity field and remove the jump observed in Fig. 4.15. In addition, large velocity gradients at the channel walls are expected for higher relaxation times due to an intensified shear thinning effect, which requires a sufficiently high spatial resolution as well. At the same time, regarding $\mathbf{M}$ from Eq. (4.13) as a function of $\frac{\partial u}{\partial y}$ gives the impression, that a fine mesh is demanded close to the centre line of the channel, since the Diffusion Tensor consists of a large gradient at $\frac{\partial u}{\partial y} \approx 0$ as depicted in Fig. 4.16.

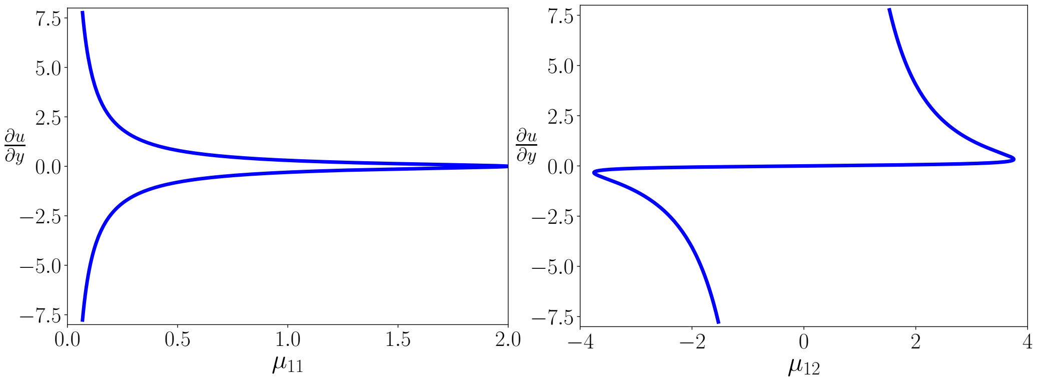


(a) $\mu_{11} = \mu_{22}$ profile

(b) μ_{12} profile

Figure 4.16: Diffusion Tensor profiles for the Wagner model at $\Lambda = 1.0$, $f = 0.57$, $n_1 = 0.31$, $n_2 = 0.106$, $\eta_0 = 1.0$, $\beta = 0.0$, $U = 1.0$

But, applying a globally finer resolution in the two-dimensional mesh actually decreases the critical relaxation time similar to the one-dimensional framework, that is even smaller Λ are computable. In detail, not even the solution corresponding to $\Lambda = 1.7$ can be computed by taking into account one additional mesh refinement. Moreover, the Diffusion Tensor consists of a much larger gradient in case of the Giesekus model for the parameter set $\Lambda = 10.0$, $\alpha = 0.1$ and $\beta = 0.0$ (see Fig. 4.17), which leads to a pronounced shear thinning effect as depicted in Fig. 4.7. But nevertheless, such an extreme configuration is computable by means of the Tensor Diffusion approach – in contrast to the Wagner model for the much "easier" parameter set discussed above, which especially provides a less pronounced shear thinning effect.



(a) $\mu_{11} = \mu_{22}$ profile

(b) μ_{12} profile

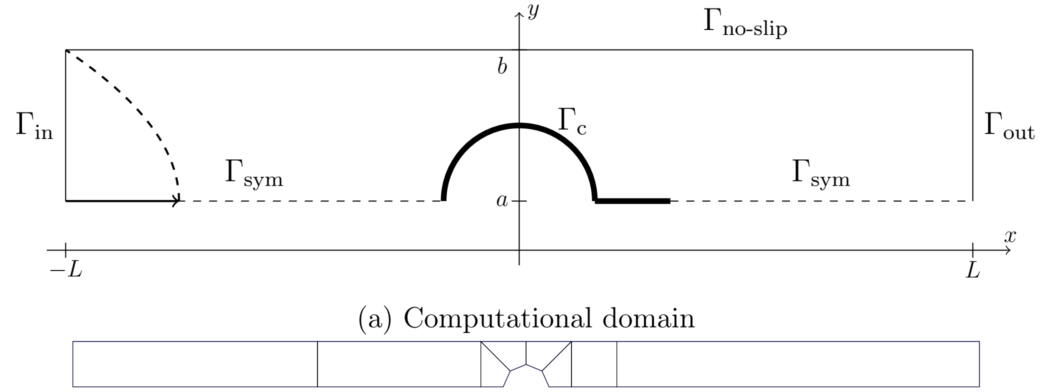
Figure 4.17: Diffusion Tensor profiles for the Giesekus model at $\Lambda = 10.0$, $\alpha = 0.1$, $\eta_0 = 1.0$, $\beta = 0.0$, $U = 1.0$

Concluding, the above remarks encourage, that no numerical artefacts cause the failure in computing numerical solutions regarding higher relaxation times. In fact, difficulties similar to the one-dimensional Finite Difference framework are obtained with respect to the generalised Tensor Stokes problem (4.11) treated in a Finite Element context. Instead, probably the structural limits of the Wagner model, which are already derived in Sec. 3.2.2, prevent the calculations of reasonable numerical solutions regarding higher relaxation times. Moreover, the velocity profiles for the Wagner model at $\Lambda = 2.0$ depicted in Fig. 4.15 give the impression, that indeed a slip instead of a no-slip boundary condition might be a reasonable choice regarding the velocity field in this setting. Recall, that this is already proposed in Sec. 3.2.2 as a remedy for computing "arbitrary" large relaxation times by means of the Giesekus or Wagner model.

In summary, phenomena arising in viscoelastic channel flow configurations can be predicted by simply solving a generalised Tensor Stokes problem in the unknowns $(\mathbf{u}, p)$. In terms of this Stokes-like problem, the complex rheology arising from the viscoelastic material behaviour is completely hidden inside the Diffusion Tensor. In addition, it is even possible to explicitly model the nonlinear tensor-valued viscosity function depending on the "shear rate" $\frac{\partial u}{\partial y}$ in case of the Wagner model. Recall, that such flows are originally characterised by nonlinear (integro-)differential flow models models including the velocity, stress and pressure fields as (numerical) variables. Thus, the need of considering a separate stress tensor or a constitutive equation for computing the nonlinear viscoelastic solution might be avoided, which dramatically reduces the implied numerical effort. Moreover, the numerical framework for simulating viscoelastic fluid flows can be improved significantly: In this context, discretisation and solution techniques may be applied, which are designed especially for solving the incompressible (Navier-)Stokes equations. Hence, numerical schemes of a further developed state than corresponding techniques regarding viscoelastic fluid flows could be taken into account, which probably lead to more robust and efficient approaches compared to the state of the art.

4.1.2 The Flow around cylinder benchmark

So far, the proposed Tensor Diffusion approach is analysed only in the context of fully developed channel flows due to the corresponding simple flow properties, for which it is possible to verify the validity of this novel approach. As a main result, a decomposition of the extra-stress tensor according to $\mathbf{\Sigma} = \mathbf{M} \cdot \mathbf{D}(\mathbf{u})$ can be derived, such that the Diffusion Tensor $\mathbf{M}$ is given explicitly depending on the (gradient of the) velocity field $\mathbf{u}$ – especially regarding integral viscoelastic models. Furthermore, viscoelastic material behaviour can be predicted by means of the Tensor Diffusion approach in quasi one-dimensional channel flows. In the following, the Tensor Diffusion approach is validated in terms of more general two-dimensional flow configurations, in detail the well-known Flow around cylinder benchmark [46, 47, 61]. The underlying flow configuration is depicted in Fig. 4.18(a), where the considered computational domain is of length $x \in [-L, L]$ for $L = 20$ and height $y \in [a, b] = [0, 2]$ with a confined cylinder of radius $R = 1$ placed in the origin.



(a) Computational domain

(b) Coarse mesh

Figure 4.18: Configuration for the Flow around cylinder benchmark

But for such actual two-dimensional flow states, an explicit derivation of the corresponding tensor-valued viscosity is not (yet?) possible. Thus, this quantity needs to be determined numerically, which is implemented straightforward into the existing numerical method for simulating viscoelastic fluid flows from Refs. [47, 54] by means of the algebraic approach as described in Sec. 2.4. Hence, the four-field formulations from Eqs. (2.50) and (2.51), that is the original problem formulation as well as the Tensor Stokes problem, are taken into account below to highlight the foundational applicability of the Tensor Diffusion approach even for more complex flow configurations than channel flows. The two problem formulations are discretised by means of the different Finite Element techniques described in Secs. 2.2.3 and 2.4. Details on the underlying coarse mesh (see Fig. 4.18(b)) are given in Tab. 4.1, where mesh refinement is realised in each element by connecting the midpoints of opposite edges. Additionally, the mesh is adapted to the cylinder surface Γ_c for increasing refinement. Moreover, the numbers of degrees of freedom refer to the full set of unknowns, that is the velocity and pressure fields as well as the (symmetric) extra-stress tensor and (nonsymmetric) Diffusion Tensor.

level l	elements	nodes	edges	degrees of freedom
0	8	18	25	311
1	32	51	82	1049
2	128	165	292	3821
3	512	585	1096	14549
4	2048	2193	4240	56741
5	8192	8481	16672	224069

Table 4.1: Mesh information for the Flow around cylinder benchmark

Regarding boundary conditions, a fully developed velocity profile giving a mean velocity of $U_\mathrm{mean} = 1.0$ is prescribed on Γ_in and Γ_out, where the corresponding parabolic profile applied for the Oldroyd-B or UCM model reads

$$\mathbf{u} = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u\,(y) \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{3}{2}\left(1 - \frac{y^2}{4}\right) \\ 0 \end{pmatrix}$$

In addition, $\Gamma_\text{no-slip}$ and Γ_c are set to be no-slip boundaries with respect to the velocity field, where $\mathbf{u} \equiv 0$, and a do-nothing boundary condition [21] is applied on Γ_sym due to the assumed symmetry of the computational domain. Accordingly, the resulting stress profiles are prescribed on Γ_in, while the extra-stress tensor is treated in terms of the do-nothing boundary condition on all remaining boundary segments. No boundary condition needs to be assigned to the pressure field as well as the Diffusion Tensor, since the corresponding degrees of freedom are located in the interiour of each element (see Secs. 2.2 and 2.4).

However, the resulting discrete nonlinear systems are solved via Newton's method, where the initial guess for computing the solution for a given Λ on a fixed mesh level is set to the solution on the same level for a lower Λ. For increasing mesh refinement at a fixed Λ, the solution prolongated from the coarser level is chosen as initial value. The arising linear systems are solved by means of a direct solver, although multigrid techniques may be applied for the Oldroyd-B model at least in terms of the original problem formulation. But, a direct solver is used throughout the following study to be able to evaluate the particular behaviour of the nonlinear solver also regarding the non-solvent case, where multigrid techniques can hardly be applied as illustrated in Secs. 2.2.2 and 2.4. Furthermore, the scope of the present analysis is to evaluate the Tensor Diffusion approach mainly with respect to the solution quality and not the behaviour of the (linear) solver. Note, that applying multigrid techniques in terms of the Tensor Diffusion approach is investigated in Sec. 4.2.2.

For evaluating the accuracy of the simulation results, the drag coefficient $C_D(\mathbf{\Pi})$ resulting on the cylinder surface at several relaxation times is considered. In detail, this quantity is computed based on the total stress tensor $\mathbf{\Pi} \in \mathbb{R}^{2\times 2}$ from Eq. (2.2) according to

$$C_D(\mathbf{\Pi}) = \frac{2}{U_{\text{mean}}^2 R} F_D(\mathbf{\Pi}), \quad F_D(\mathbf{\Pi}) = \int_{\mathcal{T}_{\text{c}}} (\pi_{11} n_1 + \pi_{12} n_2) \frac{\partial v}{\partial x} dx \tag{4.14}$$

Here, $\mathcal{T}_{\text{c}}$ denotes the set of elements next to the cylinder surface Γ_{c} and n_1 as well as n_2 refer to the components of the corresponding local normal vector $\mathbf{n}$. Furthermore, v represents a test function with support on $\mathcal{T}_{\text{c}}$ from the discrete function space regarding the approximation of the velocity field, which is unity in the degrees of freedom attached to the cylinder surface. Recall, that actually two different discrete problem formulations are taken into account due to the varying discretisation techniques concerning the original and Tensor Diffusion approach. Thus, the total stress tensor for computing the drag coefficient from Eq. (4.14) based on the numerical solution is problem dependent. In the following, $\mathbf{\Pi}_\sigma$ denotes the (discretised) total stress tensor arising from the original viscoelastic flow model (2.50), and $\mathbf{\Pi}_\mu$ the one corresponding to the (symmetrised) Tensor Stokes problem (2.51). On the continuous level, these tensors read

$$\begin{aligned} \mathbf{\Pi}_\sigma &= -p\mathbf{I} + 2\eta_s \mathbf{D}(\mathbf{u}) + \mathbf{\Sigma}, \\ \mathbf{\Pi}_\mu &= -p\mathbf{I} + 2\eta_s \mathbf{D}(\mathbf{u}) + \frac{1}{2}\left(\mathbf{M} \cdot \mathbf{D}(\mathbf{u}) + \mathbf{D}(\mathbf{u}) \cdot \mathbf{M}^\top\right) \end{aligned}$$

where $\mathbf{\Sigma}$ is replaced by the symmetrised stress decomposition from Eq. (2.48) to obtain $\mathbf{\Pi}_\mu$ from $\mathbf{\Pi}_\sigma$. In the below study, the drag coefficients calculated via the discrete version of $\mathbf{\Pi}_\mu$ for a sequence of Weissenberg numbers $\text{We} = \Lambda \frac{U_{\text{mean}}}{R} = \Lambda$ or relaxation times Λ are compared to reference results as well as results based on $\mathbf{\Pi}_\sigma$, that is in principle resulting from the validated original approach presented in Ref. [47, 51]. Based on this, the applicability of the Tensor Diffusion approach is validated also with respect to actual two-dimensional flow states, which is done up to now only concerning quasi one-dimensional configurations.

Benchmark configuration

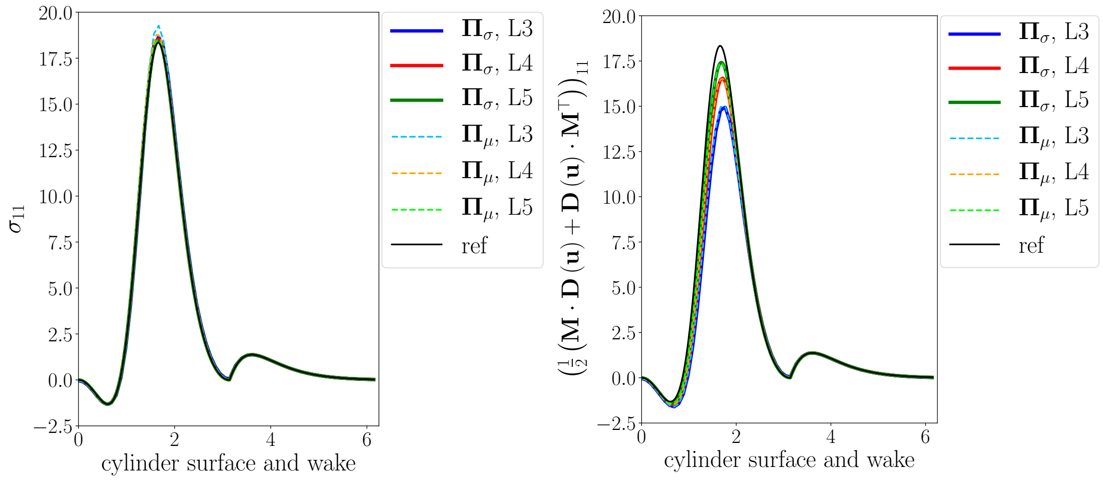
In a first step, the typical benchmark configuration regarding the Oldroyd-B model is considered, that is Eqs. (2.50) and (2.51) are combined with Eq. (2.4) for a total viscosity of $\eta_0 = \eta_s + \eta_p = 1.0$ and an amount $\beta = \frac{\eta_s}{\eta_0} = 0.59$ of solvent contribution. A summary of the resulting drag coefficients is presented in Tab. 4.2 depending on the mesh level l and the Weissenberg number We. In addition, also the number of nonlinear iterations, N_σ or N_μ, of the Newton scheme is given as well as the amount of EOFEM stabilisation regarding the velocity, stress and Diffusion Tensor variable. It turns out, that solutions of the unstabilised Tensor Stokes problem already at a Weissenberg number of 0.2 are not computable, although a present solvent viscosity is taken into account. Moreover, a step-length control (see Ref. [47, 51] for details) is activated in the Newton scheme and an appropriate initial solution is chosen, which does neither lead to a converging nonlinear solver. Similar holds for varying the parameter ε for calculating the approximated Jacobian according to Eq. (2.30). Instead, applying EOFEM stabilisation with respect to the velocity field as well as the Diffusion Tensor, which is presented in Eqs. (2.28) as well as (2.54), respectively, allows for successfully calculating numerical results. Thereby, a relatively large amount of stabilisation has to be taken into account for obtaining solutions of the Tensor Stokes problem (2.51), especially for larger We numbers. Naturally, the stabilisation parameters are intended to be chosen as small as possible to provide a certain accuracy of the numerical results, while the solution becomes inaccurate for a higher or possibly oscillating for a too less amount of stabilisation. In the present case, the stabilisation is applied mainly to prevent the nonlinear solver from diverging, which is further illustrated in Sec. 4.2.1. To be able to reasonably evaluate the results of the Tensor Diffusion approach, the original problem formulation is solved for the same amount of EOFEM stabilisation. In turn, the deviations of results obtained from both formulations compared to the reference increase for higher relaxation times.

We	l	$C_D\,(\mathbf{\Pi}_\sigma)$	N_σ	$C_D\,(\mathbf{\Pi}_\mu)$	N_μ	Ref. [46]	γ_u	γ_σ	γ_μ
0.1	3	130.065	3	130.137	3	130.36	0.0	0.0	0.0
	4	130.283	2	130.303	4		0.0	0.0	0.0
	5	130.342	2	130.348	11		0.0	0.0	0.0
0.2	3	126.360	2	126.628	4	126.62	0.001	0.0	0.001
	4	126.549	3	126.617	4		0.001	0.0	0.001
	5	126.605	2	126.624	4		0.001	0.0	0.001
0.3	3	122.961	2	123.563	4	123.19	0.001	0.0	0.001
	4	123.121	3	123.269	5		0.001	0.0	0.001
	5	123.172	2	123.212	4		0.001	0.0	0.001
0.4	3	120.330	2	121.068	5	120.59	0.05	0.0	0.001
	4	120.486	3	120.597	10		0.05	0.0	0.001
	5	120.552	2	120.548	8		0.05	0.0	0.001
0.5	3	118.458	3	119.095	5	118.83	0.5	0.0	0.1
	4	118.634	3	118.752	4		0.5	0.0	0.1
	5	118.747	3	118.751	3		0.5	0.0	0.1
0.6	3	117.454	3	118.570	4	117.78	0.5	0.0	0.5
	4	117.581	3	118.059	5		0.5	0.0	0.5
	5	117.694	3	117.970	3		0.5	0.0	0.5
0.7	3	116.910	3	118.256	4	117.32	1.0	0.0	1.0
	4	117.072	3	117.629	5		1.0	0.0	1.0
	5	117.214	3	117.545	3		1.0	0.0	1.0

Table 4.2: Drag coefficients resulting from the Oldroyd-B model at $\eta_0 = 1.0$, $\beta = 0.59$ for several We numbers and mesh levels

Nevertheless, the results obtained from the four-field formulation (2.51) of the Tensor Stokes problem show an excellent agreement to the results computed by means of the original method. In detail, the deviations occur only in the decimal places and are thus significantly lower than one percent. Furthermore, the results of the Tensor Diffusion approach show a nice mesh-converged behaviour towards the reference solution from Ref. [46]. The nonlinear solver behaves appropriately as well, because most of the nonlinear iteration numbers are in the same order as in case of the original problem formulation. Note, that the original approach allows for successfully solving for all Weissenberg numbers mentioned in Tab. 4.2 without applying any stabilisation. Hence, the resulting drag coefficients would be even more accurate compared to the reference values, while the solver behaviour is similar to the stabilised setting.

Additionally, cutlines of the normal component of the extra-stress parts of $\mathbf{\Pi}_\sigma$ and $\mathbf{\Pi}_\mu$ acting in flow direction are compared along the surface and wake of the cylinder, that is the bold boundary parts in Fig. 4.18(a). In detail, the component σ_{11} of the extra-stress tensor $\mathbf{\Sigma}$ is plotted plus the corresponding component of the symmetrised stress decomposition from Eq. (2.48) for both, the original model and the four-field formulation of the Tensor Stokes problem. Note, that the symmetrised Tensor Diffusion operator is evaluated in case of the original model during post-processing based on the calculated velocity field.



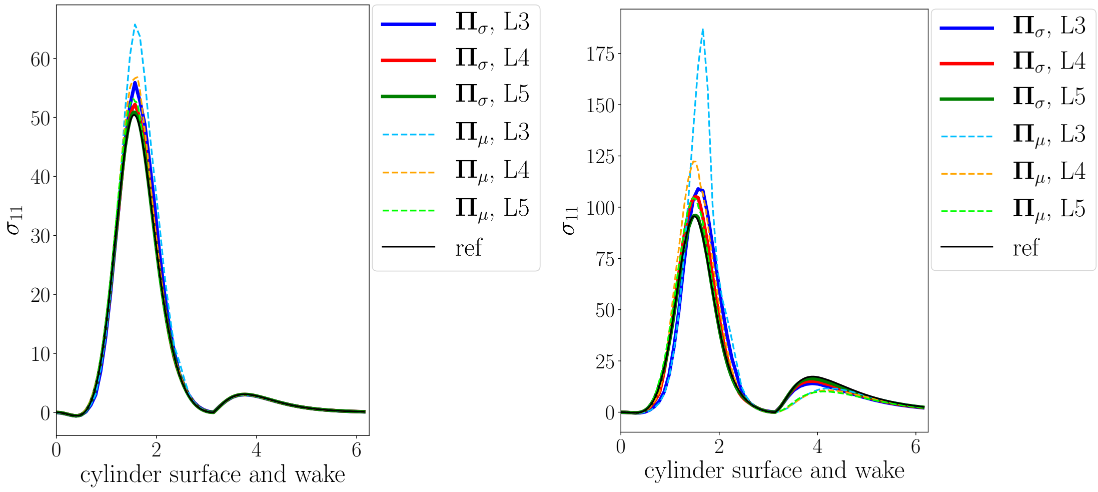
(a) extra-stress tensor

(b) symmetrised Diffusion Tensor

Figure 4.19: Cutlines of the normal stress in flow direction for the Oldroyd-B model at We $= 0.1$, $\eta_0 = 1.0$, $\beta = 0.59$ for several mesh levels

For a low relaxation time of $\Lambda = 0.1$, the cutlines of the component σ_{11} of the extra-stress tensor show a good agreement when computed from Eqs. (2.50) as well as (2.51), among each other and also compared to the "reference" (see Fig. 4.19(a)), which is computed by means of the unstabilised original approach on a finer mesh. In contrast, considering the corresponding component of the symmetrised stress decomposition from Eq. (2.48) leads to deviations of the computational results from the reference. Since $\mathbf{\Sigma}$ is approximated in Q_2, a higher accuracy of the resulting cutlines is expected than based on $\mathbf{M}$ being approximated in Q_0, which explains the inaccuracy of the cutlines from Fig. 4.19(b). Thus, cutlines of $\mathbf{\Sigma}$ – or in fact σ_{11} – resulting from both problem formulations, that is Eqs. (2.50) and (2.51), are considered in further investigations.

