On numerical simulations of viscoelastic fluids.

Dariusz Niedziela

Vom Fachbereich Mathematik der Universität
Kaiserslautern zur Verleihung des akademischen
Grades Doktor der Naturwissenschaften (Doctor rerum
naturalium, Dr. rer. nat.) genehmigte Dissertation.

1. Gutachter: Priv.-Doz. Dr. Oleg Iliev,
2. Gutachter: Prof. Dr. Raytcho Lazarov.

Vollzug der Promotion: 29. Juni 2006

D 386

To my wife Ewa and my daughter Maja.

Acknowledgments.

I would like to thank all my friends, my colleagues and my family for their support during the time of my PhD research. In particular, my supervisors Priv.-Doz. Dr. Oleg Iliev and Priv.-Doz. Dr. Arnulf Latz for their help, many discussions and very useful advises during all the stages of my PhD research. Next, to my wife Ewa and my daughter Maja for their support and every single day we have spent together. Further, I would like to thank to Prof. Helmut Neunzert, Prof. Wojciech Okrasinski and Prof. Michael Junk for giving me possibility to do my PhD in Kaiserslautern.

I would also like to thank Dr. Konrad Steiner and Fraunhofer ITWM institute for offering me PhD position at the department of Flows and Complex Structures.

Finally, thank to Dr. Maya Neytcheva, Dr. Dimitar Stoyanov, Dr. Joachim Linn and Dr. Vadimas Starikovicius for their help and productive discussions that accelerated my work.

Contents

1 Introduction and outline. 7

1.1 Constitutive equations for Non–Newtonian fluids. 8
1.2 Objectives and outline of the thesis. 10

2 Governing equations. 15

2.1 The balance equations. 15
2.1.1 Conservation of mass. 17
2.1.2 Conservation of momentum. 17
2.2 Newtonian fluids. 19
2.3 Generalized Newtonian fluids. 20
2.3.1 Shear viscosity. 20
2.3.2 Extensional viscosity. 21
2.3.3 Dependence of viscosity on pressure and temperature. 22
2.4 Viscoelastic fluids. 23
2.4.1 Integral constitutive equation. 23
2.4.2 Doi-Edwards model. 25
2.4.3 Oldroyd B model. 27
2.5 Summary. 29

3 Solution of the governing equations. 31

3.1 Time discretization. 31
3.2 Projection type methods. 33
3.2.1 Coupled momentum projection algorithm. 34
3.3 Fully coupled method. 36
3.4 Finite Volume discretization. 37
3.4.1 Discretization of the momentum equations. 38
3.4.2 Discretization of the mixed derivatives. 40
3.4.3 Discretization of the pressure correction equation (PCE). . . . 41
3.4.4 Discretization of discrete divergence and gradient operators. . 44

4 Approximation of the constitutive equation. 47

4.1 Backward Lagrangian Particle Method (BLPM). 47
4.2 Deformation Field Method (DFM). 50
4.3 Calculation of the partial orientation tensor. 51

4.4 Approximation of the extra stress tensor. 52
4.5 Non–uniform discretization of the memory integral. 52
4.6 Calculation of the chain stretch. 54
4.7 Few words about additional storage and approximation used in BLPM. 55

5 Preconditioning techniques for the saddle point problems. 57
5.1 Preconditioners for coupled momentum projection method. 58
5.2 Preconditioners for untransformed fully coupled system. 62
5.2.1 Block Gauss–Seidel preconditioner to untransformed saddle-point problem. 62
5.2.2 Indefinite block triangular preconditioner to untransformed saddle-point problem. 63
5.3 Preconditioners for transformed fully coupled system. 64
5.3.1 Block diagonal preconditioner to transformed saddle-point problem. 65
5.3.2 Block lower triangular preconditioner to transformed saddle-point problem. 66
5.3.3 Block Gauss–Seidel preconditioner to transformed saddle-point problem. 68

6 Numerical results. 71
6.1 Simulations of shear–thinning fluids. 71
6.2 Extensional viscosity effect. 74
6.3 Simulations of viscoelastic fluids. 77
6.3.1 Oldroyd B constitutive equation. 78
6.3.2 Doi Edwards constitutive equation. 84
6.4 Performance of iterative solvers. 90
6.5 Summary. 97

7 Concluding remarks. 103

List of Symbols. 105

List of Figures. 109

List of Tables. 112

Bibliography. 113

Chapter 1

Introduction and outline.