Comparing such cutlines for higher relaxation times leads to larger deviations between results concerning $\mathbf{\Pi}_\sigma$ and $\mathbf{\Pi}_\mu$ as well as compared to the "reference", that is the mesh-independent result computed with the validated original method. In case of $\Lambda = 0.3$, probably applying one more mesh refinement would lead to a very good match of σ_{11} resulting from the Tensor Stokes problem with the original results and the reference. Actually, the results on level 5 are already quite close (see Fig. 4.20(a)). Considering the plots for $\Lambda = 0.6$ depicted in Fig. 4.20(b) gives the impression, that σ_{11} from $\mathbf{\Pi}_\mu$ might also converge towards the reference for increasing mesh refinement at least on the cylinder surface. In fact, the corresponding stress cutline on level 5 coincides with the cutline taken on level 4 with respect to the solution of the original approach. But, the results of the Tensor Diffusion approach do not show the expected pronounced stress growth in the wake of the cylinder, since the corresponding stress peak seems to converge to a different value than the original or the reference results. At least, mesh-converged results seem to be obtained as well. Note, that a quite large amount of stabilisation has to be applied to obtain results for this high relaxation time (see Tab. 4.2), which obviously affects the solution quality. Actually, only the stabilisation regarding the velocity (and stress) fields enters the solution in case of the original approach, since $\mathbf{M}$ is determined in post-processing fashion and is not recoupled to $\mathbf{u}$ and $\mathbf{\Sigma}$. In turn, a "doubled" amount of stabilisation is active in the Tensor Diffusion approach compared to the original approach, consequently having a larger impact on the solution quality. But nevertheless, the drag coefficients as well as the stress cutlines obtained by the Tensor Diffusion approach show an acceptable agreement to the original and reference results. Thereby, the (only small or moderate) deviations may arise due to the lower order approximation of $\mathbf{M}$ compared to $\mathbf{\Sigma}$ as well as the applied amount of EOFEM stabilisation, which is required to generate numerical solutions of the Tensor Stokes problem.



(a) $\Lambda = 0.3$

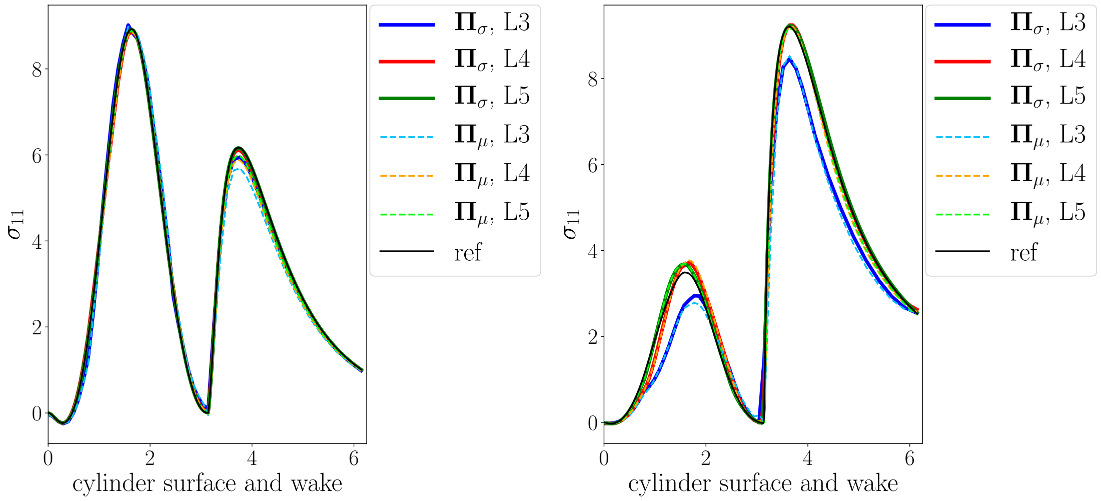
(b) $\Lambda = 0.6$

Figure 4.20: Cutlines of the normal stress in flow direction for the Oldroyd-B model at $\eta_0 = 1.0$, $\beta = 0.59$ for several We numbers and mesh levels

In principle, the same observations are made, when the Giesekus model with a mobility factor of $\alpha = 0.1$ in Eq. (2.5) is considered instead of the Oldroyd-B model. The resulting drag coefficients are listed in Tab. 4.3, where higher Weissenberg numbers compared to the Oldroyd-B model can be reached for applying less stabilisation, probably due to the stabilising character of the quadratic stress term in the constitutive equation [4, 6]. Consequently, the drag coefficients obtained from the Tensor Diffusion approach show an even better agreement to results provided by the original approach as well as the reference, where the latter is not widely available for larger relaxation times. Hence, the Tensor Stokes results for higher Weissenberg numbers are evaluated by a comparison with the original approach, only. Similar to the Oldroyd-B model in Tab. 4.2, the drag coefficients obtained by the Tensor Diffusion approach converge with mesh refinement also in case of the Giesekus model, and thus even for large Weissenberg numbers. Furthermore, the nonlinear solver shows a very stable behaviour as well, which is comparable to the case of the original problem formulation.

We	l	$C_D\,(\mathbf{\Pi}_\sigma)$	N_σ	$C_D\,(\mathbf{\Pi}_\mu)$	N_μ	Ref. [61]	γ_u	γ_σ	γ_μ
0.1	3	125.313	2	125.361	3	125.58	0.001	0.0	0.001
	4	125.512	2	125.526	4		0.001	0.0	0.001
	5	125.567	2	125.572	3		0.001	0.0	0.001
0.5	3	103.533	4	103.718	5	103.73	0.001	0.0	0.001
	4	103.675	5	103.729	4		0.001	0.0	0.001
	5	103.717	5	103.733	5		0.001	0.0	0.001
1.0	3	95.349	2	95.623	5	95.55	0.005	0.0	0.005
	4	95.495	3	95.584	3		0.005	0.0	0.005
	5	95.536	2	95.568	3		0.005	0.0	0.005
5.0	3	85.029	4	85.261	5	–	0.005	0.01	0.005
	4	85.166	3	85.243	4		0.005	0.01	0.005
	5	85.210	3	85.248	6		0.005	0.01	0.005
10.0	3	82.853	2	83.006	4	–	0.005	0.01	0.005
	4	83.012	4	83.069	4		0.005	0.01	0.005
	5	83.047	4	83.068	6		0.005	0.01	0.005

Table 4.3: Drag coefficients resulting from the Giesekus model at $\alpha = 0.1$, $\eta_0 = 1.0$, $\beta = 0.59$ for several We numbers and mesh levels



(a) $\Lambda = 1.0$

(b) $\Lambda = 10.0$

Figure 4.21: Cutlines of the normal stress in flow direction for the Giesekus model at $\alpha = 0.1$, $\eta_0 = 1.0$, $\beta = 0.59$ for several We numbers and mesh levels

In addition, the corresponding cutlines of the extra-stress tensor component σ_{11} match the original and reference results very well, even for higher relaxation times (see Fig. 4.21). Note, that the stress curves for increasing Λ show a different behaviour than for the Oldroyd-B model. In detail, the stress stays more or less bounded for the Giesekus model, probably due to the stabilising quadratic contribution in the constitutive equation [6]. Moreover, for increasing relaxation times the stress peak in the wake of the cylinder becomes larger, while the stress on the cylinder surface is decreased. In contrast, the magnitude of both stress peaks grows according to Λ in case of the Oldroyd-B model, where similar effects occur in terms of the fully developed flow profiles calculated in Sec. 3.1.

Overall, the Tensor Diffusion approach gives satisfactory results in terms of the Flow around cylinder benchmark, especially for lower relaxation times in case of the Oldroyd-B model, but also for higher relaxation times in case of the Giesekus model. For computing higher Weissenberg numbers regarding the Oldroyd-B model, a more or less significant amount of EOFEM stabilisation has to be inserted into the (discrete) system for obtaining a converging nonlinear solver or a stable solution. Nevertheless, the corresponding solution quality is still sufficiently high to emphasise, that the Tensor Diffusion approach could provide a reasonable numerical scheme for simulating viscoelastic fluid flows – particularly bearing in mind the applied lower-order approximation of $\mathbf{M}$. A further study on the EOFEM stabilisation concerning the shape and stability of the Diffusion Tensor variable is performed in Sec. 4.2.1, while the validation of the Tensor Diffusion approach is continued in the following.

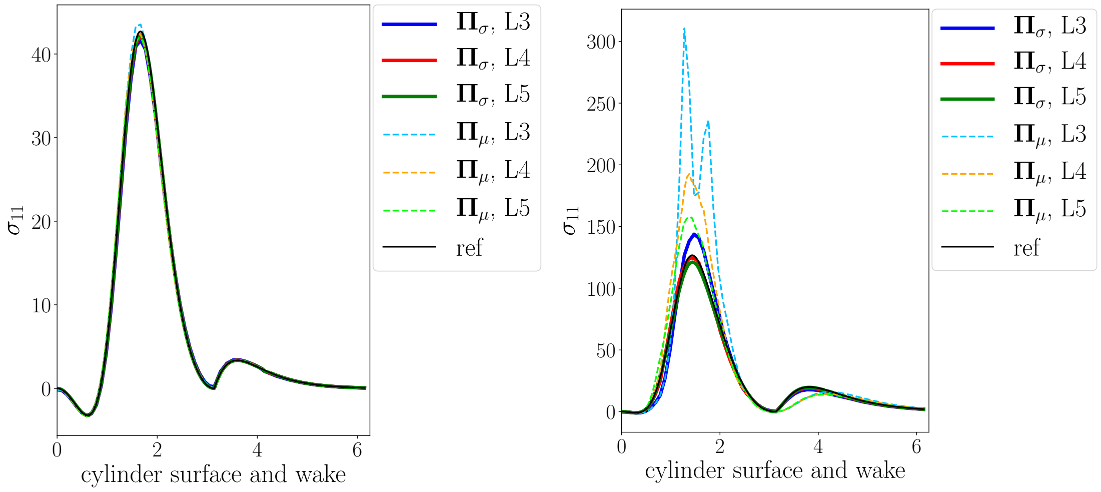
The non-solvent case

The more challenging configuration compared to the setting discussed above is represented by considering the non-solvent case of the Flow around cylinder benchmark. Unfortunately, no reference results are available for this flow configuration in case of considering the Giesekus model, which is why the corresponding solutions of the Tensor Stokes problem are again compared below only with the original approach. Furthermore, the simulation results are not evaluated with respect to mesh refinement, since it is considered as sufficient to focus on the overall solution quality of the Tensor Diffusion approach just by means of results computed on a fine mesh.

We	$C_D\,(\mathbf{\Pi}_\sigma)$	N_σ	$C_D\,(\mathbf{\Pi}_\mu)$	N_μ	Ref. [62]	γ_u	γ_σ	γ_μ
0.1	127.373	3	127.403	3	127.41	0.1	0.0	0.1
0.2	117.782	3	117.899	6	117.81	0.1	0.0	0.1
0.3	108.626	3	109.091	7	108.66	0.25	0.0	0.25
0.4	101.372	5	102.461	6	101.41	0.5	0.0	0.5
0.5	96.046	4	98.054	5	96.08	1.0	0.0	1.0
0.6	92.301	4	95.054	9	92.33	1.5	0.0	1.5
0.7	89.754	4	93.184	8	89.79	2.0	0.0	2.0

Table 4.4: Drag coefficients resulting from the UCM model at $\eta_0 = 1.0$ for several We numbers on level 5

In a first step, the four-field formulations from Eqs. (2.50) and (2.51) are considered with respect to the UCM model, that is choosing the model function from Eq. (2.4) in the constitutive equation and setting $\eta_s = 0$ in the momentum equation of the flow model. Besides stabilising the Diffusion Tensor variable similar to $\eta_s > 0$, certain difficulties in the context of the non-solvent case outlined in Sec. 2.4 need to be treated as well. Consequently, a quite large amount of stabilisation is required for successfully performing corresponding simulations. However, the Tensor Stokes results show a good agreement to the results of the original problem as well as the reference from Ref. [62] when analysing the calculated drag coefficients given in Tab. 4.4, especially for lower We. For higher Weissenberg numbers, the deviations of the Tensor Stokes to the original and reference results increase due to a larger amount of stabilisation being applied especially regarding the Diffusion Tensor. Recall, that the same amount of stabilisation is taken into account in the Tensor Diffusion as well as the original approach. In the latter, the stabilisation with respect to $\mathbf{M}$ is not active, which is why the corresponding results are closer to the reference. In fact, numerical solutions can be successfully computed by means of the unstabilised original approach until We $= 0.4$, which in principle reproduces the drag values from the reference. Also regarding higher relaxation times, only a slight amount of EOFEM stabilisation in the range of $\gamma_u \in [0.001, 0.01]$ needs to be applied for obtaining numerical solutions from the original approach, which still leads to simulation results close to the reference. Nevertheless, applying the Tensor Diffusion approach gives results of a similar accuracy as the original approach regarding drag coefficients as well as solver behaviour. In detail, the maximum deviation in the drag coefficient is around 2.5% and the nonlinear iteration numbers are at most a factor of two larger, which is why the results from the Tensor Stokes problem are of acceptable quality.



(a) $\Lambda = 0.1$

(b) $\Lambda = 0.5$

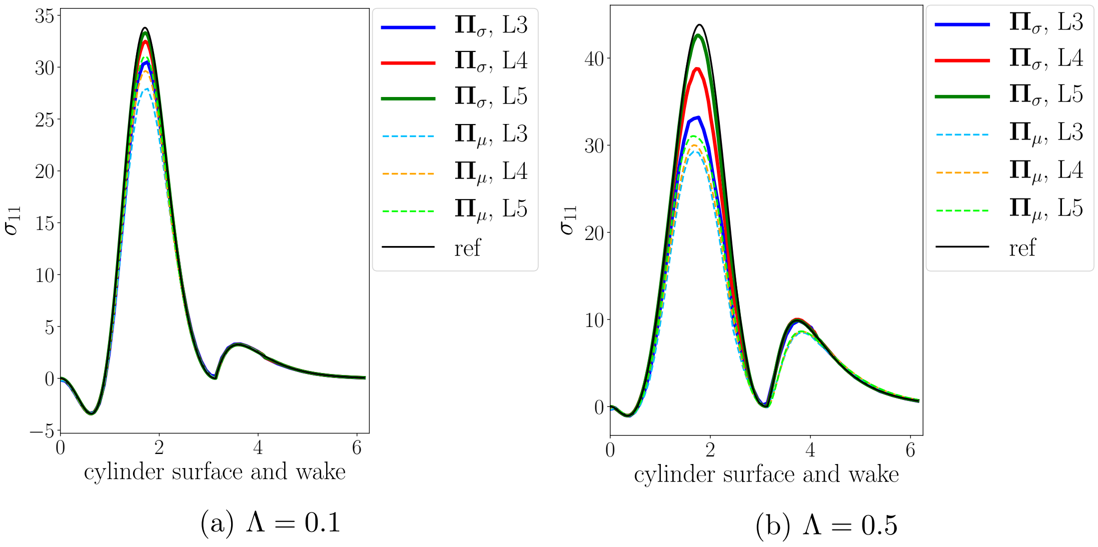
Figure 4.22: Cutlines of the normal stress in flow direction for the UCM model at $\eta_0 = 1.0$ for several We numbers and mesh levels

In principle, similar observations are made for the corresponding stress cutlines depicted in Fig. 4.22. Based on the results of the Tensor Stokes problem, a very good agreement with the original and reference results is observed for a small relaxation time of $\Lambda = 0.1$, which may be further improved when considering an even finer mesh level. Naturally, the deviations increase for higher Λ and an accordingly increasing amount of stabilisation, which leads to deviations of 30% regarding the first stress peak at $\Lambda = 0.5$. Possibly, this will also be reduced on finer mesh leves, but the results obtained on mesh level 5 differ significantly. In the wake, again a less pronounced and slightly shifted stress peak results from the Tensor Stokes solution, which at least shows a mesh-converged behaviour.

We	$C_D\,(\mathbf{\Pi}_\sigma)$	N_σ	$C_D\,(\mathbf{\Pi}_\mu)$	N_μ	γ_u	γ_σ	γ_μ
0.1	115.377	3	115.508	3	0.1	0.0	0.1
0.2	93.855	3	94.166	5	0.1	0.0	0.1
0.3	78.822	3	79.517	3	0.5	0.0	0.5
0.4	68.401	3	69.222	7	0.5	0.0	0.5
0.5	60.804	8	61.992	4	1.0	0.0	1.0
0.6	55.059	9	56.509	4	1.5	0.0	1.5
0.7	50.472	9	52.191	4	2.0	0.0	2.0

Table 4.5: Drag coefficients resulting from the Giesekus model at $\alpha = 0.1$, $\eta_0 = 1.0$, $\beta = 0.0$ for several We numbers on level 5

Again, the same configuration is simulated for the Giesekus model including a mobility factor of $\alpha = 0.1$, which results in an appropriate quality of the drag values provided by the Tensor Diffusion approach. But, the deviations to the original approach increase for higher Weissenberg numbers, probably due to the required large amount of stabilisation, which is also applied with respect to the Diffusion Tensor $\mathbf{M}$ (see Tab. 4.5). But, the maximum deviation is still below 5%. Moreover, the number of nonlinear iterations is in the same range as in case of $\eta_s > 0$ and the solver concerning the Tensor Stokes problem seems to be superior to the original approach for higher relaxation times.



(a) $\Lambda = 0.1$

(b) $\Lambda = 0.5$

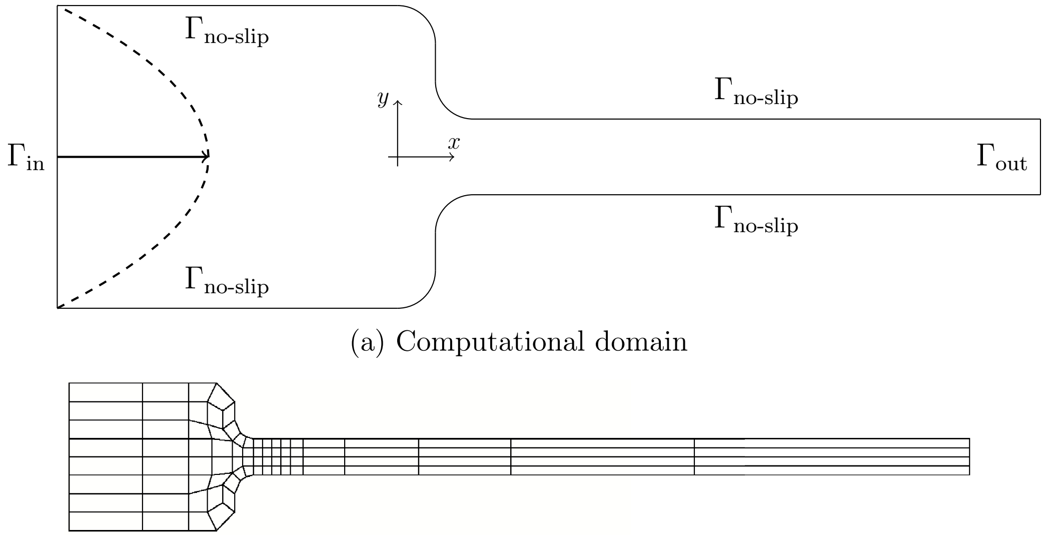
Figure 4.23: Cutlines of the normal stress in flow direction for the Giesekus model at $\alpha = 0.1$, $\eta_0 = 1.0$, $\beta = 0.0$ for several We numbers and mesh levels

It becomes clear, that the results of the Tensor Diffusion approach are not as accurate as in case of a present solvent contribution, which is also realised by means of the stress cutlines depicted in Fig. 4.23. In terms of accuracy on the finest mesh, the results are roughly of the same quality as for the (actually more unstable) UCM model, probably due to the required large amount of stabilisation plus the lower-order approximation of $\mathbf{M}$ in Q_0. But, the Tensor Stokes results obtained from the Giesekus model are smoother than in case of the UCM model, and at the same time the stress peak in the wake is reproduced much better. However, for the Giesekus model it is not possible to reach significantly larger Weissenberg numbers than for the UCM model – in contrast to a present solvent viscosity, which further illustrates the complexity of this flow configuration. But still, the Tensor Diffusion approach provides reasonable results, since the drag coefficients are quite close to the "reference" values and in similar way the stress cutlines qualitatively correlate to the results obtained from the original approach.

In summary, the Tensor Diffusion approach leads to very acceptable results compared to the original (differential) viscoelastic problem formulation also in case of actual two-dimensional flow configurations like the Flow around cylinder benchmark – even for the very challenging non-solvent case. Together with the validation performed in the context of Poiseuille-like flows in Sec. 4.1.1, the basic implementation of the Tensor Diffusion approach discussed above may be regarded as a (potentially) reasonable numerical procedure for simulating viscoelastic fluid flows. However, no significant improvement compared to the original numerical technique is established at the current stage of the apporach. But, it has to be emphasised that the quite accurate results of the Tensor Diffusion approach are generated despite the lower-order approximation of the tensor-valued viscosity as well as a partly significant amount of stabilisation. Naturally, this novel approach needs to be further developed by improving the numerical calculation of the Diffusion Tensor, for example by means of the PDE approach illustrated in Sec. (2.4), the design of suitable solution techniques and/or an explicit modelling of the introduced tensor-valued viscosity. In the following section, the Tensor Diffusion approach is further evaluated in terms of two-dimensional Finite Element simulations to detect the limits as well as highlight the actual benefits of the current implementation.

4.2 Evaluation for contraction flows

After having validated the Tensor Diffusion approach in the previous section, some further aspects of this novel approach are analysed in the following. First, the actual shape of the components of the Diffusion Tensor is studied in Sec. 4.2.1 in terms of two-dimensional flow configurations, as this is done so far only in the context of (fully developed) channel flows. Thereby, the purpose of the specific discretisation and stabilisation techniques regarding the Diffusion Tensor variable, which are proposed in Sec. 2.4, are illustrated. Furthermore, the improvement of the linear solver within the Newton scheme is investigated in Sec. 4.2.2 in terms of applying multigrid techniques in case of a vanishing solvent viscosity. Recall, that the linear systems are solved in the non-solvent case of the original approach typically via direct solvers, which imposes certain restrictions on the considered problem. Finally, the Tensor Diffusion approach is conceptually analysed concerning the High Weissenberg Number Problem in Sec. 4.2.3, followed by an evaluation of prototypical iteration schemes including the Diffusion Tensor, where a fully coupled as well as decoupled approach is taken into account in Sec. 4.2.4. The underlying configuration considered in the current section is given by a viscoelastic fluid flow within a contraction. Thereby, a rounded instead of an abrupt contraction is taken into account to avoid numerical difficulties and oscillations arising due to sharp corners, although the latter are typically regarded in benchmark computations [63]. Moreover, the present configuration deviates from the common setting by considering the full geometry instead of only the upper half (see Fig. 4.24(a)).