Non–Newtonian fluids abound in many aspects of life. They appear in nature, where most of body fluids like blood and mucus are non-Newtonian ones. Also, many food products like, for example, mayonnaise, ketchup, egg white, honey, cream cheese, molten chocolate belong to such class of fluids. Paints, that must be easily spread under the action of stress, but should not flow spontantenously once applied to the surface, as well as printer inks, lipstick are further examples. Another huge area of appearance of non–Newtonian fluids is plastic industry. The examples are molten plastics and other man–made materials formed to produce everyday wealth like textiles, plastic bags, plastic toys, through the processes like extrusion, moulding, spinning, for example. Often non–Newtonian materials are created by addition of various polymers. The detergent industry adds polymers to shampoos, gels, liquid cleaning to improve their rheological properties. Non–Newtonian fluids are also used in motor industry. Multi–grade oils have polymer additives that change the viscosity properties under extremes of pressure and temperature. Precise and low cost prediction of properties of viscoelastic fluids, mentioned above, can help to reduce the overall production cost of goods made of those fluids. One of the means to achieve this goal is to use simulation tools that involves mathematical (numerical) methods. Therefore, in this thesis we focus on numerical simulations of viscoelastic fluids. As a possible area of applicability of the work presented here one can think, for example, of the plastic molding.

Mathematically, the set of the equations describing incompressible fluids is expressed by continuity and momentum equations as

$$\nabla \cdot \mathbf{v} = 0, \qquad \tfrac{D(\rho \mathbf{v})}{Dt} = -\nabla p + \nabla \cdot \boldsymbol{\tau},$$

where and $\frac{D}{Dt}$ denotes the material derivative, $\nabla\cdot$ and ∇ denote divergence and gradient, respectively. $\mathbf{v}$ stands for velocity, ρ for density, p denotes pressure and $\boldsymbol{\tau}$ denotes stress tensor. Clearly, if thermal flows are modeled, the equation of conservation of the energy has to be added to the above system. However, in this thesis we consider incompressible and isothermal viscoelastic fluids. To close the above system of equations, the stress tensor has to be completed by a constitutive equation.

1.1 Constitutive equations for Non–Newtonian fluids.

Viscoelastic fluids are examples of a class of fluids called non–Newtonian. These are the fluids, for which, contrary to the Newtonian ones, a linear relation between the stress tensor ($\boldsymbol{\tau}$) and the rate–of–deformation tensor ($\boldsymbol{\gamma}$) do not hold. Therefore, they require more complicated constitutive relations to close the system of equations, that has to be solved. Among a huge number of models one can distinguish between three main classes of fluids involving an algebraic, a differential or an integral constitutive equation.

Generalized Newtonian fluids. The first class express stress tensor through some algebraic formula postulated a priori. Such models fit an experimental measurements for various data like shear–rate ($\dot{\gamma}$), extensional–rate ($\dot{\epsilon}$), pressure (p), etc (see [25, 32]). All those variables can influence viscosity (η) of non–Newtonian fluids, what leads further to different flow patterns, stress distributions, pressure drops comparing with a Newtonian ones. These kind of models are referred to as the generalized Newtonian fluids, and stress tensor in this case can be written in a general form as

$$\boldsymbol{\tau} = f(\eta(\dot{\gamma}, \dot{\epsilon}, p, ...), \boldsymbol{\gamma}).$$

Here, f denotes model dependent algebraic relation. Despite a clear drawback of not capturing the elastic effects of viscoelastic fluids, such generalized Newtonian models are still widely used in industrial applications, and therefore are also considered in this thesis.