$\Gamma_{\text{no-slip}}$
$\Gamma_{\text{no-slip}}$
y
x
Γ_{in}
Γ_{out}
$\Gamma_{\text{no-slip}}$
$\Gamma_{\text{no-slip}}$

(a) Computational domain

(b) Coarse mesh

Figure 4.24: Configuration of the rounded contraction flow

By doing so, there is no need to set do-nothing or symmetry boundary conditions on the symmetry axis of the computational domain, which might cause unphysical effects. These modifications may simplify the complexity of the resulting (discrete) problem, since numerical difficulties arising from the overall configuration, for example the geometrical shape, are intended to be avoided. Thus, it should be possible to analyse the numerical treatment of the novel Tensor Diffusion approach, without taking account additional numerical issues. The computational domain regarded in the following consists of a 4:1 contraction, where the in- and outflow egdes are of height $y \in [-4, 4]$ and $y \in [-1, 1]$, respectively. The entrance section of the contraction is realised by means of two quarter circles of radius 1 on the upper as well as lower half of the geometry, which are located at $(x, y) = (-1, \pm 3)$ and $(x, y) = (1, \pm 2)$. The up- and downstream channels are of length 8 and 40, respectively. Information on the considered computational meshes is given in Tab. 4.6 based on the coarse mesh depicted in Fig. 4.24(b), where the refinement levels as well as the degrees of freedom are to be understood the same way as in Sec. 4.1.2.

level l	elements	nodes	edges	degrees of freedom
0	84	109	192	2513
1	336	385	720	9557
2	1344	1441	2784	37253
3	5376	5569	10944	147077
4	21504	21889	43392	584453

Table 4.6: Mesh information for the rounded contraction flow

Similar to the Flow around cylinder benchmark, the original problem formulation from Eq. (2.16) (or (2.50)) as well as the four-field formulation of the Tensor Stokes problem from Eq. (2.51) are treated below in terms of two-dimensional Finite Element simulations. Thus, again the discretisation and solution techniques presented in Secs. 2.2.3 and 2.4 are applied, where the arising discrete systems typically are solved monolithically by means of Newton's method based on Eq. (2.29), including a direct linear solver. In contrast, applying multigrid techniques in terms of the Tensor Diffusion approach is investigated in Sec. 4.2.2, while a decoupled solution approach is discussed in Sec. 4.2.4, which consists of solving a set of linear subproblems of Eq. (2.51).

Concerning boundary conditions, a fully developed velocity profile according to the specific material model is imposed on the in- and outflow edges Γ_{in} and Γ_{out} of the computational domain, together with no-slip on the remaining boundaries. The corresponding stress profiles are prescribed only on the inflow egde Γ_{in}, while do-nothing boundary conditions regarding the stress tensor are set on all other boundaries. Similar to the Flow around cylinder benchmark discussed above, no boundary conditions need to be assigned to the pressure field as well as the Diffusion Tensor. Regarding the velocity field, the profiles

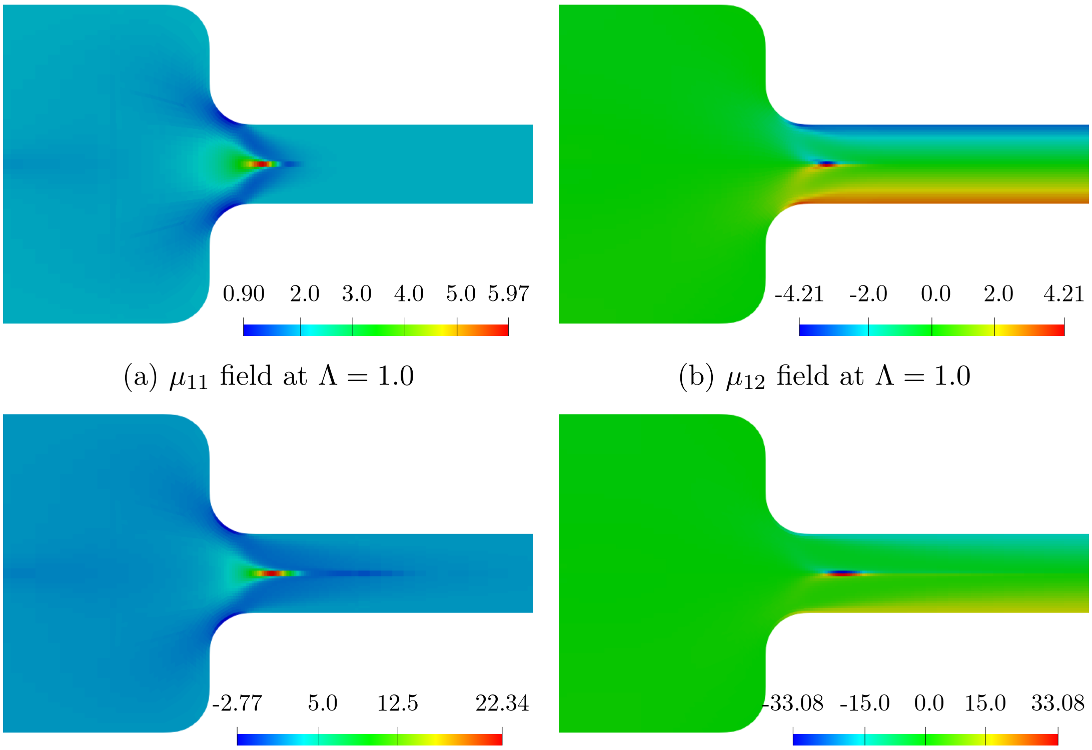
$$u_{\text{in}}\left(y\right) = \frac{U}{16}\left(16 - y^2\right),\ y \in [-4,4], \quad u_{\text{out}}\left(y\right) = 4U\left(1 - y^2\right),\ y \in [-1,1] \tag{4.15}$$

are applied in case of the UCM model, where a velocity parameter of $U = 0.1$ is considered giving a mean velocity of $U_{\text{mean}} = 0.2\bar{6}$ in the downstream channel of height 2. Hence, the resulting flow leads to a Weissenberg number of $\text{We} = \Lambda\frac{U_{\text{mean}}}{2} = 0.1\bar{3}\Lambda$. Besides, the same average velocity needs to be obtained by the boundary data computed within the framework introduced in Sec. 3.1 as well, in case a nonlinear material model is chosen. However, the exponential PTT model is taken into account in the following studies, although it is not evaluated above in terms of fully developed channel flows. But, it makes sense to consider such a well-defined constitutive equation, since especially in Sec. 4.2.3 viscoelastic fluid flow simulations at higher relaxation times are investigated. Thus, underlying limits, for example provided by means of the Giesekus model, are avoided and do not affect the analysis of the Tensor Diffusion approach. For completeness, the exponential PTT model is briefly analysed in the subsequent sections as well, that is also with respect to the stabilisation of the Diffusion Tensor variable and the application of multigrid techniques.

4.2.1 Stabilising the Diffusion Tensor variable

In the previous investigations, the structure of the Diffusion Tensor is discussed only in the context of channel flows, that is basically in terms of one-dimensional configurations. In this context, the arising tensor-valued viscosity can be given even analytically in case of considering certain material models, which is not possible regarding actual two-dimensional flow states. Naturally, the corresponding shape of the Diffusion Tensor is of interest as well, to be able to derive a similar explicit representation regarding more complex configurations. But, even for applying the Tensor Diffusion approach to simulate the Flow around cylinder benchmark, no study is performed with respect to the shape of this tensor-valued quantity. Although no according analytical expression or simple representation can be given, the resulting Diffusion Tensor is of minor interest, since the main goal is to reproduce reference results regarding the velocity and stress fields. Hence, a corresponding analysis concerning the shape of the Diffusion Tensor is performed below, especially demonstrating the use of the particular Finite Element approximation as well as EOFEM stabilisation with respect to $\mathbf{M}$ proposed in Sec. 2.4.

In this regard, the four-field formulation of the Tensor Stokes problem is solved in a first step for the UCM model on level 3 of the coarse mesh depicted in Fig. 4.24(b), where the Diffusion Tensor is approximated by means of element-wise constant polynomials. Simulation results regarding the components of $\mathbf{M}$ for the two relaxation times $\Lambda = 1.0$ and $\Lambda = 5.0$ are depicted in Fig. 4.25. Note, that $\mathbf{M}$ evolves to the fully developed shape given in Sec. 4.1.1 in the up- and downstream channels, which are not presented below in full. In addition, the shape of the components of the Diffusion Tensor is discussed at the example of μ_{11} and μ_{12}, since the remaining components μ_{21} and μ_{22} behave similarly.



(a) μ_{11} field at $\Lambda = 1.0$

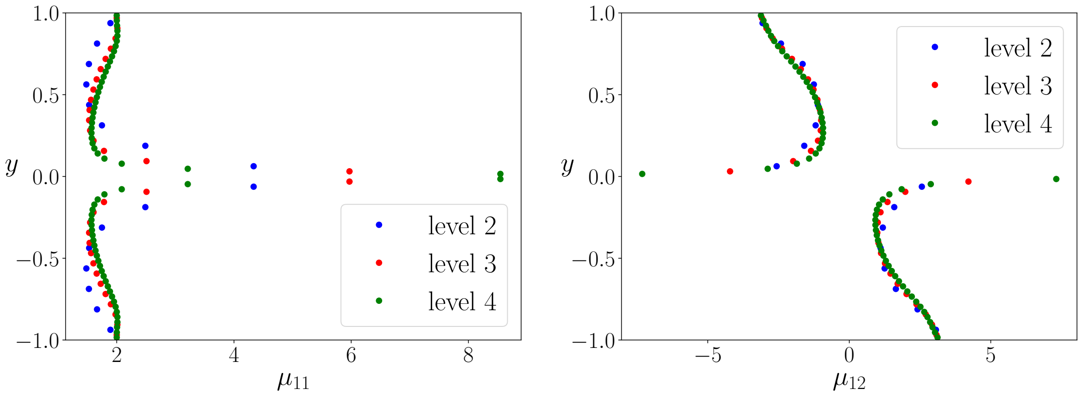
(b) μ_{12} field at $\Lambda = 1.0$

(c) μ_{11} field at $\Lambda = 5.0$

(d) μ_{12} field at $\Lambda = 5.0$

Figure 4.25: Diffusion Tensor solution in Q_0 for the UCM model at $\eta_0 = 1.0$ for EOFEM parameters $\gamma_u = 0.1$, $\gamma_\sigma = 0.01$, $\gamma_\mu = 0.0$, on level 3

It turns out, that the Diffusion Tensor exhibits singularities or pronounced discontinuities building up right behind the contraction, which decay relatively fast in the downstream channel. Thereby, the magnitude of the peaks is proportional to Λ and also increases with mesh refinement. The latter is illustrated by the plots of the $\mathbf{M}$-components over the channel height at the x-position of the actual maximum peak (see Fig. 4.26), which varies more or less slightly depending on the mesh level and the specific component. Moreover, μ_{11} and μ_{22} behave according to y^{-m} close to $y = 0$ for an even $m \in \mathbb{N}$, while the off-diagonal components μ_{12} and μ_{21} correspond to $(\mp y)^{-m}$ for an odd m.

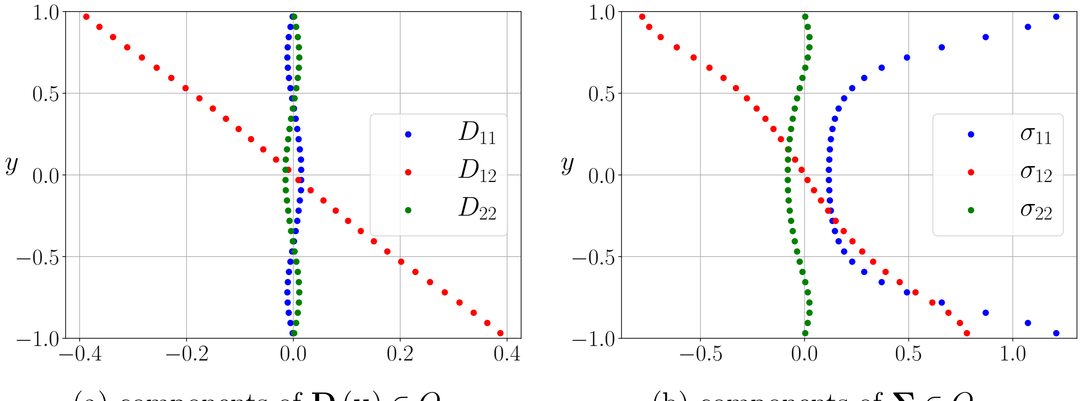


(a) $\mu_{11} \in Q_0$

(b) $\mu_{12} \in Q_0$

Figure 4.26: Diffusion Tensor profiles for the UCM model at $\Lambda = 1.0$, $\eta_0 = 1.0$ for EOFEM parameters $\gamma_u = 0.1$, $\gamma_\sigma = 0.01$, $\gamma_\mu = 0.0$ at the maximum $\mathbf{M}$-peak for several mesh levels

Although the highlighted behaviour of the Diffusion Tensor seems to be unphysical – in case this tensor-valued viscosity can be interpreted as a physical quantity at all – there are several aspects indicating, that indeed its natural form is observed: In fact, the cutlines of the components of the strain-rate as well as extra-stress tensor in this section of the downstream channel depicted in Fig. 4.27 illustrate, that no numerical artefacts are observed. Instead, the small values of $\mathbf{D}(\mathbf{u})$ together with the rather large values of $\mathbf{\Sigma}$ naturally lead to large values of $\mathbf{M}$ based on the stress decomposition from Eq. (2.47). Furthermore, there is a certain smooth build-up towards the singularity on the symmetry axis located at $y = 0$ (see Figs. 4.26 and 4.28). Moreover, no oscillating behaviour or sudden appearance of perturbations in the solution is observed, which would indicate a numerical source of the peaks arising in $\mathbf{M}$.



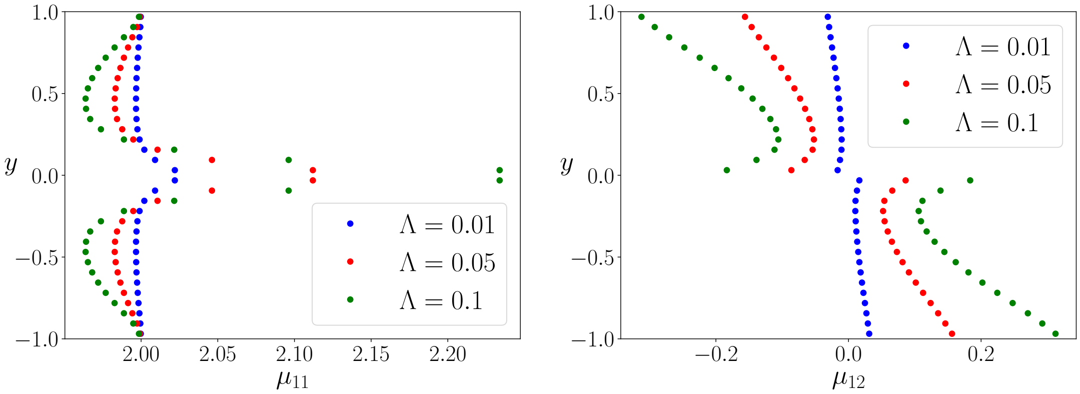
(a) components of $\mathbf{D}(\mathbf{u}) \in Q_0$

(b) components of $\mathbf{\Sigma} \in Q_0$

Figure 4.27: Cutlines of flow quantities for the UCM model at $\Lambda = 5.0$, $\eta_0 = 1.0$ for EOFEM parameters $\gamma_u = 0.1$, $\gamma_\sigma = 0.01$, $\gamma_\mu = 0.0$ at the maximum $\mathbf{M}$-peak on level 3

Furthermore, the singular shape of $\mathbf{M}$ successively evolves for increasing relaxation times or Weissenberg numbers. Thus, a continuous growth of the Diffusion Tensor depending on the prescribed parameter(s) is observed concerning the profiles of $\mathbf{M}$

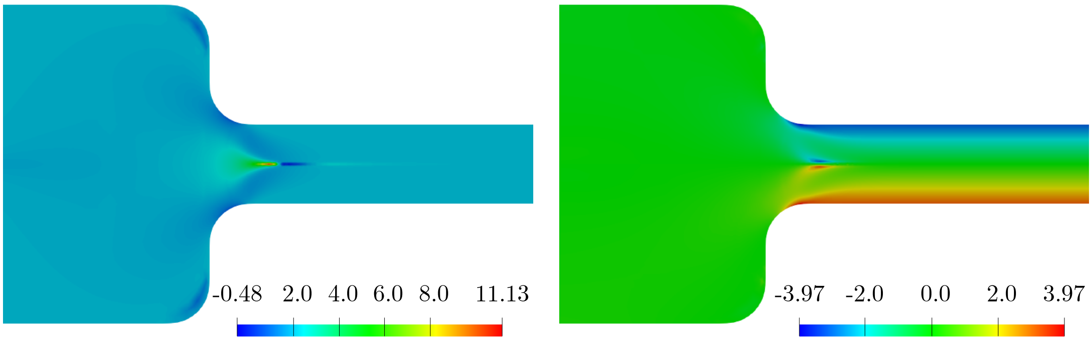
depicted in Fig. 4.28. In addition, the Diffusion Tensor indeed might consist of a large gradient close to the symmetry line, which is discussed in Sec. 4.1.1 in case of channel flows. Hence, a similar behaviour could also occur in more complex cases like considered here, but a higher resolution might be required for appropriately capturing the components of $\mathbf{M}$.



(a) $\mu_{11} \in Q_0$ (b) $\mu_{12} \in Q_0$

Figure 4.28: Diffusion Tensor profiles for the UCM model at $\eta_0 = 1.0$ for EOFEM parameters $\gamma_u = 0.1$, $\gamma_\sigma = 0.01$, $\gamma_\mu = 0.0$ at the maximum $\mathbf{M}$-peak on level 3 for several relaxation times

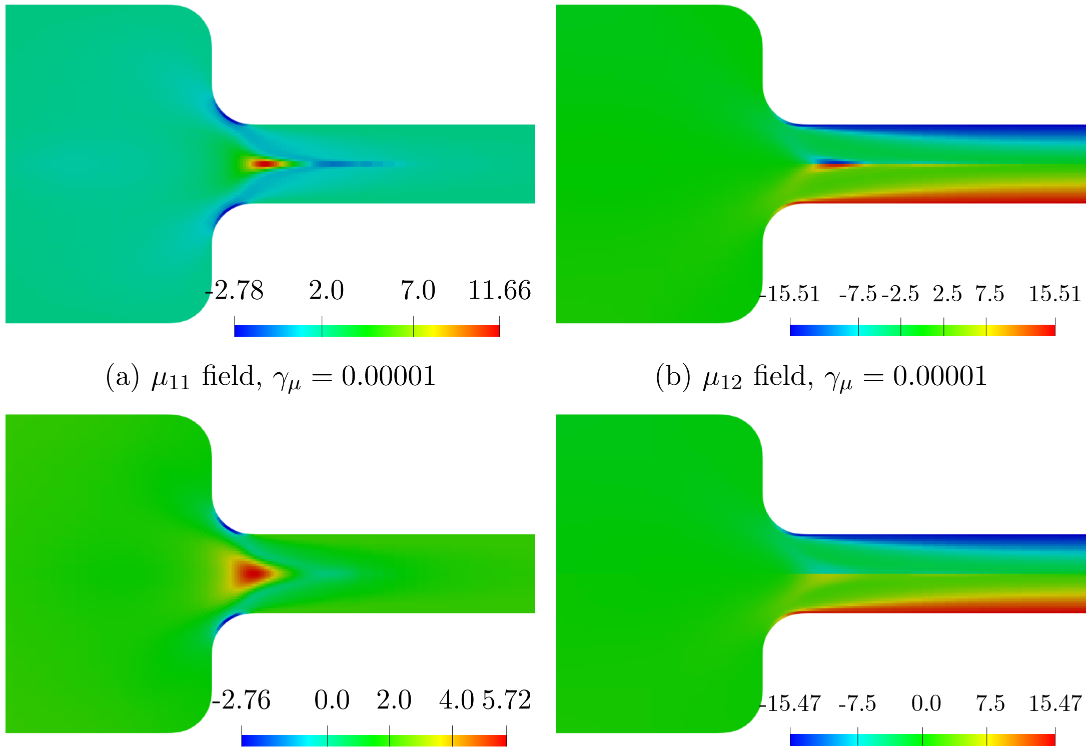
Consequently, the approximation of the Diffusion Tensor as element-wise constant quantity, which is proposed in terms of the Finite Element techniques described in Sec. 2.4, indeed is a reasonable choice for the following analysis as well as the benchmark computations from Sec. 4.1.2. In particular, jumps of $\mathbf{M}$ over element edges are allowed by means of this discretisation, as such phenomena seem to arise naturally in the corresponding variable. But at the same time, singular effects within the Diffusion Tensor might not be optimal regarding the numerical treatment of the Tensor Stokes problem. This is outlined in the subsequent sections, escpecially concerning the behaviour of the iterative solvers or the stability of the corresponding solutions. Thus, enabling a global continuity of $\mathbf{M}$ or at least damped singularities is a reasonable goal.



(a) Q_1-interpolant of $\mu_{11} \in Q_2$ (b) Q_1-interpolant of $\mu_{12} \in Q_2$

Figure 4.29: Diffusion Tensor solution for the UCM model at $\Lambda = 1.0$, $\eta_0 = 1.0$ for EOFEM parameters $\gamma_u = 0.1$, $\gamma_\sigma = 0.01$, $\gamma_\mu = 0.0$, on level 3

Therefore, the Diffusion Tensor may be approximated by means of biquadratic Finite Elements even within the algebraic approach, which is also proposed in the context of the three-field formulation of the Tensor Stokes problem from Eq. (2.57). But, this is not a satisfying approach, since the “singularities” in $\mathbf{M}$ are still present in the approximations in Q_2, and the corresponding solution consists of even larger peaks (see Fig. 4.29). The latter is observed due to some of the considered degrees of freedom being located on the symmetry axis. Thus, the obtained eigenvalues of $\mathbf{D}(\mathbf{u})$ are probably even smaller than in the midpoints of elements next to $y = 0$ corresponding to Q_0 as indicated by means of Fig. 4.27. Possibly, an overall improved approach for calculating the Diffusion Tensor is provided in terms of the three-field formulation of the Tensor Stokes problem illustrated in Sec. 2.4, which enables a global coupling of the degrees of freedem concerning $\mathbf{M}$.