Viscoelastic fluids: differential constitutive equations. The second class of constitutive relations, that include elasticity effects, consist of differential models. They can be written in a general form as

$$\tfrac{D\boldsymbol{\tau}}{Dt} = f(\nabla \mathbf{v}, \boldsymbol{\gamma}),$$

where f is model dependent tensor function. These are the most common models used nowadays in simulations of viscoelastic fluids. Among many, one can list the most often taking a stand models like Oldroyd–type, FENE–type, Phan–Thien Tanner, Giesekus (see [1, 2, 3, 17, 32, 38]). For an isothermal problem, the set of highly coupled differential equations, consisting of the continuity equation, the momentum equations and the constitutive equation, have to be solved. A variety of numerical approaches have been used to solve viscoelastic flow problems, like finite difference methods, finite element methods, spectral methods and finite volume methods. Often the calculations were restricted to stationary creeping flows. In the case of the axisymmetric abrupt 4:1 contraction flows, simulations have showed growth of the vortices, although for higher Weissenberg numbers than observed in experiments. Weissenberg number We is defined by the ratio of a characteristic length L in the specific flow and a characteristic velocity U multiplied by a characteristic relaxation time τ_{relax}, i.e. $We = \frac{\tau_{relax} U}{L}$. Here, the relaxation time τ_{relax} defines how much past deformations influence the stress. For purely elastic materials $\tau_{relax} = \infty$, i.e. they never forget their initial state, and for purely viscous fluids $\tau_{relax} = 0$. The viscoelastic materials are somehow in between, for which holds $0 < \tau_{relax} < \infty$. Over years, the researchers had to face the problem of performing stable calculations of viscoelastic fluids, modeled by differential constitutive equations, for high Weissenberg number flows. Early attempts to solve viscoelastic fluids problems failed to converge beyond $We = \mathcal{O}(10^{-1})$, which is merely a perturbation of the Newtonian case. Over decades, this problem has been partially resolved. Now, there exist many algorithms being able to perform stable simulations up to $We = \mathcal{O}(10)$ for various domains. One can argue, however, that viscoelastic flow computations are not yet robust and reliable procedure as for example classical Newtonian flow problems, and further improvements are still needed.

Viscoelastic fluids: integral constitutive equations. The last and final class of models discussed here consist of integral constitutive equations, which take a general form as

$$\boldsymbol{\tau} = \int_{-\infty}^{t} \mu(t,t') f_t(t'),$$

where $\mu(t,t')$ is the memory function and $f_t(t')$ is a model dependent nonlinear strain measure relative to the current time t. These are the most physically adequate models, since they take the full history of the deformations into account, not only the one which can be determined from current stress. The integral models express the memory of polymeric liquids, namely that the polymer stress carried by a fluid particle at current time of simulations is a function of the deformation history experienced at past times by this particle following its trajectory. The particle paths along which one has to compute the memory integral are not known a priori. Therefore, the problem is highly nonlinear, even under creeping flow conditions. Another challenge here is that the Lagrangian formulation of the constitutive model do not involve the Eulerian velocity field in an explicit manner. This can be resolved by Backward Lagrangian Particle Method (BLPM), for example. This method decouples the Lagrangian calculations of the stress tensor by recalculating the (upstream) particle paths at each time step of the simulations, with the Eulerian calculations of conservation of the mass and the momentum equations. In BLPM the stress tensor is calculated along particle paths. This method, however, has a drawback of being highly time and memory consuming, since the particle paths have to be calculated for each Eulerian grid point and each time step of simulations. Moreover, in most of the cases the intermediate positions of particle tracking do not coincide with the grid nodes. Therefore, some approximation formula has to be used at this point. Also a certain number of velocities and the quantities appearing in the integrals have to be stored additionally. This number depends on the relaxation time exhibited by a fluid, i.e. the longer the relaxation time τ_{relax} is, the higher is the number of stored quantities. An alternative method used to approximate integral constitutive equations is Deformation Field Method (DFM). This is the first Eulerian technique for solving time dependent flows with an integral constitutive equation introduced by Peters at al. ([37]). The basic idea behind DFM is that the deformation history is described by a finite number of deformation fields, which are convected and deformed by the flow field. The main advantage of this Eulerian technique is that it removes the need of recalculating the particle paths and the calculation of the extra stress tensor along them.