(a) μ_{11} field, $\gamma_\mu = 0.00001$

(b) μ_{12} field, $\gamma_\mu = 0.00001$

(c) μ_{11} field, $\gamma_\mu = 0.0005$

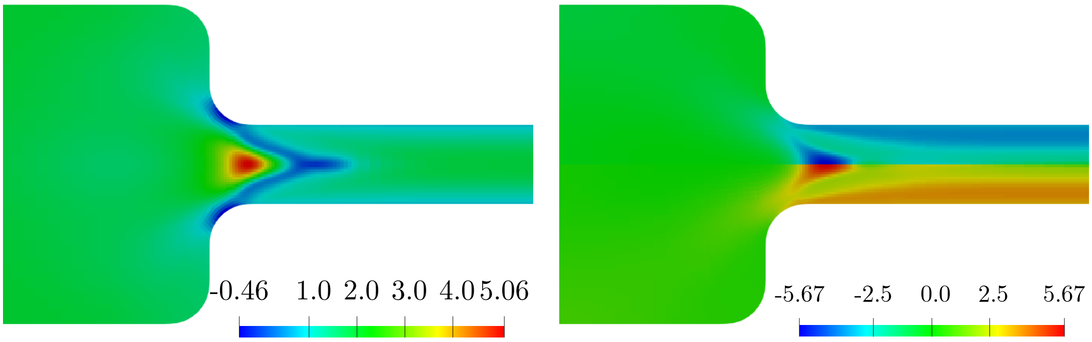
(d) μ_{12} field, $\gamma_\mu = 0.0005$

Figure 4.30: Diffusion Tensor solution for the UCM model at $\Lambda = 5.0$, $\eta_0 = 1.0$ for EOFEM parameters $\gamma_u = 0.1$, $\gamma_\sigma = 0.01$ and several choices of γ_μ on level 3

As an alternative, the jumps in the Diffusion Tensor can be damped by introducing EOFEM stabilisation with respect to $\mathbf{M}$ by means of the (discrete) operator proposed in Eq. (2.54), which is already successfully applied in Sec. 4.1.2. The components of the Diffusion Tensor resulting from the UCM model at a relaxation time of $\Lambda = 5.0$ are depicted in Fig. 4.30 regarding two choices of the stabilisation parameter γ_μ, while the undamped fields are presented in Fig. 4.25. Indeed, a smoothening effect especially regarding the jumps in $\mathbf{M}$ discussed above is observed, while including a too large amount of stabilisation naturally leads to artificial representations. However, damping the jumps in the Diffusion Tensor mainly improves the behaviour of the numerical solver, which is pointed out in Secs. 4.1.2 and 4.2.3. At the same time, the solution regarding the primal variables $(\mathbf{u}, \mathbf{\Sigma}, p)$

and also $\mathbf{D}(\mathbf{u})$ might consist of numerical oscillations, which are not significantly improved by applying EOFEM stabilisation with respect to $\mathbf{M}$. Instead, this is achieved in terms of directly stabilising the velocity or stress variables.

For completeness, the Diffusion Tensor resulting from a nonlinear differential viscoelastic model, in detail the exponential version of the PTT model including the model function from Eq. (2.6b), is depicted in Fig. 4.31. The resulting components of the Diffusion Tensor for a relaxation time of $\Lambda = 5.0$ and a model parameter of $\kappa = 0.1$ principally show the same behaviour as in case of the UCM model. But, the effect of applying a relatively small amount of EOFEM stabilisation seems to be too strong especially concerning μ_{12} from Fig. 4.31(b), since this quantity is not "smooth" along the downstream channel. Unfortunately, no solution can be computed from the Tensor Stokes problem in case of a lower value of γ_μ even for applying a damped Newton scheme and performing a successive reduction of the stabilisation parameter. In fact, the solution regarding a vanishing stabilisation with respect to $\mathbf{M}$ can be computed for $\Lambda \leq 0.1$ only, where the resulting Diffusion Tensor naturally looks similar to the UCM-case. Thus, the four-field formulation of the Tensor Stokes problem including the exponential PTT model at higher relaxation times can not be solved successfully without applying EOFEM stabilisation with respect to $\mathbf{M}$ – in contrast to the UCM model.



(a) μ_{11} field (b) μ_{12} field

Figure 4.31: Diffusion Tensor solution for the PTT model at $\Lambda = 5.0$, $\kappa = 0.1$, $\eta_0 = 1.0$, $\beta = 0.0$ for EOFEM parameters $\gamma_u = 0.00125$, $\gamma_\sigma = 0.0$, $\gamma_\mu = 0.0005$ on level 3

Overall, it becomes clear that discretising the Diffusion Tensor by means of a discontinuous Finite Element approximation is a reasonable approach, since discontinuities actually occur in the corresponding flow field. Moreover, the behaviour of the numerical solver probably is improved by introducing EOFEM stabilisation with respect to $\mathbf{M}$, since the jumps in the Diffusion Tensor are damped. At the same time, it makes sense to increase the amount of stabilisation for higher Weissenberg numbers due to the accordingly increasing "singularities" in $\mathbf{M}$. Note, that these aspects of stabilising the Diffusion Tensor variable are already exploited in terms of simulating the Flow around cylinder benchmark in Sec. 4.1.2. Nevertheless, the choice of the EOFEM parameter and the way, in which the stabilisation regarding the Diffusion Tensor is introduced in Sec. 2.4, may be improved. In detail, the stabilisation parameter might be linked to the (local) mesh size to apply a comparable amount of stabilisation on each considered mesh level. Furthermore, the EOFEM stabilisation could be applied only locally to damp the significant jumps in the Diffusion Tensor, while it might be neglected in the remaining part of the computational domain.

4.2.2 Multigrid techniques within the Tensor Diffusion approach

In terms of the monolithic solution scheme regarding the simulation of non-solvent viscoelastic fluid flows applied in this work, the linear systems arising within Newton's method typically are solved by means of a direct solver. Hence, corresponding numerical investigations are limited to small or moderate problem sizes, that is in particular relatively coarse meshes, due to the involved quite extensive memory requirements. Instead, applying multigrid techniques as described in Sec. 2.2.3 offers a promising alternative for obtaining an efficient (as well as fast) numerical solution scheme. But, the vanishing diffusive operator in the momentum equation of the original problem formulation leads to numerical difficulties in combination with multigrid solvers as outlined below. Based on that, it is demonstrated that inserting Tensor Diffusion into the (discrete) problem indeed allows for successfully applying multigrid techniques in terms of numerically treating viscoelastic fluid flow models without a solvent contribution to the viscosity. Therefore, the four-field formulation of the Tensor Stokes problem from Eq. (2.51) is solved below by means of Newton-multigrid techniques for the flow through the rounded contraction depicted in Fig. 4.24. In detail, the applied multigrid solver performs an F-cycle between a coarse mesh of level $l_{\min} \in \{1, 2, 3\}$ and a fine mesh of level $l_{\max} \in \{3, 4\}$, which are connected by suitable grid transfer operators as described in Sec. 2.2.3. Information on the considered mesh levels can be found in Tab. 4.6. Furthermore, a Vanka-like smoother is utilised including a number of n_s smoothing steps as well as a damping factor ω_s when updating the solution vector. Additionally, damping might also be applied when prolongating the solution from a lower to a higher mesh level, which is realised by the weighting factor ω_p. Besides, the number N of nonlinear steps, the number L of average linear steps per nonlinear iteration and the applied parameters regarding the EOFEM stabilisation are given below as well when discussing the simulation results. However, a relative tolerance of 0.1 is defined within the multigrid solver, while an absolute tolerance of 10^{-8} is assigned to the outer Newton iteration. Naturally, reducing the tolerance of the linear solver will lead to an increasing average number of linear iterations (and probably a lower number of nonlinear steps), but a diverging linear solver will not become convergent.

In a first step, the failure of applying multigrid solvers is pointed out regarding the non-solvent case of the original approach, that is with respect to the problem formulation from Eq. (2.16). In doing so, the UCM model is considered for the material parameters defined above, in detail a velocity parameter of $U = 0.1$ in Eq. (4.15) and a total viscosity of $\eta_0 = 1.0$. For starting off with a quite easy configuration, the solver behaviour of Newton's method combined with a two-grid solver is analysed for calculating the solution regarding $\Lambda = 0.2$ on mesh level 3. Thereby, the coarse-grid solver, that is a direct solver, is applied already on a quite fine mesh, which is why the solution scheme in some sense represents a first extension of an overall direct linear solver towards actual Newton-*multigrid* techniques. Probably, considering a larger sequence of mesh levels within the linear solver will not lead to a successful scheme, in case already the two-grid solver does not converge. However, the solution corresponding to $\Lambda = 0.1$ on the same mesh level is used as initial solution in terms of the outer nonlinear solver, which is taken to be undamped throughout the following study.

Λ	n_s	ω_s	ω_p	N	L	γ_u	γ_σ	γ_μ
0.2	2	0.125	0.125	6	44	0.1	0.01	–
	4	0.125	0.125	✗	✗			
	16	0.125	0.125	✗	✗			
	2	0.25	0.5	✗	✗			
	2	0.5	1.0	✗	✗			
	4	0.5	1.0	✗	✗			
	16	0.5	1.0	✗	✗			
	2	0.125	0.125	6	54	0.1	0.001	–
	16	0.125	0.125	✗	✗			
	2	0.5	0.5	✗	✗			
	2	0.125	0.125	✗	✗	0.1	0.0	–
				✗	✗	0.01	0.0	–
				✗	✗	0.01	0.01	–
				✗	✗	1.0	0.01	–
				✗	✗	10.0	0.1	–
	2	0.25	0.25	✗	✗	100.0	0.1	–
	2	0.5	0.5	✗	✗			
	4	0.125	0.125	✗	✗			
	4	0.5	0.5	✗	✗			
	16	0.25	0.25	✗	✗			
	32	0.5	1.0	✗	✗	1.0	0.01	–
				✗	✗	10.0	0.01	–
				✗	✗	100.0	0.01	–

Table 4.7: Behaviour of the two-grid solver ($l_{\min} = 2, l_{\max} = 3$) for the original problem formulation including the UCM model at $\Lambda = 0.2$ for several parameters of the linear solver and EOFEM parameters

As indicated above, a corresponding solution of the original problem formulation from Eq. (2.16) can hardly be computed due to the failure of the two-grid solver. This is illustrated by the results given in Tab. 4.7, where ✗ denotes a diverging (linear) solver typically obtained within the first nonlinear step. In few cases, the residual can be successfully reduced by one digit within the first Newton iteration(s), but divergence of the linear two-grid solver keeps occurring during the solution process. Obviously, a convergent Newton scheme including a two-grid solver is obtained only for a very small number of smoothing steps and a significant damping in the smoothing as well as prolongation operators. Even slightly varying the "successful" parameters causes the linear solver to diverge. Thereby, a moderate or even large smoother damping parameter $\omega_s \in]0, 1]$ leads to divergence independently of the number of smoothing steps as well as the amount of EOFEM stabilisation. Moreover, the chosen EOFEM parameters should not be too high and naturally not too small either. Surprisingly, even considering a large amount of EOFEM stabilisation does not lead to a converging solution scheme, where the stabilising operator should be the dominant part in the Stokes subproblem naturally resulting in a too diffusive (and thus unphysical) solution. Overall, the two-grid solver clearly behaves non-robust in the non-solvent case of the original problem formulation, at least regarding the UCM model. Besides, the only converging configurations listed in Tab. 4.7 consist of a very large average number of linear steps, which does not make this solution approach very attractive. In addition, no two-grid setting is found regarding a successful computation of the solution corresponding to $\Lambda = 0.3$, where the initial solution is chosen according to $\Lambda = 0.2$.

Λ	n_s	ω_s	ω_p	N	L	γ_u	γ_σ	γ_μ
0.2	2	0.125	0.125	6	83	0.1	0.01	0.0
	2	0.25	0.25	6	58			
	2	0.5	1.0	✗	✗			
	4	0.25	0.25	✗	✗			
	8	0.5	1.0	✗	✗			
	16	0.25	0.25	6	12			
	16	0.5	1.0	✗	✗			
	32	0.5	1.0	6	2			
	32	0.25	1.0	5	5			
	64	0.25	1.0	5	2			
	64	0.5	1.0	✗	✗			
	32	0.5	1.0	✗	✗	0.05	0.01	0.0
	32	0.25	1.0	✗	✗			
	32	0.125	0.125	6	10			
	16	0.125	0.125	6	13			
	64	0.125	0.125	7	8			
	16	0.25	0.5	✗	✗			
	16	0.125	0.25	✗	✗			
	32	0.25	0.25	✗	✗			
	32	0.5	1.0	5	2	0.05	0.01	0.001

Table 4.8: Behaviour of the two-grid solver ($l_{\min} = 2, l_{\max} = 3$) for the Tensor Stokes problem including the UCM model at $\Lambda = 0.2$ for several parameters of the linear solver and EOFEM parameters

Next, applying multigrid techniques is analysed for solving the linear systems arising within Newton's method regarding the four-field formulation of the Tensor Stokes problem from Eq. (2.51). Therefore, a similar setting as above for the original problem formulation is considered, that is mesh level 2 is regarded as the coarsest and level 3 as the finest mesh within a two-grid solver. Again, the computation of $\Lambda = 0.2$ is investigated in the first place, which is used for fine-tuning the two-grid solver concerning the numerical calculation of successively increasing relaxation times in a second step. The resulting behaviour of the two-grid solver given in Tab. 4.8 is improved compared to the original approach, since very good convergence rates – that is the combination of nonlinear and average linear steps N/L – are obtained at least for certain solver settings. In detail, only few nonlinear steps are needed and at the same time, a number of less than ten average linear iterations can be reached. Furthermore, the two-grid solver seems to be more robust compared to the original problem formulation, but still the linear solver diverges in case of moderately varying the "successful" parameters. As before, the applied EOFEM stabilisation including $\gamma_u = 0.1$ and $\gamma_\sigma = 0.01$ can not be changed significantly to maintain the good convergence behaviour of the solver. Accordingly, slightly reducing the stabilisation with respect to the velocity field obviously increases the complexity of solving the linear problem via multigrid techniques, which are still applicable successfully though. But, the convergence rates obtained for the larger amount of EOFEM stabilisation with respect to $\mathbf{u}$ can be recovered in case of $\gamma_u = 0.05$ by stabilising the Diffusion Tensor variable as well.

Λ	n_s	ω_s	ω_p	N	L	γ_u	γ_σ	γ_μ
0.2	32	0.5	1.0	6	2	0.1	0.01	0.0
0.3	32	0.5	1.0	5	3	0.1	0.01	0.0
0.4	32	0.5	1.0	5	3	0.1	0.01	0.0
0.5	32	0.5	1.0	5	3	0.1	0.01	0.0
0.6	32	0.5	1.0	5	3	0.1	0.01	0.0
0.7	32	0.5	1.0	5	3	0.1	0.01	0.0
0.8	32	0.5	1.0	5	3	0.1	0.01	0.0
0.9	32	0.5	1.0	5	3	0.1	0.01	0.0
1.0	32	0.5	1.0	5	3	0.1	0.01	0.0
1.25	32	0.5	1.0	5	4	0.1	0.01	0.0
1.5	32	0.5	1.0	6	5	0.1	0.01	0.0
1.75	32	0.5	1.0	5	7	0.1	0.01	0.0
2.0	32	0.5	1.0	✗	✗	0.1	0.01	0.0
	64	0.5	1.0	✗	✗	0.1	0.01	0.0
	32	0.25	0.5	✗	✗	0.1	0.01	0.0
	16	0.25	1.0	✗	✗	0.1	0.01	0.0
	4	0.25	1.0	✗	✗	0.1	0.01	0.0
	64	0.25	1.0	✗	✗	0.1	0.01	0.0
	64	0.125	0.25	✗	✗	0.1	0.01	0.0
	32	0.5	1.0	6	117	0.1	0.01	0.00005
	32	0.5	1.0	6	77	0.1	0.01	0.0001
	32	0.5	1.0	6	57	0.1	0.01	0.000225
	32	0.5	1.0	✗	✗	0.1	0.01	0.00025
	32	0.5	1.0	✗	✗	0.1	0.01	0.0005

Table 4.9: Behaviour of the two-grid solver ($l_{\min} = 2, l_{\max} = 3$) for the Tensor Stokes problem including the UCM model at increasing Λ for several parameters of the linear solver and EOFEM parameters

Also in case of successively increasing the relaxation time Λ, the two-grid solver can be applied successfully in terms of solving the four-field formulation of the Tensor Stokes problem from Eq. (2.51). In detail, very stable convergence rates are obtained even at higher relaxation times when choosing a number of 32 smoothing steps together with a smoother damping parameter of $\omega_s = 0.5$ (see Tab. 4.9). Thereby, the solution corresponding to the lower relaxation time is used as initial solution within the Newton scheme for computing the current Λ. In doing so, the number of nonlinear steps is more or less unaffected when increasing Λ, though the number of average linear iterations increases for the higher relaxation times considered here. It turns out, that $\Lambda = 2.0$ is computable for this setting only when introducing EOFEM stabilisation with respect to the Diffusion Tensor. But even in this case, the convergence rates are not of the same quality as in case of lower relaxation times and can not be improved for varying the EOFEM parameters or the parameters specifying the linear solver. Moreover, the complexity of applying multigrid techniques increases somehow suddenly. Nevertheless, introducing the Tensor Diffusion approach indeed improves the numerical framework of simulating non-solvent viscoelastic fluid flows concerning the application of two-grid solvers within Newton schemes. In detail, no solution of the original approach at $\Lambda > 0.2$ can be obtained for this setting, while the Tensor Diffusion approach allows successful computations including higher relaxation times.

Λ	n_s	ω_s	ω_p	$l_{\min}$	$l_{\max}$	N	L	γ_u	γ_σ	γ_μ
0.2	32	0.5	1.0	1	3	6	6	0.1	0.01	0.0
0.3	32	0.5	1.0	1	3	5	8	0.1	0.01	0.0
0.4	32	0.5	1.0	1	3	5	8	0.1	0.01	0.0
0.5	32	0.5	1.0	1	3	5	9	0.1	0.01	0.0
0.2	32	0.5	1.0	3	4	✗	✗	0.1	0.01	0.0
	32	0.25	0.5	3	4	✗	✗			
	32	0.125	0.25	3	4	5	91			
	32	0.25	0.25	3	4	✗	✗			
	8	0.125	0.125	3	4	✗	✗			
	16	0.125	0.125	3	4	✗	✗			
0.2	32	0.5	1.0	2	4	✗	✗	0.1	0.01	0.0
	32	0.125	0.25	2	4	✗	✗			
	64	0.25	0.5	2	4	✗	✗			
	2	0.125	0.125	2	4	6	135			
	8	0.125	0.125	2	4	5	75			
	16	0.125	0.125	2	4	6	67			
	32	0.125	0.125	2	4	✗	✗			
	8	0.25	0.25	2	4	5	26			

Table 4.10: Behaviour of several configurations of multigrid solvers for the Tensor Stokes problem including the UCM model at increasing Λ for different EOFEM parameters

In Tab. 4.10, the convergence behaviour of actual *multigrd* solvers regarding varying mesh levels is presented, which results from solving the Tensor Stokes problem at moderate relaxation times. As before, the solution corresponding to the next lower Λ is used as initial solution within Newton's method for successively increasing relaxation times, while the solution regarding $\Lambda = 0.1$ is applied for computing $\Lambda = 0.2$ in this context. In case of the three-grid solver consisting of $l_{\min} = 1$ and $l_{\max} = 3$, again a quite stable behaviour of the solution scheme is obtained including acceptable convergence rates, although the average number of linear iterations is slightly higher than in case of the corresponding two-grid solver (see Tab. 4.9). However, when considering a two-grid solver with respect to the mesh levels 3 and 4, the performance of the linear solver is downgraded significantly. In detail, the chosen damping parameters of the smoother as well as prolongation need to be descreased, which causes a large number of average linear steps. One might have the idea, that considering $l_{\min} = 2$ might improve the behaviour of the multigrid solver, since the jumps of the Diffusion Tensor are less pronounced on level 2 compared to level 3 (see Sec. 4.2.1). Indeed, considering a lower coarse mesh level seems to cause a slightly better solver behaviour, as the number of average linear iterations per nonlinear step can be reduced. In addition, the resulting solver seems to behave more robust, since more convergent configurations of the linear solver can be found. Nevertheless, the performance is still inferior to configurations consisting of $l_{\max} = 3$, as the convergent settings involve a significant smoother and prolongation damping, which in turn leads to a high number of average linear iterations. Possibly, the inferior behaviour of the multigrid solver concerning a higher maximum mesh level is obtained due to the larger $\mathbf{M}$-peaks compared to level 3. Thus, applying EOFEM stabilisation with respect to the Diffusion Tensor could again improve the solver behaviour, since the increasing jumps in $\mathbf{M}$ are damped.

Finally, the applicability of the two-grid solver is briefly discussed regarding the nonlinear exponential PTT model. Therefore, the original problem formulation as well as the four-field formulation of the Tensor Stokes problem from Eqs. (2.16) and (2.51) are considered, where the model function is chosen according to Eq. (2.6b). In Tab. 4.11, the convergence behaviour of the Newton scheme together with a two-grid solver is presented for solving the original problem formulation regarding moderate relaxation times.

Λ	n_s	ω_s	ω_p	N	L	γ_u	γ_σ	γ_μ
0.2	64	0.5	1.0	5	2	0.00125	0.0	–
0.3	64	0.5	1.0	5	3	0.00125	0.0	–
0.4	64	0.5	1.0	5	3	0.0025	0.0	–
0.5	64	0.5	1.0	5	3	0.01	0.0	–
0.6	64	0.125	0.25	6	18	0.1	0.0	–
0.7				✗	✗			

Table 4.11: Behaviour of the two-grid solver ($l_{\min} = 2, l_{\max} = 3$) for the original problem formulation including the exponential PTT model at $\kappa = 0.1$, $\beta = 0.0$ for increasing Λ

In contrast to the UCM model, stable and good convergence rates are obtained even for successively increasing Λ, while the amount of EOFEM stabilisation needs to be slightly increased, too. At the same time, only few nonlinear iterations are needed to solve the discrete systems and the involved average number of linear steps stays small. However, the linear solver experiences certain difficulties for $\Lambda > 0.5$, since small damping parameters regarding both, smoothing as well as prolongation, need to be applied for maintaining a convergent solver at $\Lambda = 0.6$. Concerning $\Lambda = 0.7$, no convergent Newton-multigrid scheme can be configured, although various two-grid and EOFEM settings are tried similar to the above investigations. Nevertheless, especially the linear solver shows an improved behaviour compared to the UCM model, where no two-grid setting is found for successfully performing simulations regarding $\Lambda > 0.2$ in case of the original problem formulation.

Possibly, the (exponential) PTT model consists of similar stabilising properties as the Giesekus model, where the additional quadratic stress contribution in the constitutive law damps the growth of the stress variable (see Sec. 3.1 or Ref. [6]).

Λ	n_s	ω_s	ω_p	N	L	γ_u	γ_σ	γ_μ
0.2	64	0.5	1.0	4	1	0.00125	0.0	0.0005
0.3	64	0.5	1.0	4	1			
0.4	64	0.5	1.0	4	1			
0.5	64	0.5	1.0	6	1			
0.6	64	0.25	1.0	5	4			
0.7				✗	✗			

Table 4.12: Behaviour of the two-grid solver ($l_{\min} = 2, l_{\max} = 3$) for the Tensor Stokes problem including the exponential PTT model at $\kappa = 0.1$, $\beta = 0.0$ for increasing Λ

As mentioned in Sec. 4.2.1, EOFEM stabilisation with respect to the Diffusion Tensor has to be taken into account for successfully solving the four-field formulation of the Tensor Stokes problem including the exponential PTT model. In doing so, the Newton scheme combined with a two-grid solver shows an even better behaviour than in case of the original approach, which is realised from Tab. 4.12. It turns out, that stabilisation as well as multigrid settings can be kept constant for obtaining nearly optimal convergence rates up to $\Lambda = 0.5$, where only one linear step needs to be performed inside every nonlinear iteration. In addition, a solution regarding $\Lambda = 0.6$ can be calculated only by reducing the smoother damping by a factor of two, while the prolongation damping as well as the number of smoothing steps and EOFEM parameters stay unchanged. Naturally, the average number of linear iterations increases, although not that drastically as in case of the original approach, where the damping factors need to be further reduced (see Tab. 4.11). However, $\Lambda = 0.7$ is neither computable when solving the linear subproblems by means of a two-grid solver in terms of the Tensor Diffusion approach, even in case of varying the EOFEM stabilisation or multigrid parameters.