In this thesis we focus on the numerical simulations of generalized Newtonian fluids, obeying nonlinear relation between stress and strain tensors, and of viscoelastic fluids modeled by the integral constitutive equations. For the latter, we choose the most successful kinetic theory model for linear polymers, the Doi Edwards reptation model, allowing simulations of the concentrated polymer solutions (see [9, 19, 27, 36, 47, 48, 49]), as well as the integral Oldroyd B model, widely used in the simulations of dilute polymer solutions (see [1, 2, 3, 17, 38]).

1.2 Objectives and outline of the thesis.

This thesis aims to contribute
- to the analysis, development and validation of models for generalized Newtonian and for viscoelastic (non–Newtonian) fluids,
- to development and validation of robust and reliable algorithms for simulation of the generalized Newtonian flows,
- to comparison and validation of robust and reliable algorithms for simulation of the viscoelastic (non–Newtonian) flows, as well as
- to the software implementation of the developed algorithms.

The first objective of the thesis is to systematically study existing models for generalized Newtonian fluids and for non–Newtonian fluids, and to propose and analyze their proper extensions. One of the widely used models for the generalized Newtonian fluids is the one named after Carreau. This model has a drawback of not being able to predict experimentally observed growth of the vortices for shear-thinning fluids. To overcome this flaw, we propose a new anisotropic viscosity model. It possess two viscosities describing shear and extensional properties of the fluid, and can be considered as a natural extension of the isotropic viscosity Carreau model. The anisotropic viscosity model gives ability to predict growth of the vortices even for shear-thinning fluids, if additionally extensional-thickening is taken into account. That is exactly what the experimentalists observe (growth of the vortices is related to the extensional properties of the stress, i.e. fluids with unbounded extensional stress growth show growth of the vortices). Afterwards, we present validation of the models used in this thesis, describing the generalized Newtonian (Carreau constitutive equation, anisotropic viscosity model) and the non–Newtonian fluids (integral Oldroyd B and integral Doi Edwards constitutive equation), and compare them with already existing numerical results from simulations of the viscoelastic fluids obtained by differential counterpart (Oldroyd B model), or differential approximation (Doi Edwards model), as well as with physical experiments.

The second objective of this thesis is to develop and implement a robust and reliable algorithm for generalized Newtonian fluids. For such fluids the viscosity is modeled as a function that varies in both space and time. This is contrary to the Newtonian case where the viscosity is a constant value. Such variations of viscosity may differ couple of order of magnitude (see [22]). This results in strong coupling of the momentum equations through their viscous term. Additionally, the momentum equations are coupled through the convective term and the pressure. Due to varying viscosities, one has to take into account discretization of the mixed derivatives appearing in the momentum equations. In many commercial solvers those terms are treated in explicit way, i.e. calculated from the corresponding values computed in the previous time step and taken as a source term. This may lead to the restriction on the time step used in simulations due to the stability problems. To avoid it, here the momentum equations are discretized and treated in coupled manner, i.e. the mixed derivatives are discretized in implicit way. Efficient numerical simulations of the generalized Newtonian flows is not a trivial task. While numerical methods for the Newtonian flows are, in general, well studied (see [13, 15, 23, 43, 45, 50]), this is not a case for the generalized Newtonian ones. For the Newtonian fluids, projection type methods that decouple the momentum and the continuity equations are often used (for details and further references on projection methods see [13, 15, 23, 43]). Such methods might, however, not be efficient in the generalized Newtonian fluids simulations, especially for highly varying and large viscosities. In such cases an alternative to the segregated solvers, the fully coupled solvers, have to be used instead. Now, the momentum and the continuity equations are solved together. An interesting question is, which kind of solver, segregated or fully coupled, performs better and more stable simulations for the generalized Newtonian fluids. In most of the cases the system of partial differential equations, after linearization and discretization, is solved via an iterative method. To improve the performance of the iterative process, it is supplemented with an appropriate preconditioner. Again, much work has been done for Newtonian fluids (see [4, 10, 26, 42]). Recently, a numerical study of the performance of various iterative solvers applied to the generalized Newtonian flow equations was presented in [22]. However, still additional analysis of various preconditioners presented there is needed for better understanding.

The third objective of this thesis is to implement, validate, compare and systematically analyze algorithms for simulation of viscoelastic fluids modeled by integral–type constitutive equation. We consider two methods, namely Backward Lagrangian Particle Method and Deformation Field Method. Since the implemented algorithms, presented in this thesis, should be applicable for simulations of industrial processes, such as plastic moulding, they should be applicable for arbitrary two and three dimensional domains. This gives rise to two constraints: Firstly, the algorithm has to perform stable simulations in classical benchmark problems, like abrupt contraction, or contraction–expansion domains, also for high Weissenberg numbers. Secondly, the performed simulations should not take too much time and consume too much computer memory. An acceptable computational time, which clearly depend on the size of considered problem, should be rather of order of hours than days. All this is realized by careful approximation of the time integral constitutive equation.

The thesis is organized as follows. In Chapter 2, the equations describing the isothermal flow of viscous and viscoelastic fluids are derived. Throughout the thesis we consider incompressible fluids. The set of partial differential equations that form the well known system of Navier–Stokes equations is given. The equations are derived by using the conservation laws for mass and momentum, and imposing a linear relation between stress and rate–of-deformation tensors. Next, generalized Newtonian liquids are considered. As aforementioned, the class of the generalized Newtonian fluids consists of fluids having nonlinear relation between the stress tensor and the gradients of velocity, which leads to a strain rate dependent viscosity in addition to the dependence on thermodynamic variables, pressure and temperature. First, we show Carreau viscosity model being able to deal with shear–thinning, or shear–thickening properties of the fluid. Then, we discuss a similar model that additionally includes extensional viscosity properties of the generalized Newtonian liquids. The simulations are performed at constant temperature. To complete the picture of the complexity of the behavior of viscosity, we present further possible extensions of those models for the dependence on pressure and temperature. The last class of fluids discussed also here, the class of viscoelastic fluids, consist of both viscous and elastic effects. The elastic effects are described by integral constitutive equations. First, we present a general form of integral constitutive equation. Then two different models, describing concentrated and dilute polymer solutions respectively, are given. The concentrated polymer solutions and polymer melts are described by the integral Doi Edwards tube model. It is supplemented by the chain length fluctuation. The dilute polymer solutions are described by the integral Oldroyd B model.

Chapter 3 describes discretization and numerical methods used to solve the system of differential equations supplemented by a constitutive equations, given in Chapter 2. First, we describe how to handle the system of coupled integro–differential equations. The integral constitutive equation is splitted from the Navier–Stokes equations and taken as a source term in the momentum equations. Next, projection type solution methods are presented. In particular, a coupled momentum projection method, which is a modification of SIMPLE (Semi–Implicit Method for Pressure Linked Equations) algorithm, is given. The momentum equations are solved coupled, i.e. the discretization of all viscous terms is performed in implicit way. Next, we give short discussion about fully coupled methods, where the continuity and the momentum equations are solved together. Further, we present a finite volume discretization used in the simulations. Since a collocated variable arrangement is chosen, a special discussion focused on avoiding the checker–board pressure field is performed.

In Chapter 4, two known methods for approximating general integral constitutive equations are discussed. First, a Lagrangian method, namely Backward Lagrangian Particle Method, is presented. Then, we give an alternative Eulerian method, called Deformation Field Method. In both methods an integral constitutive equation is approximated by an infinite sum. We discuss possible ways to truncate this infinite summation in order to minimize the computational time and memory requirements needed for such discretization of the memory integrals.