In summary, the results presented above demonstrate the (potentially) improved behaviour of the Newton-multigrid solver in case of applying the Tensor Diffusion approach for both, linear as well as nonlinear differential constitutive equations. Concerning the original problem formulation, a non-robust behaviour even of two-grid techniques is observed for solving the UCM model, which further allow the successful computation of only small relaxation times. This is improved significantly by considering the Tensor Stokes problem formulation, where numerical solutions for successively increasing relaxation times can be obtained including quite stable convergence rates. Moreover, actual multigrid techniques can be successfully applied within Newton's method as well. However, two-grid solvers might also show a robust behaviour in terms of the original approach, for example regarding the non-solvent exponential PTT model. But still, an at least slight improvement is obtained again by means of the Tensor Diffusion approach. Naturally, the specific multigrid configuration may be exploited and fine-tuned for each considered relaxation time, mesh level and material model. In addition, applying such solution techniques needs to be evaluated for other flow configurations like the Flow around cylinder benchmark as well. Nevertheless, a general improvement of the Newton-multigrid scheme is required also in terms of the Tensor Diffusion approach. In fact, a overall direct linear solver still leads to a more stable solution scheme, especially concerning higher mesh levels and relaxation times, where the latter becomes clear in the following section.

4.2.3 A note on the High Weissenberg Number Problem

Concerning numerical simulations of viscoelastic fluid flows, a serious issue is provided by means of the High Weissenberg Number Problem (HWNP, [36, 37, 46]), which typically limits successful computations to the range of low or moderate Weissenberg numbers. But especially in the polymer industry, viscoelastic materials consisting of a relaxation time spectrum reaching over several decades of seconds are processed. Thus, corresponding numerical simulations including high relaxation times and hence high Weissenberg numbers are of intensified interest. In this regard, the Tensor Diffusion approach is applied in the following to evaluate the solvability of resulting viscoelastic flow problems regarding higher relaxation times compared to the original problem formulation. Therefore, the flow of a viscoelastic fluid, which consists of a vanishing solvent viscosity, in the rounded contraction depicted in Fig. 4.24 is considered for the same material parameters as above. Hence, again a Weissenberg number of $\mathrm{We} = \Lambda\frac{U_{\mathrm{mean}}}{2} = 0.1\bar{3}\Lambda$ is obtained in the downstream channel for a given relaxation time Λ.

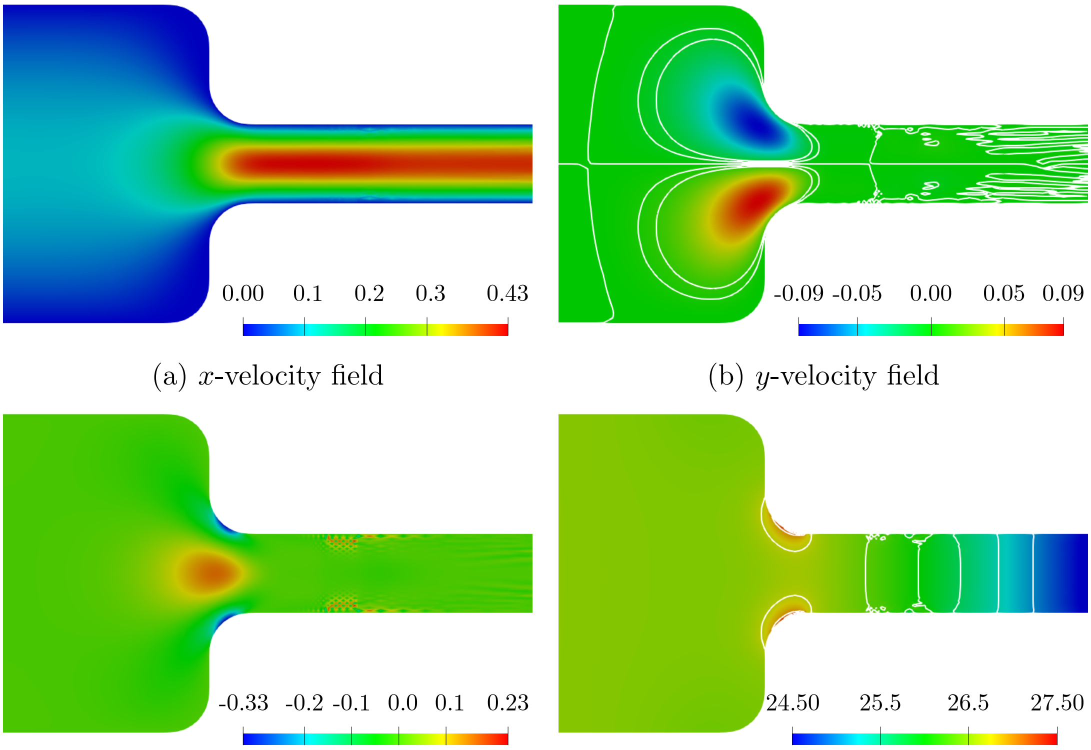
For determining the maximum Λ computable by means of the original as well as Tensor Diffusion approach, this quantity is increased successively starting from a quite small value, until no solution can be obtained anymore. Thus, the stability of the nonlinear solution scheme is of interest instead of the accuracy or stability of the corresponding numerical solution, which is already analysed in Sec. 4.1.2. As discussed in the previous section, Newton-multigrid solvers do not behave reliably robust even in case of the Tensor Diffusion approach. Hence, an (undamped) Newton scheme as described in Sec. 2.2.3 is combined below with a direct solver for treating the arising discrete nonlinear systems, similar to the computations in terms of the Flow around cylinder benchmark. It turns out, that only few nonlinear iterations are needed for computing the solution, that is obtaining residuals smaller than 10^{-8}, regarding the next higher relaxation time, which is usually increased in additive 0.1-steps. But, the nonlinear solver suddenly stagnates or diverges even in the damped case when solving for $\Lambda_{\mathrm{max}} + 0.1$, although only two or three nonlinear steps are needed for all lower relaxation times. Once a diverging or stagnating nonlinear solver is obtained, the applied amount of stabilisation is adjusted with the aim to successfully compute even higher relaxation times. In case no such setting can be found, the according configuration in the following tables is denoted by ✗, that is no solution for a larger relaxation time than the corresponding Λ_{init} can be computed. At the same time, the actual limits of the specific configuration, that is the problem formulation together with the chosen material model as well as stabilisation parameters, are not exhausted. In fact, the computations are usually stopped at a certain parameter configuration offering a suitable comparison of both approaches, since the main goal is to detect, whether (potentially) higher relaxation times can be reached by means of the Tensor Diffusion approach.

In a first step, again the UCM model is considered in the original as well as

Tensor Stokes formulation of the differential stationary viscoelastic flow model, that is Eq. (2.4) is inserted into the nonlinear problem from Eq. (2.16) (or (2.50)) as well as (2.51), respectively, at $\eta_s = 0$. According to the computations in terms of the Flow around cylinder benchmark discussed in Sec. 4.1.2, the corresponding results presented in the following are denoted by $\mathbf{\Pi}_\sigma$ and $\mathbf{\Pi}_\mu$, although the total stress tensors do not play an important role in this context. However, the maximum computable relaxation times are given in Tab. 4.13 for a specific amount of EOFEM stabilistation with respect to the velocity and stress fields as well as the Diffusion Tensor.

	γ_u	γ_σ	γ_μ	Λ_{init}	$\Lambda_{\max}$
$\mathbf{\Pi}_\sigma$	0.01	0.001	0.0	0.1	14.5
	“>0.01”	0.001	0.0	14.5	✗
	0.1	0.01	0.0	0.1	9.9
	0.5	0.01	0.0	9.9	21.3
$\mathbf{\Pi}_\mu$	0.01	0.001	0.0	0.1	✗
	0.01	0.001	0.0005	0.1	11.5
	0.01	0.001	0.00075	11.5	14.1
	0.01	0.001	0.001	14.1	16.9
	0.1	0.01	0.0	0.1	12.0
	0.1	0.01	0.0005	12.0	20.8

Table 4.13: Maximum computable relaxation times in terms of a gradual increase for the UCM model on level 3 when adjusting the EOFEM parameters



(a) x-velocity field

(b) y-velocity field

(c) D_{11} field

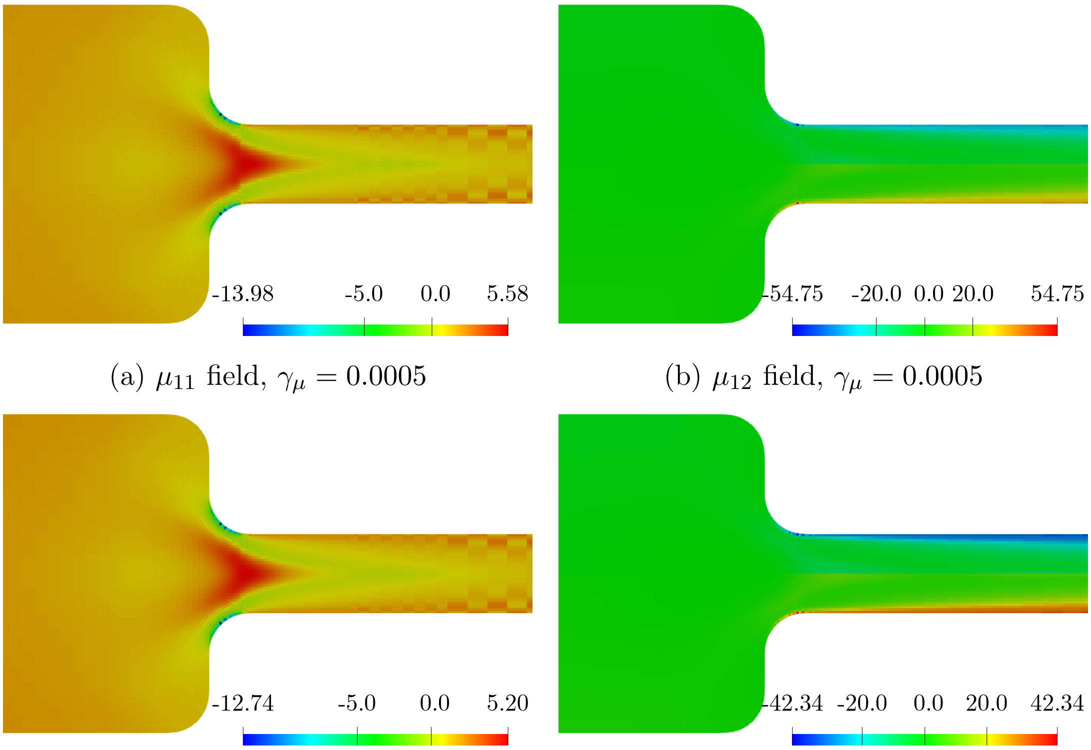
(d) p field

Figure 4.32: Flow quantities resulting from the original problem formulation including the UCM model at $\Lambda_{\max} = 14.5$, $\eta_0 = 1.0$ on level 3 for EOFEM parameters $\gamma_u = 0.01$, $\gamma_\sigma = 0.001$

Concerning the original problem formulation, an EOFEM configuration including fixed stabilisation parameters of $\gamma_u = 0.01$ and $\gamma_\sigma = 0.001$ allows for computing a maximum relaxation time of 14.5, while higher Λ can not be reached even by increasing the amount of stabilisation. In contrast, larger EOFEM parameters already applied for $\Lambda_{\text{init}} = 0.1$ make higher Λ computable for increasing γ_u and γ_σ, although the intermediate $\Lambda_{\max}$ is smaller. The solution of the original approach corresponding to the lower amount of EOFEM stabilisation at $\Lambda_{\max} = 14.5$ is depicted in Fig. 4.32. The visible significant perturbation in $\mathbf{D}$ and p (and also $\mathbf{\Sigma}$) as well as slight oscillations in u and v indicate the non-convergent behaviour of the nonlinear solver regarding higher relaxation times: As realised by means of the discrete problem formulation in Eq. (2.29) or Eq. (2.55), the (Jacobian) matrix of the nonlinear system is built based on such polluted solutions according to Eq. (2.30), which naturally sooner or later might affect the numerical solver. Probably, the solution at $\Lambda_{\max}$ for $\gamma_u = 0.01$ and $\gamma_\sigma = 0.001$ is already so distorded, that the nonlinear solver is not able to converge due to the shape of the resulting Jacobian – even in case of an intensified stabilisation. In fact, similar perturbations are observed for all solutions corresponding to $\Lambda_{\max}$, also regarding the Tensor Diffusion approach. But, these effects build up slowly for increasing Λ and do not appear all of a sudden, which is why the nonlinear solver may be converging for lower relaxation times despite the actually perturbed shape of the flow quantities.

When considering the Tensor Stokes problem formulation, a stabilising effect of the "artificial" tensor-valued viscosity or diffusion is observed compared to the original approach. Regarding the EOFEM parameters $\gamma_u = 0.1$, $\gamma_\sigma = 0.01$ and $\gamma_\mu = 0$, higher Λ can be reached by considering the Tensor Diffusion approach, which consequently seems to stabilise the numerical problem by itself. Moreover, taking into account a small amount of stabilisation with respect to $\mathbf{M}$ allows for computing a similar maximum relaxation time as in case of quite extensively increasing the stabilisation with respect to the velocity field to $\gamma_u = 0.5$ in terms of the original approach. Thus, the numerical approach can indeed be interpreted as improved: By considering the Tensor Diffusion approach, solutions regarding similar maximum Weissenberg numbers can be calculated successfully using less stabilisation compared to the original approach. In addition, higher Λ are computable for increasing the stabilisation with respect to $\mathbf{M}$ based on the lower initial choice of $\gamma_u = 0.01$ and $\gamma_\sigma = 0.001$, which is not established by means of an increasing amount of stabilisation in case of the original problem formulation.

In fact, the velocity, stress and pressure fields resulting from the Tensor Stokes problem at $\Lambda_{\max}$ show similar oscillations as presented above. Thus, only the shape of the Diffusion Tensor is analysed in the following, again concerning EOFEM stabilisation. Thereby, the stabilising effect of $\gamma_\mu > 0$ with respect to the Diffusion Tensor in case of $\gamma_u = 0.1$ and $\gamma_\sigma = 0.01$ as well as a maximum relaxation time of $\Lambda = 12.0$ is similar to the results presented in Sec. 4.2.1 or Fig. 4.30. In contrast, the shape of $\mathbf{M}$ differs from these results in case of the smaller parameters $\gamma_u = 0.01$ as well as $\gamma_\sigma = 0.001$, which is observed by means of the components of $\mathbf{M}$ at $\Lambda = 11.5$ depicted in Fig. 4.33. Obviously, the resulting Diffusion Tensor consists of significant perturbations in a broader section of the computational domain, which is why the EOFEM stabilisation is not mainly addressing jumps of the Diffusion Tensor along the symmetry axis, but also at the corners of the contraction and in the downstream channel. Although the oscillations in the Diffusion Tensor are damped only moderately, the maximum relaxation time can be further increased for a slightly larger amount of EOFEM stabilisation with respect to $\mathbf{M}$. In detail, the nonlinear solver recovers a converging behaviour for setting $\gamma_\mu = 0.00075$ instead of $\gamma_\mu = 0.0005$, where $\mathbf{M}$ still shows significant oscillations. Note, that similar observations are made for increasing the EOFEM parameter with respect to the Diffusion Tensor at $\Lambda = 14.1$ (see Tab. 4.13).



(a) μ_{11} field, $\gamma_\mu = 0.0005$

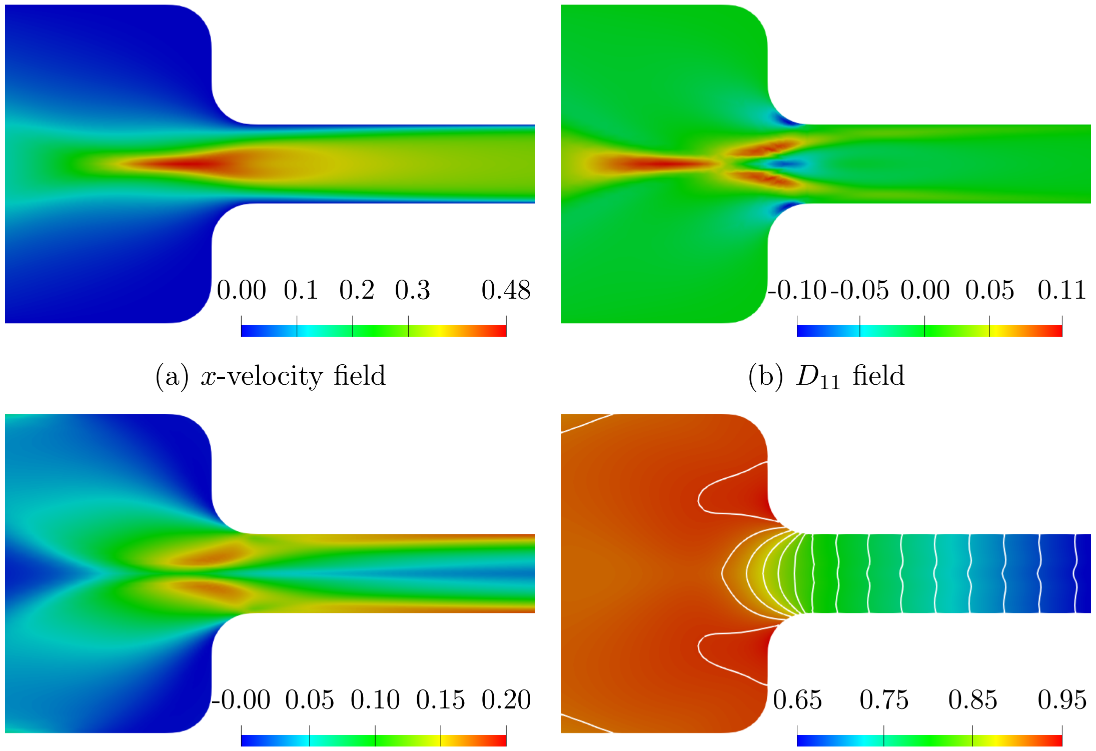
(b) μ_{12} field, $\gamma_\mu = 0.0005$

(c) μ_{11} field, $\gamma_\mu = 0.00075$

(d) μ_{12} field, $\gamma_\mu = 0.00075$

Figure 4.33: Diffusion Tensor solution in Q_0 for the UCM model at $\Lambda = 11.5$, $\eta_0 = 1.0$ for EOFEM parameters $\gamma_u = 0.01$, $\gamma_\sigma = 0.001$, $\gamma_\mu = 0.0$, on level 3

In addition to the UCM model, similar numerical experiments are performed regarding the exponential version of the PTT model including the model function from Eq. (2.6b) at a model parameter of $\kappa = 0.1$. Thereby, suitable fully developed flow profiles are set on the in- and outflows egdes of the computational domain, which deviate from the parabolic shape. In case of the original problem formulation, no maximum relaxation time is found and computations are stopped at $\Lambda = 250.0$, where a smaller amount of EOFEM stabilisation is chosen compared to the UCM model. Due to the very large relaxation times computable in this setting, a computational domain consisting of an extended upstream channel with the inflow edge being located at $x = -25$ instead of $x = -8$ is taken into account. By doing so, nonlinear effects arising at the entrance of the contraction do not reach up to the inflow, which would conflict with the assumption of a fully developed flow entering the computational domain. Note, that the relaxation of the flow variables towards the fully developed state at the in- and outflow edges is not visible completely, as the entire up- and downstream channel are not presented in the below results. However, the corresponding solution looks quite smooth in the “interesting” section of the computational domain including the contraction depicted in Fig. 4.34. But, slight oscillations or perturbations are visible in the flow quantities, for example in $\mathbf{D}(\mathbf{u})$ and p. Furthermore, significant deviations compared to the UCM model depicted in Fig. 4.32 are observed regarding the general shape of the flow fields. In fact, $\mathbf{u}$, $\mathbf{D}$ and $\mathbf{\Sigma}$ show distinct phenomena reaching upstream towards the inflow edge of the computional domain. Probably, this is caused by the significant nonlinear material behaviour within the contraction, which is not that pronounced in case of the UCM model (and a lower relaxation time).



(a) x-velocity field

(b) D_{11} field

(c) σ_{11} field

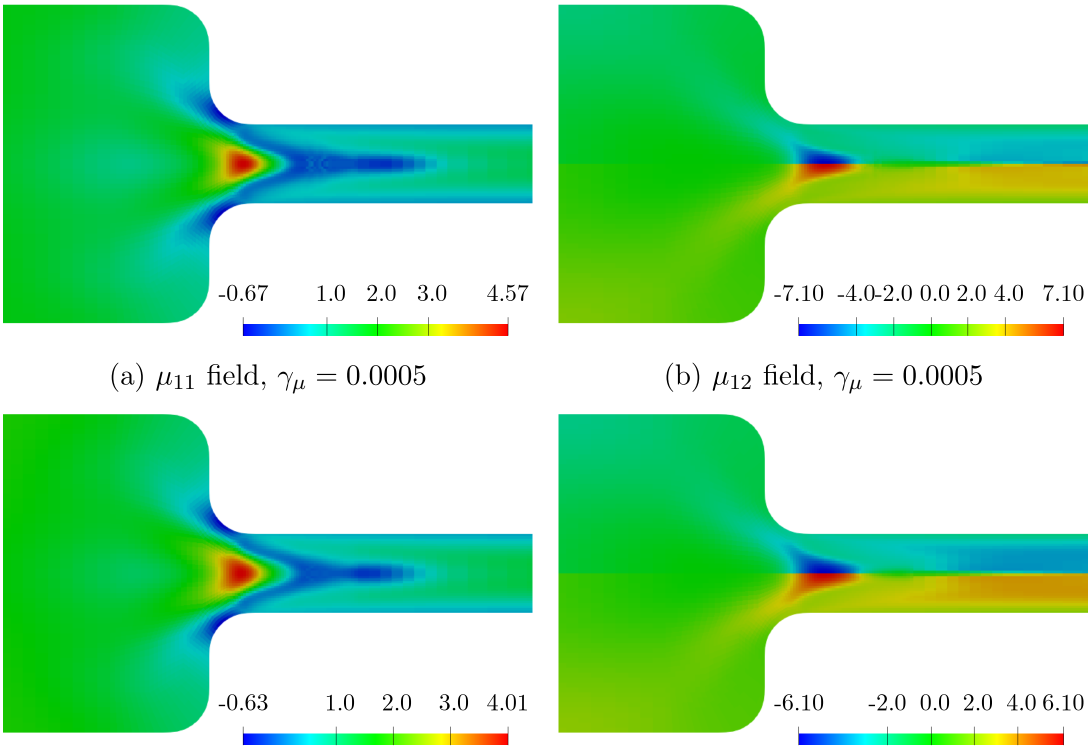
(d) p field

Figure 4.34: Flow quantities resulting from the original problem formulation including the exponential PTT model at $\Lambda = 250.0$, $\kappa = 0.1$, $\eta_0 = 1.0$, $\beta = 0.0$ for EOFEM parameters $\gamma_u = 0.00125$, $\gamma_\sigma = 0.0$, $\gamma_\mu = 0.0$, on level 3

	γ_u	γ_σ	γ_μ	Λ_{init}	Λ_{max}
$\mathbf{\Pi}_\sigma$	0.00125	0.0	0.0	0.1	≥ 250.0
$\mathbf{\Pi}_\mu$	0.00125	0.0	0.0005	0.1	13.8
	0.00125	0.0	0.001	13.8	17.1
	0.00125	0.0	0.00125	17.1	22.4
	0.01	0.0	0.0025	22.4	136.1

Table 4.14: Maximum computable relaxation times in terms of a gradual increase for the exponential PTT model at $\kappa = 0.1$, $\beta = 0.0$ on level 3 when adjusting the EOFEM parameters

Concerning the Tensor Diffusion approach, no solution can be computed regarding the (exponential) PTT model in case of a neglected EOFEM stabilisation with respect to $\mathbf{M}$. Thus, a relatively small parameter of $\gamma_\mu = 0.0005$ similar to Sec. 4.2.1 is chosen with respect to the initial configuration for successively increasing the relaxation time. According to the original problem formulation, Weissenberg numbers significantly larger than for the UCM model can be reached here (see Tab. 4.14), which might be explained by means of a stabilising character of the constitutive equation similar to the Giesekus model – although the material behaviour becomes highly nonlinear. However, such large relaxation times as in case of the original problem formulation can not be obtained straightforward by means of the Tensor Diffusion approach. In detail, the EOFEM parameters need to be adjusted in terms of increasing Λ in contrast to the original approach, where the amount of stabilisation can be kept constant for all considered relaxation times.