In Chapter 5, we present preconditioners for saddle point problems. In particular, generalized Newtonian flow equations with varying viscosity are analyzed. Since the whole system of partial differential equations, that describe such fluids, is too complicated to analyze, the convective terms from the momentum equations are dropped. We can do so, since the interest is in slow flow regimes of viscoelastic fluids, that usually possesses large viscosities, thus having low Reynolds numbers. We give analytical results concerning eigenvalue distributions for both projection–type and fully coupled solvers. We also present and discuss the influence of the mixed derivatives on the performance of the iterative solvers.

In Chapter 6 we present numerical results. First, simulations of generalized Newtonian fluids are discussed. We perform the validation of our work for shear–thinning fluids. Next, extensional–thickening effect is taken in addition. As a result of such addition, contrary to the shear–thinning fluid only, we obtain growth of the vortices, which is in qualitative agreement with experimental observations. Next, we give the results from the simulations of dilute polymer solutions described by the integral Oldroyd B model. We show, that for low Weissenberg numbers it produces similar results as its differential counterpart. However, much higher Weissenberg number flows are accessible when integral constitutive equation is chosen. Moreover, good qualitative agreement with experiments is achieved. Further, we give simulation results of concentrated polymer solutions modeled by the integral Doi Edwards model. Again, we get very good agreement with experiments and the results obtained by its differential approximation. At the end of this Chapter, we present the discussion of the performance of different iterative solvers supplemented by different preconditioners in terms of the time step, number of grid points, changes of viscosity.

Finally, the research performed in this thesis is summarized in the concluding remarks.

Chapter 2

Governing equations.

In this Chapter, we present governing equations, which are used to describe the isothermal flow of viscous and viscoelastic fluids. In Section 2.1, we derive the physical and mathematical formulation of conservation laws, which lead to the system of Navier-Stokes equations describing the motion of viscous fluids. To obtain a complete set of equations, the stress tensor has to be specified by a constitutive equation. It describes typical behavior of considered material. Newtonian fluids obey linear relation between stress and rate–of–deformation tensors. In Section 2.2 we discuss this relation. Next, the class of non-Newtonian fluids is discussed. Such fluids can not be described by Navier-Stokes equations with a constant viscosity. The viscoelastic materials show both viscous and elastic behavior when they are deformed and require much more complicated constitutive equations. In Section 2.3, we present generalized Newtonian fluids. They relate stress and strain by some algebraic formula, being able to capture variations of viscosity depending on shear rates, elongational rates, pressure or temperature, for example. In Section 2.4, we discuss the models involving integral constitutive relation in order to account for the history of deformations. To simulate flows of dilute and concentrated polymer solutions, we use the integral Oldroyd B and Doi Edwards models, respectively.

2.1 The balance equations.

Many physical processes may be described by fundamental principles like conservation of certain quantities. Let $\mathbf{x}(t') \in \Omega \subset \mathbb{R}^d$ be the particle position in the domain Ω, which is a subset of $\mathbb{R}^d$ with space dimension d, at some time t'. Then the velocity $\mathbf{v} \in \Omega \subset \mathbb{R}^d$ of that particle at time $t' = t$ is defined as

$$\mathbf{v}(\mathbf{x}(t), t) = \frac{d\mathbf{x}(t')}{dt'}|_{t'=t}. \tag{2.1}$$

Let us consider transport of the physical quantity $\phi(\mathbf{x}, t)$ in an arbitrary volume $\mathcal{V}(t)$.

The total amount of the quantity ϕ contained in the volume $\mathcal{V}(t)$ equals

$$\int_{\mathcal{V}(t)} \phi(\mathbf{x}, t) d\mathbf{x}. \tag{2.2}$$

Thus, the rate of change of the quantity ϕ is expressed as

$$\frac{d}{dt} \int_{\mathcal{V}(t)} \phi(\mathbf{x}, t) d\mathbf{x}, \tag{2.3}$$

where both the integrand $\phi(\mathbf{x}, t)$, as well as the integration domain $\mathcal{V}(t)$, depend on t. Therefore we need the mathematical tool to be able to calculate such expressions. It is provided by the following theorem.