(a) μ_{11} field, $\gamma_\mu = 0.0005$

(b) μ_{12} field, $\gamma_\mu = 0.0005$

(c) μ_{11} field, $\gamma_\mu = 0.001$

(d) μ_{12} field, $\gamma_\mu = 0.001$

Figure 4.35: Diffusion Tensor solution in Q_0 for the exponential PTT model at $\Lambda = 13.8$, $\kappa = 0.1$, $\eta_0 = 1.0$, $\beta = 0.0$ for EOFEM parameters $\gamma_u = 0.01$, $\gamma_\sigma = 0.001$ and different choices of γ_μ on level 3

Compared to the UCM model, the oscillations in the Diffusion Tensor are damped in a much more meaningful way in case of the exponential PTT model when increasing the corresponding amount of EOFEM stabilisation. In fact, the slight perturbations visible in $\mathbf{M}$ for $\Lambda = 13.8$ are removed by increasing the EOFEM parameter γ_μ from 0.0005 to 0.001 (see Fig. 4.35), which probably helps the (nonlinear) solver to recover a convergent behaviour. Moreover, the components of $\mathbf{M}$ resulting in case of the UCM model for the same stabilisation parameter of $\gamma_u = 0.01$ (see Fig. 4.33) are way more distorted than in case of the PTT model. In fact, these oscillations are not even damped significantly, while a clear smoothening effect is observed regarding the PTT model. However, the stabilised component μ_{12} (and also μ_{21}) of the Diffusion Tensor does not look smooth downstream along the symmetry axis of the domain, which clearly indicates a required upgrade of the corresponding stabilisation in terms of the PTT model.

Summarising, only a slight improvement regarding the computation of higher relaxation times is achieved by means of the Tensor Diffusion approach for considering the UCM model. In contrast, the original approach allows for calculating solutions corresponding to larger Λ in case of the (exponential) PTT model. Actually, the (potential) stabilising effect of introducing the Diffusion Tensor might be more pronounced in case of the UCM model, since the PTT model seems to be already more stable per se as indicated by the behaviour of the original approach. Possibly, considering a higher-order approximation together with an advanced determination of the Diffusion Tensor may allow for computing higher relaxation times in context of the PTT model on a similar level as the original problem formulation. Overall, no significant progress concerning the High Weissenberg Number Problem is established by means of the Tensor Diffusion approach. Admittedly, this novel approach is not intended to tackle such an issue, which is why no improved numerical scheme can be expected concerning this matter.

4.2.4 Prototypical solution schemes based on the Diffusion Tensor

As outlined by means of the Flow around cylinder benchmark in Sec. 4.1.2, the behaviour of the nonlinear solver is basically of similar quality concerning both, the original as well as Tensor Stokes problem formulation. But, the latter does not provide as accurate solutions as the three-field formulation of the original differential viscoelastic flow model. For keeping the potential benefits of the Tensor Diffusion approach and improving the resulting solution quality, iteration procedures actually minimising the original residual might be taken into account, which at the same time make use of the Diffusion Tensor. Therefore, a prototypical monolithic solution scheme – especially applicable regarding differential constitutive equations – is presented below, followed by a decoupled approach, which could be a suitable candidate in the context of integral material laws. The solver behaviour of both approaches is analysed by means of the flow configuration considered throughout this section, that is the flow in a rounded contraction including the material and model parameters introduced above. Moreover, the material behaviour is characterised by means of the (differential) Oldroyd-B as well as UCM model to analyse the impact of a present solvent contribution on the solver behaviour.

The mixed iteration within a monolithic approach

In a first step, a so-called mixed iteration taking into account the (differential) four-field formulations introduced in Sec. 2.4 is designed. Thereby, the residual of the original viscoelastic model from Eq. (2.50) and the Jacobian resulting from the corresponding Tensor Stokes problem (2.51) are combined in terms of a Newton-like method based on Eq. (2.29). Thus, the solution corresponding to the original problem formulation is obtained, and at the same time (additional) stabilisation or artificial diffusion is provided by means of the Diffusion Tensor, which could lead to a potentially improved solution scheme. Note, that the arising linear systems are again solved by means of a direct linear solver, while multigrid techniques should show a similar behaviour as outlined in Sec. 4.2.2. However, the resulting (undamped) iteration procedure reads

$$\mathrm{x}^{k+1} = \mathrm{x}^k - \left(\mathrm{A}^\mu\left(\mathrm{x}^k\right)\right)^{-1}\mathrm{r}^\sigma\left(\mathrm{x}^k\right) \tag{4.16a}$$

for a given initial solution x^0, where the residual and Jacobian are actually taken from the differing discrete versions of Eqs. (2.50) and (2.51), in detail

$$\mathrm{x} = \begin{pmatrix}\boldsymbol{u}\\ \boldsymbol{\sigma}\\ \boldsymbol{\mu}\\ \boldsymbol{p}\end{pmatrix}, \quad \mathrm{r}^\sigma(\mathrm{x}) = \begin{pmatrix}\mathcal{L}\boldsymbol{u} + \mathcal{J}^u\boldsymbol{u} + \mathcal{C}\boldsymbol{\sigma} + \mathcal{B}\boldsymbol{p}\\ \mathcal{D}\boldsymbol{u} + \mathcal{K}(\boldsymbol{u})\boldsymbol{\sigma} + \mathcal{J}^\sigma\boldsymbol{\sigma}\\ \mathcal{S}\boldsymbol{\sigma} + \mathcal{N}(\boldsymbol{u})\boldsymbol{\mu} + \mathcal{J}^\mu\boldsymbol{\mu}\\ \mathcal{B}^\top\boldsymbol{u}\end{pmatrix} \tag{4.16b}$$

$$\mathrm{A}^\mu(\mathrm{x}) = \begin{pmatrix}\mathcal{L} + \mathcal{T}(\boldsymbol{\mu}) + \mathcal{J}^u & 0 & \mathcal{T}_\mu(\boldsymbol{\mu},\boldsymbol{u}) & \mathcal{B}\\ \mathcal{D} + \mathcal{K}_u(\boldsymbol{u},\boldsymbol{\sigma}) & \mathcal{K}(\boldsymbol{u}) + \mathcal{J}^\sigma & 0 & 0\\ \mathcal{N}_u(\boldsymbol{u},\boldsymbol{\mu}) & \mathcal{S} & \mathcal{N}(\boldsymbol{u}) + \mathcal{J}^\mu & 0\\ \mathcal{B}^\top & 0 & 0 & 0\end{pmatrix} \tag{4.16c}$$

Similar to Eq. (2.29), the subscript of a discrete operator from the Jacobian matrix given in Eq. (4.16c) denotes the derivative with respect to the indicated quantity, which is computed by means of Finite Differencing. Note, that the Jacobian matrix A^μ corresponds to the residual r^σ with respect to the stress, Diffusion Tensor as well as continuity equation, but not the momentum equation. Regarding this, the discrete operator acting on the (discretised) stress tensor is "replaced" by the Tensor Stokes operator, which occurs in the velocity instead of the stress block. Hence, a diffusive operator is provided even in the non-solvent case. Besides, the block referring to the Diffusion Tensor is non-vanishing. In the following, the unstabilised discrete problem is considered, unless stated otherwise.

Naturally, the resulting convergence behaviour will not be able to compete with an actual Newton scheme for solving the pure original or Tensor Stokes problem, which is why second-order convergence is not expected. In fact, the iteration procedure presented in Eq. (4.16) is of pure fixed-point type, which becomes clear by analysing the two problem formulations

$$P_1 = \begin{cases}-2\eta\nabla\cdot\mathbf{D}(\mathbf{u}) + \nabla p & = \mathbf{0}\\ \boldsymbol{\Sigma} - 2\eta\mathbf{D}(\mathbf{u}) & = \mathbf{0}\\ \nabla\cdot\mathbf{u} & = 0\end{cases}, \qquad P_2 = \begin{cases}-\nabla\cdot\boldsymbol{\Sigma} + \nabla p & = \mathbf{0}\\ \boldsymbol{\Sigma} - 2\eta\mathbf{D}(\mathbf{u}) & = \mathbf{0}\\ \nabla\cdot\mathbf{u} & = 0\end{cases} \tag{4.17}$$

of the Stokes equations for a given viscosity η. The set of equations denoted by P_1 represents the original Stokes problem supplemented by an additional equation determining the Newtonian stress tensor in post-processing fashion, while P_2 consists of the actual three-field formulation. Obviously, both problem formulations are equivalent on continuous, but not on discrete level: After discretisation according to the techniques presented in Sec. 2.2.3, the resulting discrete versions of the momentum equations regarding P_1 and P_2 read

$$\mathcal{L}\boldsymbol{u} + \mathcal{B}\boldsymbol{p} = \mathbf{r}_u, \quad \mathcal{C}\tilde{\mathcal{K}}^{-1}\left(\mathbf{r}_\sigma + \tilde{\mathcal{D}}\boldsymbol{u}\right) + \mathcal{B}\boldsymbol{p} = \mathbf{r}_u \tag{4.18}$$

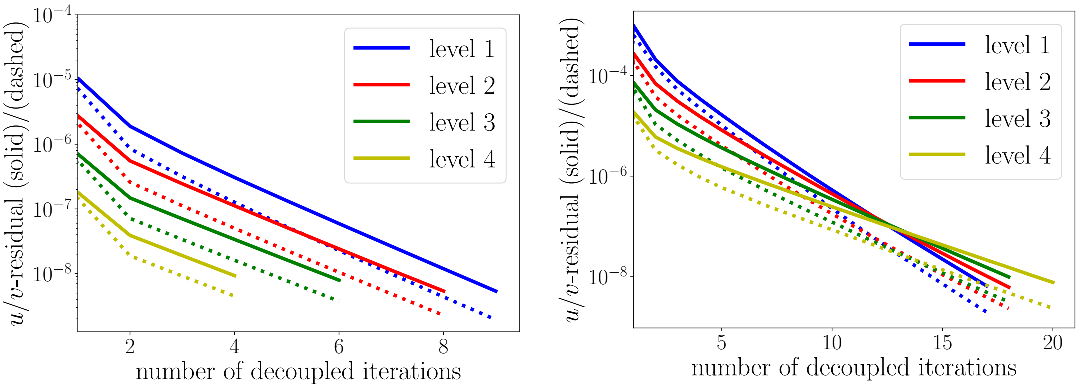
Thereby, $\tilde{\mathcal{K}}$ and $\tilde{\mathcal{D}}$ basically denote suitable simplifications or modifications of the discrete operators introduced in Eq. (2.19) regarding differential constitutive equations. However, the mixed iteration resulting from the two problem formulations given in Eq. (4.17) probably will not recover the convergence behaviour of Newton's method. In contrast to the continuous versions, the discrete problem formulations are not equivalent due to $\mathcal{C}\tilde{\mathcal{K}}^{-1}\tilde{\mathcal{D}} \approx \mathcal{L}$ (in case of neglecting $\mathbf{r}_\sigma$). Thus, only a quasi-Newton scheme or a pure fixed-point iteration would be obtained based on the discrete Jacobian from P_1 and the residual from P_2. Possibly, the convergence speed might be accelerated with increasing mesh level, since the problem formulations are equivalent continuously being the limit of the discrete problems regarding mesh refinement. Thus, the discrete momentum equations from Eq. (4.17) might converge towards each other for infinitely decreasing the mesh size. But at the same time, the structure of the Jacobian matrix resulting from problem P_1 varies compared to the Jacobian from P_2. Hence, the Jacobian matrix will not "match" the residual in terms of a mixed iteration on any mesh level, probably limiting the convergence speed in general to not recover a Newton-like behaviour. Concerning the mixed iteration from Eq. (4.16), this effect might be even more pronounced, since the structure of the Jacobian matrix differs not only regarding the operators acting on the (discrete) velocity or stress fields. Moreover, an additional contribution is present regarding the Diffusion Tensor, which will be non-vanishing even in case of a converged solution or increasing mesh refinement.

In a first step, the mixed iteration proposed above is applied for solving the Oldroyd-B model, where the number of nonlinear iterations of the Newton-like scheme from Eq. (4.16) is given in Tab. 4.15. When computing the solution corresponding to $\Lambda = 0.001$ on all mesh levels, the Newtonian case is used as initial solution, while the solution for the previous relaxation time is chosen for calculating higher Λ. Thus, the iteration numbers decrease from $\Lambda = 0.001$ to 0.01 before increasing in case of higher relaxation times. In detail, the number of nonlinear steps keeps increasing for increasing Λ on all mesh levels due to the problem accordingly becoming more nonlinear.

level \ Λ	0.001	0.01	0.1	1.0	2.5	5.0
1	12	9 (2)	12 (2)	17	21	✗
2	10	8 (2)	11 (2)	18	23	✗
3	7	6 (1)	10 (2)	18	35	✗
4	4	4 (1)	8 (1)	20	✗	

Table 4.15: Number of nonlinear iterations of the mixed iteration (compared to an actual Newton scheme) for the Oldroyd-B model at $\beta = 0.5$ for increasing Λ and mesh levels

As expected, an increased number of nonlinear steps is obtained from the mixed iteration compared to an actual Newton scheme, where the corresponding iteration numbers for certain relaxation times are given in brackets in the same table. Hence, probably not even a quasi-Newton method is established by means of the mixed iteration, which consequently leads to an inferior solution approach. But, it seems like the possible acceleration of the iteration scheme mentioned above is observed in case of small relaxation times, that is $\Lambda \in \{0.001, 0.01, 0.1\}$, where the number of nonlinear steps indeed decreases with increasing mesh refinement. The performance of Newton's method is not reached for the considered mesh levels though. But actually, the convergence speed stays constant for increasing mesh levels, which is realised by means of the residual plots depicted in Fig. 4.36(a). Instead of a faster convergence, the initial residual decreases with mesh refinement causing a decreasing number of iterations, which probably holds in all cases indicated above as "accelerating" iterations. In the same way, the nonlinear solver seems to behave more or less independently of the mesh size at $\Lambda = 1.0$, which is also contradicted by the residual plots from Fig. 4.36(b). It turns out, that the convergence speed actually decreases for increasing refinement, which leads to a rather constant number of iterations due to the decreasing initial residuals. In case of $\Lambda = 2.5$, the iteration numbers increase in case of a converging solution scheme, which is not obtained on level 4. When considering $\Lambda = 5.0$, no solution can be computed on any mesh level for the current configuration. The latter observations may result again from the increasing complexity or nonlinearity of the problem, once too high relaxation times are considered.



(a) $\Lambda = 0.01$

(b) $\Lambda = 1.0$

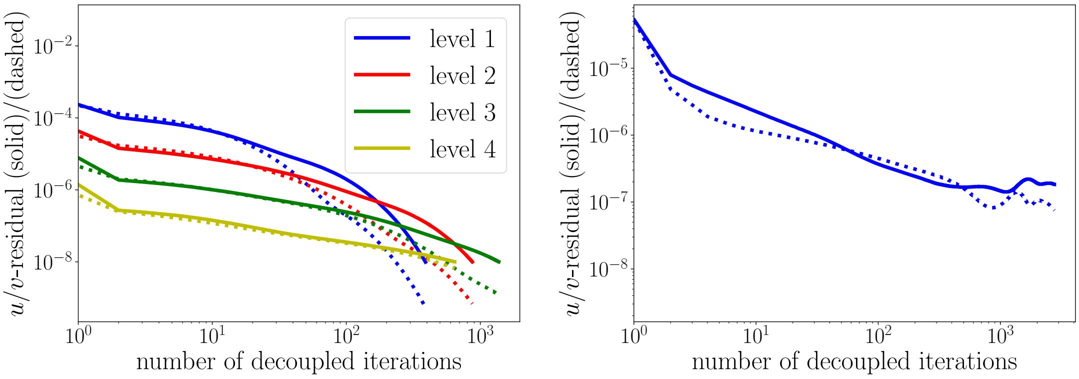
Figure 4.36: Convergence behaviour of the mixed iteration for the Oldroyd-B model at $\beta = 0.5$ for different Λ and mesh levels

Concerning the UCM model, the iteration numbers of the mixed iteration increase drastically compared to the Oldroyd-B model regarding all mesh levels as well as relaxation times (see Tab. 4.16).

Λ level	0.001	0.01	0.1	0.5
1	390	100	227	327
2	873	118	329	1347
3 ($\gamma_u = 0.0$)	1374	90	396	>5000 ✗
3 ($\gamma_u = 0.1$)	11	7	10	52
4	644	38	266	>5000 ✗

Table 4.16: Number of nonlinear iterations of the mixed iteration for the UCM model at increasing Λ and mesh levels

Again, the iteration numbers decrease from $\Lambda = 0.001$ to 0.01 before increasing again due to the varying initial solution: The solution obtained for the Oldroyd-B model on the same level is chosen in case of $\Lambda = 0.001$, while the solutions on each mesh level regarding higher Λ are computed based on the solution for the next lower relaxation time. However, the larger iteration numbers compared the Oldroyd-B model probably result from the vanishing solvent viscosity, although a diffusive – and thus stabilising – operator is supposed to be provided in terms of the Tensor Diffusion approach. In turn, the at least subliminal stabilising effect of the Diffusion Tensor is highlighted when neglecting the corresponding diffusion operator in the "Jacobian" matrix from Eq. (4.16c) in case of $\eta_s > 0$, which leads to a stagnating and thus inferior behaviour of the nonlinear solver. In addition, neglecting the Diffusion Tensor regarding the unstabilised UCM model does not make the mixed iteration applicable at all due to the resulting properties of the "Jacobian".



(a) $\Lambda = 0.001$

(b) $\Lambda = 0.5$ on level 3

Figure 4.37: Convergence behaviour of the mixed iteration for the (unstabilised) UCM model at different Λ and mesh levels

To further illustrate the convergence behaviour of the mixed iteration in terms of the UCM model, a plot of the residuals corresponding to $\Lambda = 0.001$ is given in Fig. 4.37(a) for several mesh levels. It turns out, that the initial residuals again decrease with increasing mesh refinement. In addition, the convergence speed increases during the solution process, which is not observed in case of the Oldroyd-B model. Obviously, this effect occurs at larger iteration numbers, which are not required in case of a present solvent viscosity. At the same time, a nearly linear convergence behaviour is obtained in terms of the first nonlinear steps, similar to the Oldroyd-B model. However, a relatively fast convergence close to the solution is observed especially on lower mesh levels, which is not that pronounced on finer meshes leading to an increasing number of nonlinear steps. In case of larger relaxation times, the complexity of the numerical problem seems to increase drastically, since the residuals obtained from the mixed iteration regarding $\Lambda = 0.5$ on mesh level 3 stagnate or even oscillate (see Fig. 4.37(b)), which is not observed for lower relaxation times. But even on lower mesh levels, the iteration numbers concerning $\Lambda = 0.5$ increase drastically compared to smaller relaxation times. Thus, the behaviour of the mixed iteration is downgraded significantly, particularly in the non-solvent case, but also regarding higher relaxation times, as the stability properties of the numerical problem are weakened (see Chpt. 2). Unfortunately, this lack of stability in the solution scheme can not be balanced by means of introducing the Diffusion Tensor, at least not in the current state of the approach.

Recall, that the Tensor Diffusion approach is intended to at least potentially improve the numerical approach compared to the DEVSS [41], since the Diffusion Tensor is related to the nature of the nonlinear problem – in contrast to a simple scalar artificial viscosity as applied in Eq. (2.23). But, implementing a DEVSS-like approach in terms of the UCM model by setting $\mathbf{M} = 2\eta_p\mathbf{I}$ instead of the algebraic equation (2.51c) of the four-field formulation of the Tensor Stokes problem actually results in an improved solver behaviour. Regarding $\Lambda = 0.001$, the DEVSS-like setting behaves similar to the Tensor Diffusion approach, since $\mathbf{M}$ is expected to be close to $2\eta_p\mathbf{I}$ in case of small relaxation times. In contrast, roughly 20% reduced iteration numbers are obtained on level 3 for computing the solution concerning $\Lambda \in \{0.01, 0.1\}$. Moreover, a significant improvement is achieved by means of the DEVSS for calculating the solution at $\Lambda = 0.5$, which requires less than 3000 nonlinear iterations. In contrast, no solution is obtained by means of the Tensor Diffusion approach in this case, as the nonlinear solver stagnates or even diverges (see Fig. 4.37(b)). Thus, the Tensor Diffusion approach does not give an improved numerical scheme compared to the DEVSS. But, neither does the DEVSS provide an overall satisfying numerical framework, since still very high iteration numbers are recorded in the non-solvent case. Instead, inserting EOFEM stabilisation into the system dramatically improves the solver behaviour in terms of the Tensor Diffusion approach with respect to the UCM model, which is realised from Tab. 4.16 as well as Fig. 4.38 at the example of level 3.

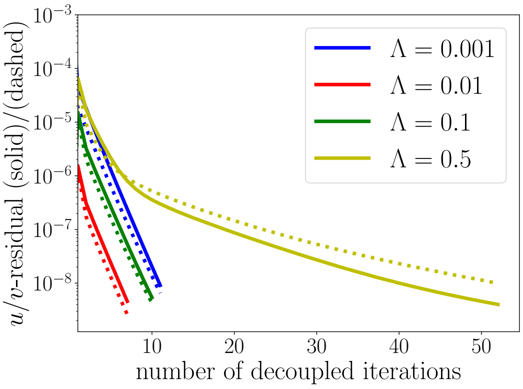


Figure 4.38: Convergence behaviour of the mixed iteration for the UCM model at different Λ and EOFEM parameters $\gamma_u = 0.1$, $\gamma_\sigma = 0.0$, $\gamma_\mu = 0.0$ on level 3

It turns out, that a choice of the EOFEM parameter regarding the velocity field of $\gamma_u = 0.1$ leads to a solver behaviour similar to the Oldryd-B model for $\Lambda \leq 0.1$ (see Tab. 4.15). But, the number of nonlinear steps increases stronger for higher relaxation times in case of the UCM model compared to $\beta > 0$. However, the Tensor Diffusion approach is superior to the DEVSS in case of applying EOFEM stabilisation with respect to $\mathbf{u}$, since twice as much nonlinear iterations are needed for convergence when setting $\mathbf{M} = 2\eta_p\mathbf{I}$. Thus, the Tensor Diffusion approach provides a reasonable solution scheme by means of the mixed iteration proposed above at most in case of a present solvent contribution or additional stabilisation as well as small relaxation times. Concerning more challenging configurations, the iteration procedure does not behave really promising overall.

The decoupled block fixed-point iteration

In this work, mainly monolithic approaches for solving various versions of discrete differential viscoelastic flow models are discussed. As pointed out in Chpt. 2, such a coupled scheme is not practical in the context of integral constitutive equations. Instead, operator splitting techniques need to be applied, where evaluating the stress integral as well as evolving the Finger tensor(s) is decoupled from the Stokes part of the flow model. But, no meaningful iteration procedure is obtained accordingly in case of a vanishing solvent viscosity, since the diffusive operator regarding the velocity field is removed from the momentum equation. Besides well-known approaches like the (D)EVSS [40, 41], a novel possibility for resolving this issue is provided by means of the Tensor Diffusion approach, which might offer a reasonable realisation of decoupled solution approaches in similar way. In the following, corresponding preliminary studies are performed based on differential constitutive equations. But naturally, segregated iteration procedures are intended to be applied mainly to integral viscoelastic material laws, as the monolothic approach described in Sec. 2.2.3 or Ref. [47, 51] offers significant advantages compared to decoupled schemes with respect to differential flow models.

Similar to the mixed iteration discussed above, the residual of the original problem formulation is intended to be minimised within the decoupled solution approach proposed below, again taking into account the Diffusion Tensor in a reasonable way. Therefore, the (symmetrised) stress decomposition from Eq. (2.48) based on the solution from the previous step is inserted into the right-hand side of the momentum equation from the Tensor Stokes problem (2.51). By doing so, the following sequence of *linear* problems can be solved even in the non-solvent case $\eta_s = 0$, which does not provide a corresponding well-posed problem based on the original problem formulation. In a first step, an updated solution $(\mathbf{u}^{n+1}, p^{n+1})$ regarding the velocity and pressure fields is calculated from

$$\begin{aligned} -2\eta_s \nabla \cdot \mathbf{D}\left(\mathbf{u}^{n+1}\right) + \nabla p^{n+1} \dots & \\ -\frac{1}{2}\nabla \cdot \left(\mathbf{N} \cdot \mathbf{D}\left(\mathbf{u}^{n+1}\right) + \mathbf{D}\left(\mathbf{u}^{n+1}\right) \cdot \mathbf{N}^\top\right) &= \dots \\ \dots \nabla \cdot \boldsymbol{\Sigma}^n - \frac{1}{2}\nabla \cdot \left(\mathbf{N} \cdot \mathbf{D}\left(\mathbf{u}^{n}\right) + \mathbf{D}\left(\mathbf{u}^{n}\right) \cdot \mathbf{N}^\top\right) & \qquad (4.19a) \\ \nabla \cdot \mathbf{u}^{n+1} &= \mathbf{0} \qquad (4.19b) \end{aligned}$$

for a given velocity as well as stress solution $(\mathbf{u}^n, \boldsymbol{\Sigma}^n)$ and some Diffusion Tensor $\mathbf{N}$. Secondly, the extra-stress tensor is determined for $\mathbf{v} = \mathbf{u}^{n+1}$ by solving

$$\mathbf{v} \cdot \nabla \boldsymbol{\Sigma}^{n+1} - \nabla \mathbf{v}^\top \cdot \boldsymbol{\Sigma}^{n+1} - \boldsymbol{\Sigma}^{n+1} \cdot \nabla \mathbf{v} + \mathbf{Z}\left(\Lambda, \eta_p, \boldsymbol{\Sigma}^{n+1}\right) = 2\frac{\eta_p}{\Lambda}\mathbf{D}\left(\mathbf{v}\right) \qquad (4.20)$$

with respect to $\boldsymbol{\Sigma}^{n+1}$. In case of integral models, this step would be replaced by evaluating the stress integral from Eq. (2.43b) based on Finger tensors evolved by means of Eq. (2.43d). As a final step, the Diffusion Tensor $\mathbf{M}^{n+1}$ is calculated according to

$$\mathbf{T} - \mathbf{M}^{n+1} \cdot \mathbf{D}\left(\mathbf{v}\right) = \mathbf{0} \qquad (4.21)$$

where $\mathbf{T} = \mathbf{\Sigma}^{n+1}$ and $\mathbf{v} = \mathbf{u}^{n+1}$. For initialising the next outer iteration of the decoupled solution method, $\mathbf{N} = \mathbf{M}^{n+1}$ is considered in Eq. (4.19). Hence, the residual of the original problem formulation indeed is recovered in case of a converging scheme, since the resulting solution $(\mathbf{u}, \mathbf{\Sigma}, p)$ satisfies

$$-2\eta_s \nabla \cdot \mathbf{D}\left(\mathbf{u}\right) - \nabla \cdot \mathbf{\Sigma} + \nabla p = \mathbf{0}$$

as the contributions including the Diffusion Tensor vanish from Eq. (4.19). Additionally, the discrete versions of the continuity as well as constitutive equations are satisfied. In this context, the numerical solution is considered to be converged, once the residual in a decoupled step regarding the Stokes subproblem, that is based on Eq. (4.19), falls below a certain tolerance. The corresponding stress tensor will be computed from Eq. (4.20) according to the converged velocity field to obtain the full nonlinear solution regarding the primal flow variables.

To contrast the operator splitting technique described above especially with the mixed iteration proposed in Eq. (4.16), the decoupled iteration procedure is written as a coupled system of the form

$$\begin{pmatrix} \mathcal{L} + \mathcal{T}\left(\boldsymbol{\mu}^n\right) + \mathcal{J}^u & \mathcal{B} & \mathbf{0} & \mathbf{0} \\ \mathcal{B}^\top & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathcal{D} & \mathbf{0} & \mathcal{K}\left(\boldsymbol{u}^{n+1}\right) + \mathcal{J}^\sigma & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathcal{S} & \mathcal{N}\left(\boldsymbol{u}^{n+1}\right) + \mathcal{J}^\mu \end{pmatrix} \begin{pmatrix} \boldsymbol{u}^{n+1} \\ \boldsymbol{p}^{n+1} \\ \boldsymbol{\sigma}^{n+1} \\ \boldsymbol{\mu}^{n+1} \end{pmatrix} = \begin{pmatrix} \mathcal{C}\boldsymbol{\sigma}^n + \mathcal{T}\left(\boldsymbol{\mu}^n\right)\boldsymbol{u}^n \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} \tag{4.22}$$

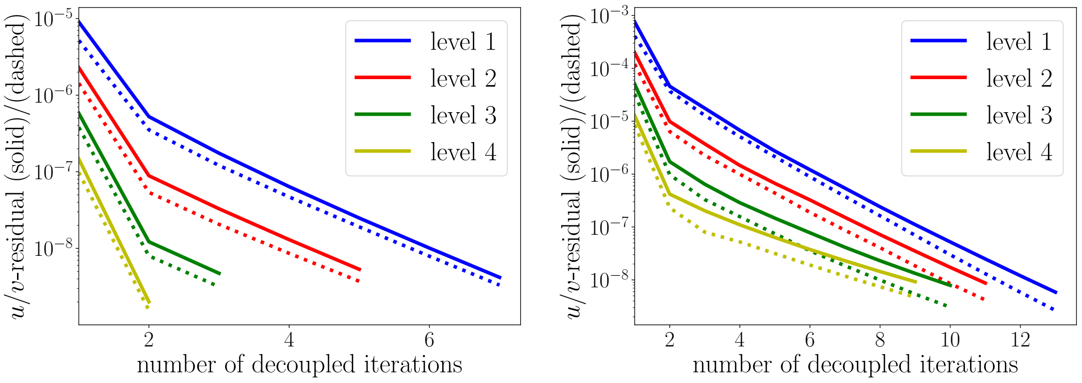
resulting from the Finite Element discretisations presented in Chpt. 2. Similar to the mixed iteration, the discrete stress operator in the momentum equation is "replaced" by the Tensor Stokes operator acting on the velocity field, which potentially provides a diffusive operator even in the non-solvent case. But, in the "Jacobian" no operator arises derived with respect to the Diffusion Tensor, since this quantity is not considered as unknown in the Stokes subproblem. In this way, the solver behaviour might be improved compared to the mixed iteration, as the – regarding the original problem formulation – "unnatural" contribution is not present. However, the separate blocks or subproblems of the above system actually are solved subsequently, which leads to a block-wise fixed-point iteration consisting of solving decoupled linear problems regarding the flow quantities $(\mathbf{u}, p)$, $\mathbf{\Sigma}$ and $\mathbf{M}$. Again, the unstabilised discrete problem is considered in the following, unless stated otherwise.

Similar to the results presented above, the numerical study on the behaviour of this block fixed-point iteration is started off by considering the Oldroyd-B model. Hence, the solvent contribution to the viscosity in the flow model is taken to be non-vanishing. The corresponding number N of decoupled steps needed for convergence is listed in Tab. 4.17 regarding several mesh levels as well as relaxation times. Note, that the initial solutions are chosen in the same way as in case of the mixed iteration analysed beforehand. It turns out, that the block fixed-point iteration shows a slightly improved behaviour compared to the mixed iteration from Tab. 4.15, since the number of outer steps is smaller in case of all considered mesh levels and escpecially for $\Lambda > 0.001$. Moreover, the decoupled solution scheme allows for computing the solution corresponding to $\Lambda = 2.5$ on level 4 and $\Lambda = 5.0$ on level 1, which is not possible by means of the coupled approach. Thereby, the iteration numbers concerning $\Lambda = 2.5$ are similar to the smaller relaxation time $\Lambda = 1.0$ in case of the mixed iteration.

Λ level	0.001	0.01	0.1	1.0	2.5	5.0
1	12	7	9	13	19	64
2	10	5	7	11	19	✗
3	7	3	5	10	20	✗
4	5	2	3	9	23	✗

Table 4.17: Number of outer iterations of the block fixed-point iteration for the Oldroyd-B model at $\beta = 0.5$ for increasing Λ and mesh levels

Hence, the block fixed-point iteration behaves (slightly) more stable than the mixed iteration, which might be explained by means of the structure of the problem matrix outlined above. But naturally, the performance quality of Newton's method applied in previous tests is not reached and (significantly) higher relaxation times than before neither are computable. It has to be emphasised, that the considered Diffusion Tensor has a significant stabilising effect regarding the numerical solver, since the number of decoupled steps increases drastically when neglecting the Tensor Stokes operator in Eq. (4.22) even for $\eta_s > 0$ and small relaxation times. In fact, a number of decoupled steps greater than 2500, which is set as maximum number of iterations here, is needed for solving the above problem formulation for $\Lambda = 0.001$, $\beta = \frac{\eta_s}{\eta_0} = 0.5$ and $\mathbf{N} \equiv \mathbf{0}$ already on mesh level 1. As indicated by the numerical results presented above, the number of decoupled steps probably will even further increase with mesh refinement – in case convergence is obtained at all.



(a) $\Lambda = 0.01$

(b) $\Lambda = 1.0$

Figure 4.39: Convergence behaviour of the block fixed-point iteration for the Oldroyd-B model at $\beta = 0.5$ for different Λ and mesh levels

In Fig. 4.39, exemplary convergence plots are depicted concerning the residuals of the components of the velocity field. It turns out, that the residuals are decreased by one digit in the first decoupled iterations, while for smaller relaxation times a linear convergence of lower speed is observed regarding the remaining iterations, similar to the mixed iteration (see Fig. 4.36(a)). In addition, the initial residual is decreased with mesh refinement and the convergence speed stays constant, which leads to a decreasing number of decoupled steps. Note, that the residuals in case of $\Lambda = 0.01$ on level 4 fall below the prescribed tolerance already after two decoupled steps, which is why no "slower range" of convergence is observed here (see Fig. 4.39(a)). Regarding higher Λ, the convergence speed decreases with mesh refinement as realised by means of Fig. 4.39(b), although not that drastically as in case of the mixed iteration, since the number of decoupled steps is still decreasing for $\Lambda = 1.0$ regarding the block fixed-point iteration.

As a next step, the block fixed-point iteration is applied for solving the UCM model, where the diffusive operator in the momentum equation from Eq. (4.19) weighted with η_s is not present. The resulting iteration numbers are presented in Tab. 4.18.

level \ Λ	0.001	0.01	0.1	0.5
1	393	98	229	326
2	873	103	324	1338
3 ($\gamma_u = 0.0$)	1373	71	334	>5000 ✗
3 ($\gamma_u = 0.1$)	11	5	13	94
4	643	19	161	>5000 ✗

Table 4.18: Number of outer iterations of the block fixed-point iteration for the UCM model at increasing Λ and mesh levels

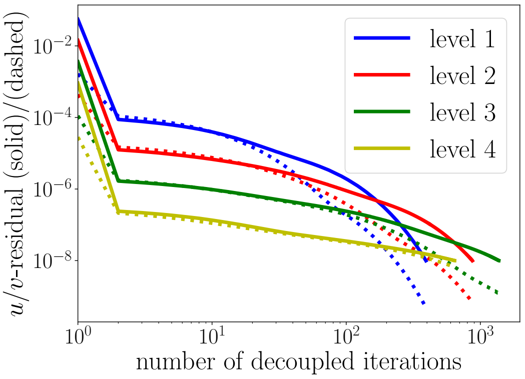
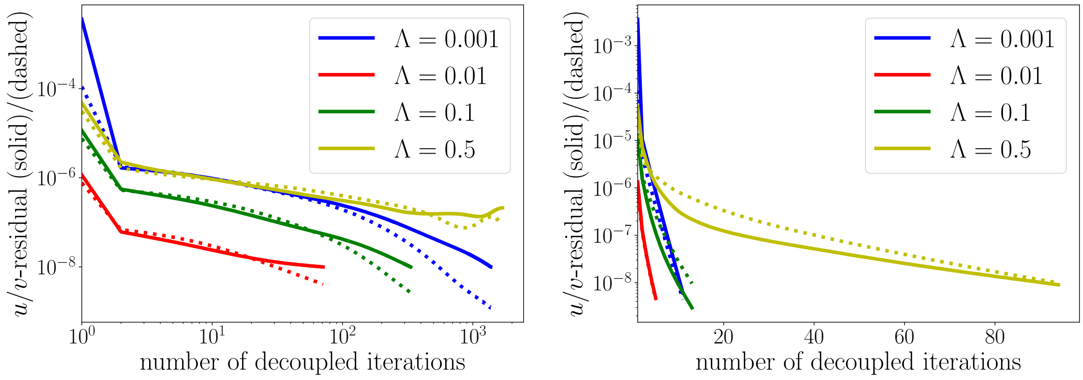


Figure 4.40: Convergence behaviour of the block fixed-point iteration for the UCM model at $\Lambda = 0.001$ for several mesh levels

As expected, an increasing number of decoupled steps is required for convergence compared to the Oldroyd-B model even in case of small relaxation times. At the same time, a decreasing convergence speed is observed with mesh refinement as realised from Fig. 4.40. But, the convergence behaviour at larger iteration numbers is similar to the mixed iteration, since the convergence speed increases towards the solution (see Fig. 4.37(a)). Note, that a smaller number of decoupled steps is needed for computing $\Lambda = 0.001$ on level 4 compared to level 2 and 3 due to the decreased initial residual. Moreover, no convergence on higher mesh levels is obtained for $\Lambda = 0.5$, similar to the mixed iteration (see Tab. 4.16), again highlighting the increased numerical difficulties in the non-solvent case.

Again, an improved convergence behaviour is observed for a DEVSS-like scheme, where the Diffusion Tensor is set as $\mathbf{M} = 2\eta_p\mathbf{I}$ in Eq. (2.51c). In case of lower relaxation times $\Lambda \in \{0.001, 0.01, 0.1\}$, the number of outer iterations is very similar to the results from Tab. 4.18, since the actual Diffusion Tensor at small Λ is close to the Newtonian setting. But, a converging iterative method is obtained by means of the DEVSS-like configuration even for the UCM model at $\Lambda = 0.5$, where roughly 2800 are needed for sufficiently reducing the corresponding residuals. In contrast, the outer decoupled solution scheme stagnates in case of the actual Tensor Diffusion approach including $\gamma_u = 0.0$ similar to the mixed iteration (see Fig. 4.41(a)). Thus, the block fixed-point iteration including the Diffusion Tensor according to the DEVSS results in a superior numerical approach, although still quite high iteration numbers are obtained. Possibly, applying the actual version of the DEVSS from Eq. (2.23) would give a further improved solver behaviour, where the strain-rate tensor itself is approximated by means of an additional flow variable instead of a basically scalar quantity as above.



(a) $\gamma_u = 0.0$ (b) $\gamma_u = 0.1$

Figure 4.41: Convergence behaviour of the block fixed-point iteration for the UCM model at different Λ for several choices of the EOFEM parameter γ_u on level 3

Similar to the mixed iteration, a possible way of improving the performance of the actual Tensor Diffusion approach is provided by the EOFEM stabilisation. The resulting accelerated convergence behaviour of the block fixed-point iteration is illustrated by means of the plots of the velocity residuals for several relaxation times regarding mesh level 3 (see Fig. 4.41(b)). Especially in case of lower relaxation times, a fast convergence speed is observed, while $\Lambda = 0.5$ again leads to a slower reduction of the residuals, which is observed for the mixed iteration as well (see Fig. 4.38). But, the latter seems to behave more consistent regarding the stabilised UCM model, since the residuals concerning $\Lambda \in \{0.001, 0.01, 0.1\}$ are reduced with similar speed. In contrast, a varying convergence speed is observed with respect to the block fixed-point iteration, in addition leading to a higher number of steps compared to the mixed iteration at $\Lambda = 0.5$. Furthermore, the DEVSS gives similar iteration numbers as the actual Tensor Diffusion approach in case of $\Lambda = 0.5$

and $\gamma_u = 0.1$, where 102 respectively 94 outer iterations are required, while the DEVSS is inferior regarding the mixed iteration.

Overall, the mixed and block fixed-point iterations lead to a similar solver behaviour, while the latter is slightly superior in case of the Oldroyd-B model. Regarding the UCM model, which is more challenging from a numerical point of view due to the vanishing solvent viscosity, both (unstabilised) solution approaches do not behave satisfactory. As expected, a significant growth of the iteration numbers in the solution process – according to the complexity of the numerical problem – is observed compared to the Oldroyd-B model. Naturally, applying EOFEM stabilisation results in an improved solver behaviour, where the mixed iteration seems to be slightly more "stable". In detail, the convergence speed is constant for lower relaxation times and the iteration numbers for $\Lambda = 0.5$ are smaller compared to the block fixed-point iteration. However, the convergence behaviour of Newton's method is not recovered by any of the two approaches, which thus are of pure fixed-point type. Consequently, none of the iteration procedures in the present state seem to be suitable candidates regarding alternative solution techniques to the pure original or Tensor Diffusion approaches. Especially concerning integral constitutive equations, future research needs to exploit the potential benefits of the Tensor Diffusion approach with respect to the design of segregated solution methods to possibly obtain improvements compared to existing techniques like the DEVSS. But, the Tensor Diffusion approach at least allows for applying a decoupled solution scheme in non-solvent case, which is not possible in terms of the original problem formulation. Furthermore, the novel approach results in a similar or even improved solver behaviour as the DEVSS in case of introducing EOFEM stabilisation, which gives some hope to be able to actually derive a reasonable decoupled solution scheme including Tensor Diffusion.

Chapter 5

Conclusion

In this thesis, a novel approach to simulate viscoelastic fluid flows is introduced with special emphasis on pure polymer melts, that is viscoelastic fluids without a solvent contribution to the viscosity. Therefore, arising numerical difficulties and challenges are outlined by describing existing Finite Element techniques for simulating such types of fluids. It turns out, that the stability as well as regularity of the numerical problem resulting from non-solvent differential or integral viscoelastic flow models are weakened, which affects the applicability of certain Finite Element (or Finite Volume) approximations and numerical solution techniques. On the one hand, the discrete spaces regarding the velocity and stress fields need to satisfy an additional inf-sup or LBB condition in order to obtain a stable mixed Finite Element formulation of the viscoelastic model. On the other hand, fast and efficient multigrid solvers as part of a Newton scheme within a monolithic solution approach behave non-robust because of the vanishing diffusive operator in the momentum equation. Due to the same reason, operator splitting techniques or decoupled solution approaches are not applicable, which are particularly required for deriving a practical numerical method for treating integral constitutive equations. In this regard, the novel Tensor Diffusion approach is proposed, which replaces the extra-stress tensor in the momentum equation of the viscoelastic flow model by a product of the strain-rate tensor and the so-called Diffusion Tensor, that is a nonsymmetric tensor-valued viscosity. In doing so, an approach similar to the DEVSS is obtained by introducing an (additional) artificial viscosity contribution into the (discrete) momentum equation, which conceptually resolves the issues of the numerical approach mentioned above. Moreover, the diffusive operator recovered in terms of this novel approach is related to the nature of the problem in a much more physical way than a simple scalar-valued artificial viscosity. Thus, a potentially improved numerical approach might be obtained compared to existing techniques. However, several problem formulations of the viscoelastic flow model can be derived in terms of the Tensor Diffusion approach, where the corresponding diffusive operator needs to be symmetrised to realise a reasonable numerical scheme.

Similar to the DEVSS, the four-field formulation of the Tensor Stokes problem is obtained by considering the Diffusion Tensor as additional flow variable, which can be determined by means of a simple algebraic equation resulting from the underlying decomposition of the extra-stress tensor. In case of differential models, this problem formulation might be further developed by directly inserting the stress decomposition also into the constitutive equation. Thereby, again a three-field formulation is derived according to the original flow model, which possibly allows a more sophisticated numerical determination of the Diffusion Tensor. But, the applicability of this approach is limited to differential material models, since an equivalent procedure concerning integral constitutive laws is not applicable straightforward. The potentially most powerful formulation of the Tensor Stokes problem is obtained by assuming, that the Diffusion Tensor is explicitly known or given according to a specific viscoelastic flow configuration and material law. Hence, the complex rheology of the fluid, originally described by means of a differential or integral viscoelastic model, is “hidden” inside the Diffusion Tensor. In this case, the need of considering the nonlinear material behaviour by means of a constitutive equation or by explicitly calculating the extra-stress tensor would be removed. Thus, the problem formulation can be reduced to a generalised Stokes-like problem, where the nonlinear material behaviour of viscoelastic fluids is characterised by means of a tensor-valued viscosity function. Assuming this can be done for several material models and complex flow configurations, the numerical effort for simulating highly nonlinear viscoelastic fluids may be reduced significantly. For example, solution techniques especially designed for solving the (Navier-)Stokes equations may be applied. Thus, a highly developed, efficient and robust numerical framework can be taken into account for simulating viscoelastic fluid flows. At the same time, addressing corresponding three- or even four-field formulations of viscoelastic flow models would not be necessary.

In this work, the algebraic four-field formulation as well as the pure Tensor Stokes problem mentioned above are taken into account for validating and evaluating the novel Tensor Diffusion approach. The underlying assumption, that a corresponding stress decomposition exists in general, is verified in a first step for Poiseuille-like flow configurations. In this framework it is worked out, that the Diffusion Tensor indeed can be given analytically in case of specific material models. Thus, the complete viscoelastic model actually can be reduced to a generalised Tensor Stokes problem, that is a generalised Stokes-like problem (in terms of velocity and pressure only) including a tensor-valued viscosity depending on the shear rate of the flow. Besides, the Diffusion Tensor is calculated numerically, in case no analytic derivation is possible. On this basis, the numerically or (semi-)analytically determined Diffusion Tensor is prescribed in the pure Tensor Stokes problem for simulating Poiseuille-like flows. In doing so, viscoelastic effects originally arising from various differential or integral (non)linear viscoelastic flow models are recovered within corresponding two-dimensional Finite Element simulations. Consequently, the original nonlinear viscoelastic flow solution can be simply computed from a Stokes-like problem, where the velocity and pressure fields are the only unknowns and the resulting extra-stress tensor is computed in a simple post-processing step. In this context it is realised, that several differential or integral constitutive equations like the Giesekus, PSM or Wagner model consist of underlying mathetical properties preventing the computation of arbitrary high relaxation times. One possibility for resolving this issue is provided by introducing slip boundary conditions, which allows for successfully calculating increasing relaxation times in case of an accordingly increasing amount of slip. In contrast, the linear or exponential PTT model does not show such a behaviour, which makes this model attractive regarding Finite Element simulations of industrial applications including realistic material and model parameters. However, the foundational applicability of the Tensor Diffusion approach is further validated concerning general two-dimensional flow configurations by means of the Flow around cylinder benchmark. In contrast to certain viscoelastic channel flows, the Diffusion Tensor needs to be determined numerically in this context, which is done by means of the four-field formulation of the Tensor Stokes problem. In terms of validating the novel approach, the drag coefficients as well as the stress profiles on the surface and in the wake of the cylinder are compared to results from the literature plus the original problem formulation, that is the three-field formulation including the velocity, stress and pressure fields. It turns out, that the numerical results are reproduced quite well for both, the typical benchmark configuration – including a non-homogeneous solvent viscosity – as well as the much more challenging non-solvent case, where the Oldroyd-B or UCM and the Giesekus model are taken into account.

Furthermore, Finite Element simulations of viscoelastic fluid flows within a rounded contraction are considered for evaluating the Tensor Diffusion approach, again by means of the four-field formulation of the Tensor Stokes problem. Thereby, it is illustrated, that approximating the Diffusion Tensor by element-wise constant polynomials, which allow jumps over element edges, is a reasonable choice. Actually, "natural" discontinuities occur in the Diffusion Tensor variable in such flow configurations, which at the same time should not become too large to not affect the behaviour of the numerical solution scheme. In this regard, the need of introducing EOFEM stabilisation with respect to the function values of the Diffusion Tensor variable is demonstrated, which is already successfully applied in terms of the Flow around cylinder benchmark. Another aspect of the Tensor Diffusion approach analysed in this context deals with the linear solver as part of a monolithic Newton scheme. Compared to the original approach, applying multigrid techniques is improved significantly when considering the UCM model within the Tensor Stokes problem, since a more robust behaviour is observed and higher relaxation times are computable. But, there is still the need to further develop such numerical solution techniques, especially regarding non-solvent viscoelastic fluid flows. In fact, a direct linear solver still is superior to multigrid techniques when applied within Newton's method, which is outlined is further investigations. In detail, the Tensor Diffusion approach is evaluated concerning the High Weissenberg Number Problem by successively increasing the relaxation time for both, the original as well as Tensor Stokes problem. Again, a stabilising effect due to the (additional) tensor-valued diffusion is observed in case of the UCM model. Thereby, slighty higher relaxation times can be reached for applying less amount of stabilisation with respect to the velocity and stress fields compared to the original problem formulation. In contrast, the original approach behaves very stable in case of the PTT model, where the Tensor Stokes formulation leads to an inferior numerical scheme. Possibly, the PTT model – similar to the Giesekus model – is more stable per se, which is why the additional diffusion provided by means of the Tensor Diffusion approach does not lead to an improved method. Recall, that the Diffusion Tensor is approximated in this study by means of piecewise constant polynomials, while the stress approximation in terms of the original approach is of higher order. Hence, a downgraded numerical scheme might be obtained also due to the lower-order approximation, possibly resulting in lower computable relaxation times. As an alternative, the differential three-field formulation of the Tensor Stokes problem might lead to an improved calculation of the Diffusion Tensor, which in turn may improve the numerical method. However, prototypical iteration procedures based on the Tensor Diffusion approach are analysed as well, which are intended to solve the original problem formulation and potentially improve the behaviour of the numerical solver. It turns out, that the proposed monolithic as well as segregated solution approaches result in an acceptable solver behaviour regarding the Oldroyd-B model at least for lower relaxation times. But, serious limits of these approaches are detected for higher relaxation times and a vanishing solvent viscosity, both leading to an increasing complexity of the numerical problem. Nevertheless, a decoupled solution approach – especially being relevant concerning integral material models – can not be realised by means of the original non-solvent problem formulation, but the Tensor Diffusion approach.

In summary, the Tensor Diffusion approach proposed in this thesis provides the basis for a reasonable alternative to existing numerical techniques for simulating viscoelastic fluid flows characterised by differential or integral material models. The fundamental applicability of this novel approach is highlighted concerning channel flows as well as two-dimensional configurations like the Flow around cylinder benchmark. Thereby, viscoelastic flow characteristics, which are initially described by the original problem formulation, are reproduced quite well by solving the Tensor Stokes problem. Moreover it is indicated, that the original approach indeed might be improved by inserting Tensor Diffusion into the (numerical) problem, in detail concerning the application of multigrid techniques within Newton's method, the High Weissenberg Number Problem or the implementation of decoupled solution schemes.

As part of future work, a further study on actually modelling the tensor-valued viscosity needs to be performed to improve the current state the approach, where the Diffusion Tensor is calculated numerically by means of an algebraic equation. For example, this might be done by means of a partial differential equation arising from inserting the stress decomposition into the (differential) constitutive equation regarding the extra-stress tensor, which is illustrated conceptually in this work as well. As an alternative, the Diffusion Tensor can be considered as a fourth-order tensor, which provides additional degrees of freedom within the (numerical) calculation to improve its underlying properties. Moreover it should be investigated, whether the numerical solver – especially the linear solver within the Newton scheme – can be further improved by fine-tuning the application of multigrid techniques. In addition, the realisation of a decoupled solution scheme based on the Tensor Diffusion approach needs to be tackled to implement a – compared to existing techniques possibly improved – numerical treatment of integral constitutive laws. However, the potentially most attractive goal of future research work regarding general two- (and also three-)dimensional configurations is provided by establishing the novel Tensor Diffusion approach in the same way as proposed for Poiseuille-like flows. Consequently, the Diffusion Tensor is desired to be modelled explicitly according to the nonlinear material behaviour of viscoelastic fluids – even for complex flow configurations. As a vision, the corresponding flow model might then be reduced to a pure Tensor Stokes problem, that is to a generalised non-Newtonian Stokes-like problem including a tensor-valued viscosity only depending on (the gradient of) the velocity field. In doing so, viscoelastic material behaviour could be predicted without the need to consider a differential or integral constitutive equation or even a stress variable at all, which might significantly improve the efficiency of corresponding numerical simulation tools.

Bibliography

[1] M. J. Crochet, Numerical simulation of viscoelastic flow: A review, Rubber Chemistry and Technology 62 (3) (1989) 426–455. doi:10.5254/1.3536253.

[2] F. P. Baaijens, An iterative solver for the devss/dg method with application to smooth and non-smooth flows of the upper convected maxwell fluid, Journal of Non-Newtonian Fluid Mechanics 75 (2) (1998) 119 – 138. doi:10.1016/S0377-0257(97)00086-4.

[3] A. Ouazzi, S. Turek, Numerical methods and simulation techniques for flow with shear and pressure dependent viscosity, in: M. Feistauer, V. Dolejsi, P. Knobloch, K. Najzar (Eds.), Numerical Mathematics and Advanced Applications, Springer, 2003, pp. 668–676, enumath 2003 Prague; ISBN-Nr. 3-540-21460-7.

[4] R. G. Larson, Constitutive Equations for Polymer Melts and Solutions, Butterworths Series in Chemical Engineering, Butterworth-Heinemann, 1988.

[5] C. W. Macosko, R. G. Larson, K. (Firm), Rheology : principles, measurements, and applications, New York : VCH, 1994, originally published as ISBN 1560815795.

[6] P. W., J. Kroll, S. Turek, Evaluation of nonlinear differential models for the simulation of polymer melts, Kautschuk Gummi Kunststoffe (317) (2017) 48–52.

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

[8] H. Giesekus, Die elastizität von flüssigkeiten, Rheologica Acta 5 (1966) 29–35.

[9] L. L. Ferrás, J. M. Nóbrega, F. T. Pinho, Analytical solutions for channel flows of phan-thien–tanner and giesekus fluids under slip, Journal of Non-Newtonian Fluid Mechanics 171-172 (2012) 97 – 105. doi:10.1016/j.jnnfm.2012.01.009.

[10] N. Phan-Thien, A nonlinear network viscoelastic model, Journal of Rheology 22 (1978) 259. doi:10.1122/1.549481.

[11] R. Keunings, Finite element methods for integral viscoelastic fluids, 2003.

[12] P. Bollada, T. Phillips, A modified deformation field method for integral constitutive models, Journal of Non-Newtonian Fluid Mechanics 163 (1) (2009) 78 – 87. doi:10.1016/j.jnnfm.2009.07.001.

[13] E. Peters, M. Hulsen, B. van den Brule, Instationary eulerian viscoelastic flow simulations using time separable rivlin–sawyers constitutive equations, Journal of Non-Newtonian Fluid Mechanics 89 (1) (2000) 209 – 228. doi:10.1016/S0377-0257(99)00026-9.

[14] M. Hulsen, E. Peters, B. van den Brule, A new approach to the deformation fields method for solving complex flows using integral constitutive equations, Journal of Non-Newtonian Fluid Mechanics 98 (2) (2001) 201 – 221. doi:10.1016/S0377-0257(01)00110-0.

[15] M. A. Hulsen, P. D. Anderson, The deformation fields method revisited: Stable simulation of instationary viscoelastic fluid flow using integral models, Journal of Non-Newtonian Fluid Mechanics 262 (2018) 68 – 78. doi:10.1016/j.jnnfm.2018.03.001.

[16] M. A. Hulsen, A sufficient condition for a positive definite configuration tensor in differential models, Journal of Non-Newtonian Fluid Mechanics 38 (1) (1990) 93 – 100. doi:10.1016/0377-0257(90)85034-V.

[17] H. M. Laun, Description of the non-linear shear behaviour of a low density polyethylene melt by means of an experimentally determined strain dependent memory function, Rheologica Acta 17 (1) (1978) 1–15. doi:10.1007/BF01567859.

[18] M. H. Wagner, J. Meissner, Network disentanglement and time-dependent flow behaviour of polymer melts, Die Makromolekulare Chemie 181 (7) (1980) 1533–1550. doi:10.1002/macp.1980.021810716.

[19] M. Tomé, J. Bertoco, C. Oishi, M. Araujo, D. Cruz, F. Pinho, M. Vynnycky, A finite difference technique for solving a time strain separable k-bkz constitutive equation for two-dimensional moving free surface flows, Journal of Computational Physics 311 (2016) 114 – 141. doi:10.1016/j.jcp.2016.01.032.

[20] J. V. D. Zanden, M. Hulsen, Mathematical and physical requirements for successful computations with viscoelastic fluid models, Journal of Non-Newtonian Fluid Mechanics 29 (1988) 93 – 117. doi:10.1016/0377-0257(88)85052-3.

[21] J. G. Heywood, R. Rannacher, S. Turek, Artificial boundaries and flux and pressure conditions for the incompressible navier–stokes equations, International Journal for Numerical Methods in Fluids 22 (5) (1996) 325–352. doi:10.1002/(SICI)1097-0363(19960315)22:5<325::AID-FLD307>3.0.CO;2-Y.

[22] M. Renardy, Mathematical Analysis of Viscoelastic Flows, Society for Industrial and Applied Mathematics, USA, 2000.

[23] E. Fernandez-Cara, F. Guillén, R. Ortega, Mathematical modeling and analysis of viscoelastic fluids of the oldroyd kind, Handbook of Numerical Analysis 8 (12 2002). doi:10.1016/S1570-8659(02)08005-5.

[24] J. Marchal, M. Crochet, A new mixed finite element for calculating viscoelastic flow, Journal of Non-Newtonian Fluid Mechanics 26 (1) (1987) 77 – 114. doi:10.1016/0377-0257(87)85048-6.

[25] R. Rannacher, S. Turek, Simple nonconforming quadrilateral stokes element, Numerical Methods for Partial Differential Equations 8 (2) (1992) 97–111. doi:10.1002/num.1690080202.

[26] Chapter i - the steady-state stokes equations, in: R. Temam (Ed.), Navier–Stokes Equations, Vol. 2 of Studies in Mathematics and Its Applications, Elsevier, 1977, pp. 1 – 156. doi:https://doi.org/10.1016/S0168-2024(09)70069-4.

[27] D. Boffi, L. Gastaldi, On the quadrilateral q2–p1 element for the stokes problem, International Journal for Numerical Methods in Fluids 39 (11) (2002) 1001–1011. doi:10.1002/fld.358.

[28] J. Baranger, D. Sandri, A formulation of stokes's problem and the linear elasticity equations suggested by the oldroyd model for viscoelastic flow, ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique 26 (2) (1992) 331–345.

[29] V. Ruas, Finite element methods for the three-field stokes system in $\mathbb{R}^3$: Galerkin methods, ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique 30 (4) (1996) 489–525. URL http://www.numdam.org/item/M2AN_1996__30_4_489_0

[30] J. Bonvin, M. Picasso, R. Stenberg, Gls and evss methods for a three-field stokes problem arising from viscoelastic flows, Computer Methods in Applied Mechanics and Engineering 190 (29) (2001) 3893 – 3914. doi:https://doi.org/10.1016/S0045-7825(00)00307-8.

[31] R. Codina, Finite element approximation of the three-field formulation of the stokes problem using arbitrary interpolations, SIAM J. Numerical Analysis 47 (2008) 699–718. doi:10.1137/080712726.

[32] V. Ervin, H. Lee, L. Ntasin, Analysis of the oseen-viscoelastic fluid flow problem, Journal of Non-newtonian Fluid Mechanics 127 (2005) 157–168.

[33] M. Artemov, E. Baranovskii, Mixed boundary-value problems for motion equations of a viscoelastic medium, Electronic Journal of Differential Equations 2015 (2015) 1–9.

[34] V. Girault, P.-A. Raviart, Finite Element Methods for Navier-Stokes Equations: Theory and Algorithms, Springer Berlin Heidelberg, Berlin, Heidelberg, 1986. doi:10.1007/978-3-642-61623-5.

[35] J. Baranger, D. Sandri, Finite element approximation of viscoelastic fluid flow: Existence of approximate solutions and error bounds, Numerische Mathematik 63 (1992) 13–27. doi:https://doi.org/10.1007/BF01385845.

[36] R. Fattal, R. Kupferman, Constitutive laws for the matrix-logarithm of the conformation tensor, J. Non-Newt. Fluid Mech. 123 (2004) 281–285.

[37] R. Fattal, 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) (2005) 23 – 37. doi:10.1016/j.jnnfm.2004.12.003.

[38] M. Crochet, V. Legat, The consistent streamline-upwind/petrov-galerkin method for viscoelastic flow revisited, Journal of Non-Newtonian Fluid Mechanics 42 (3) (1992) 283–299, cited By 56. doi:10.1016/0377-0257(92)87014-3.

[39] M. Fortin, A. Fortin, A new approach for the fem simulation of viscoelastic flows, Journal of Non-Newtonian Fluid Mechanics 32 (3) (1989) 295 – 310. doi:10.1016/0377-0257(89)85012-8.

[40] D. Rajagopalan, R. C. Armstrong, R. A. Brown, Finite element methdos for calculation of steady, viscoelastic flow using constitutive equations with a newtonian viscosity, Journal of Non-Newtonian Fluid Mechanics 36 (1990) 159 – 192. doi:10.1016/0377-0257(90)85008-M.

[41] R. Guénette, M. Fortin, A new mixed finite element method for computing viscoelastic flows, Journal of Non-Newtonian Fluid Mechanics 60 (1) (1995) 27 – 52. doi:10.1016/0377-0257(95)01372-3.

[42] A. Fortin, A. Zine, An improved gmres method for solving viscoelastic fluid flow problems, Journal of Non-Newtonian Fluid Mechanics 42 (1) (1992) 1 – 18. doi:https://doi.org/10.1016/0377-0257(92)80001-E.

[43] G. D'Avino, M. Hulsen, Decoupled second-order transient schemes for the flow of viscoelastic fluids without a viscous solvent contribution, Journal of Non-Newtonian Fluid Mechanics 165 (23) (2010) 1602 – 1612. doi:10.1016/j.jnnfm.2010.08.007.

[44] G. D'Avino, M. Hulsen, P. Maffettone, Decoupled transient schemes for viscoelastic fluid flow with inertia, Computers & Fluids 66 (2012) 183 – 193. doi:10.1016/j.compfluid.2012.06.023.

[45] O. M. Coronado, D. Arora, M. Behr, M. Pasquali, Four-field galerkin/least-squares formulation for viscoelastic fluids, Journal of Non-Newtonian Fluid Mechanics 140 (1) (2006) 132 – 144, special Issue on the XIVth International Workshop on Numerical Methods for Non-Newtonian Flows, Santa Fe, 2005. doi:10.1016/j.jnnfm.2006.03.016.

[46] M. A. Hulsen, R. Fattal, R. Kupferman, Flow of viscoelastic fluids past a cylinder at high weissenberg number: Stabilized simulations using matrix logarithms, Journal of Non-Newtonian Fluid Mechanics 127 (1) (2005) 27 – 39. doi:10.1016/j.jnnfm.2005.01.002.

[47] H. Damanik, J. Hron, A. Ouazzi, S. Turek, A monolithic fem approach for the log-conformation reformulation (lcr) of viscoelastic flow problems, Journal of Non-Newtonian Fluid Mechanics 165 (19) (2010) 1105 – 1113. doi:10.1016/j.jnnfm.2010.05.008.

[48] A. Afonso, P. Oliveira, F. Pinho, M. Alves, The log-conformation tensor approach in the finite-volume method framework, Journal of Non-Newtonian Fluid Mechanics 157 (1) (2009) 55 – 65. `doi:10.1016/j.jnnfm.2008.09.007`.

[49] O. M. Coronado, D. Arora, M. Behr, M. Pasquali, A simple method for simulating general viscoelastic fluid flows with an alternate log-conformation formulation, Journal of Non-Newtonian Fluid Mechanics 147 (3) (2007) 189 – 199. `doi:10.1016/j.jnnfm.2007.08.005`.

[50] M. A. Carrozza, M. A. Hulsen, M. Hütter, P. D. Anderson, Viscoelastic fluid flow simulation using the contravariant deformation formulation, Journal of Non-Newtonian Fluid Mechanics 270 (2019) 23 – 35. `doi:10.1016/j.jnnfm.2019.07.001`.

[51] H. Damanik, FEM simulation of non–isothermal viscoelastic fluids, Ph.D. thesis, TU Dortmund (May 2011).

[52] D. Arnold, D. Boffi, R. Falk, Approximation by quadrilateral finite elements, Mathematics of computation 71 (239) (2002) 909–922.

[53] S. Turek, A. Ouazzi, Unified edge–oriented stabilization of nonconforming FEM for incompressible flow problems: Numerical investigations, Journal of Numerical Mathematics 15 (4) (2007) 299–322.

[54] H. Damanik, J. Hron, A. Ouazzi, S. Turek, Monolithic newton-multigrid solution techniques for incompressible nonlinear flow models, International Journal for Numerical Methods in Fluids 71 (2) (2013) 208–222. `doi:10.1002/fld.3656`.

[55] S. Vanka, Block-implicit multigrid solution of navier-stokes equations in primitive variables, Journal of Computational Physics 65 (1) (1986) 138 – 158. `doi:10.1016/0021-9991(86)90008-2`.

[56] A. Ouazzi, S. Turek, Efficient multigrid and data structures for edge-oriented fem stabilization, in: A. B. de Castro, D. Gómez, P. Quintela, P. Salgado (Eds.), Numerical Mathematics and Advanced Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 520–527.

[57] H. Wobker, S. Turek, Numerical studies of vanka-type smoothers in computational solid mechanics, Advances in Applied Mathematics and Mechanics February Adv. Appl. Math. Mech 1 (2009) 29–55.

[58] J. Yoo, H. Choi, On the steady simple shear flows of the one-mode giesekus fluid, Rheologica Acta 28 (1) (1989) 13–24. `doi:10.1007/BF01354764`.

[59] G. Schleiniger, R. J. Weinacht, Steady poiseuille flows for a giesekus fluid, Journal of Non-Newtonian Fluid Mechanics 40 (1) (1991) 79 – 102. `doi:10.1016/0377-0257(91)87027-U`.

[60] J. Joie, D. Graebling, Numerical simulation of polymer flows using nonconforming finite elements, Computers & Fluids 79 (2013) 178–189. `doi:10.1016/j.compfluid.2013.03.012`.

[61] S. Claus, T. Phillips, Viscoelastic flow around a confined cylinder using spectral/hp element methods, Journal of Non-Newtonian Fluid Mechanics 200 (2013) 131 – 146, special Issue: Advances in Numerical Methods for Non-Newtonian Flows. `doi:10.1016/j.jnnfm.2013.03.004`.

[62] Y. Fan, R. Tanner, N. Phan-Thien, Galerkin/least-square finite-element methods for steady viscoelastic flows, Journal of Non-Newtonian Fluid Mechanics 84 (2) (1999) 233 – 256. `doi:10.1016/S0377-0257(98)00154-2`.

[63] F. Pimenta, M. Alves, Stabilization of an open-source finite-volume solver for viscoelastic fluid flows, Journal of Non-Newtonian Fluid Mechanics 239 (2017) 85 – 104. `doi:10.1016/j.jnnfm.2016.12.002`.