Theorem 1 (Reynolds transport theorem) *Let $\phi : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ be a continuously differentiable function. Then for each volume $\mathcal{V}(t)$ the following equality holds*

$$\frac{d}{dt} \int_{\mathcal{V}(t)} \phi(\boldsymbol{x}, t) d\boldsymbol{x} = \int_{\mathcal{V}(t)} [\frac{\partial \phi}{\partial t}(\boldsymbol{x}, t) + \nabla \cdot (\phi \boldsymbol{v})(\boldsymbol{x}, t)] d\boldsymbol{x}. \tag{2.4}$$

■

The proof may be found in [50]. Another useful mathematical tool that is used in the forthcoming derivations of conservation laws is expressed by the Gauss divergence theorem.

Theorem 2 (Gauss divergence theorem) *Let $\mathcal{V}(t)$ be a region in space with piecewise smooth, closed surface $\partial\mathcal{V}(t)$. Then the volume integral of the divergence $\nabla \cdot \phi(\boldsymbol{x}, t)$ of $\phi(\boldsymbol{x}, t)$ over $\mathcal{V}(t)$ and the surface integral of $\phi(\boldsymbol{x}, t)$ over the boundary $\partial\mathcal{V}(t)$ of $\mathcal{V}(t)$ are related by*

$$\int_{\mathcal{V}(t)} \nabla \cdot \phi(\boldsymbol{x}, t) d\boldsymbol{x} = \int_{\partial\mathcal{V}(t)} \vec{n} \cdot \phi(\boldsymbol{x}, t) d\boldsymbol{a}, \tag{2.5}$$

where $\vec{n}$ is outward pointing unit normal vector, $d\boldsymbol{x}$ is volume and $d\boldsymbol{a}$ is surface element.

■

For proof, see [50]. This theorem states, that the quantity ϕ changes the value only due to the flow into or away from the volume $\mathcal{V}(t)$ through its boundary $\partial\mathcal{V}(t)$.

2.1.1 Conservation of mass.

The first law describing the fluid motion is the law of conservation of mass. It states, that the mass of a piece of fluid contained in the volume $\mathcal{V}(t)$ is neither created nor destroyed. Let $\rho(\mathbf{x}, t)$ be the density of the fluid at a position $\mathbf{x}$ and time t. The total mass of the fluid contained in the volume $\mathcal{V}(t)$ is given as $\int_{\mathcal{V}(t)} \rho(\mathbf{x}, t) d\mathbf{x}$. The fact that the change of the mass is constant means that

$$\frac{d}{dt} \int_{\mathcal{V}(t)} \rho(\mathbf{x}, t) d\mathbf{x} = 0. \tag{2.6}$$

Applying *Theorem 1* to the above formula and taking $\phi = \rho$ leads to

$$\int_{\mathcal{V}(t)} [\frac{\partial \rho(\mathbf{x}, t)}{\partial t} + \nabla \cdot (\rho(\mathbf{x}, t)\mathbf{v}(\mathbf{x}, t))] d\mathbf{x} = 0. \tag{2.7}$$

Since the derivation holds for arbitrary volume $\mathcal{V}(t)$, we conclude the differential form of the conservation of mass, also called continuity equation, as

$$\frac{\partial \rho(\mathbf{x}, t)}{\partial t} + \nabla \cdot (\rho(\mathbf{x}, t)\mathbf{v}(\mathbf{x}, t)) = 0. \tag{2.8}$$

Fluids, such that the density of each element remains constant during the motion, what means $\frac{d\rho}{dt} = 0$, are called incompressible. In this case the equation (2.8) simplifies its form to

$$\nabla \cdot \mathbf{v}(\mathbf{x}, t) = 0. \tag{2.9}$$

2.1.2 Conservation of momentum.

The second principal physical law that fluid obeys is the conservation of momentum. This quantity is defined as the product of mass and velocity

$$\int_{\mathcal{V}(t)} \rho(\mathbf{x}, t)\mathbf{v}(\mathbf{x}, t) d\mathbf{x}. \tag{2.10}$$

According to the second Newton's law, the rate of change of the momentum of the fluid contained in the volume $\mathcal{V}(t)$ is equal to all forces acting on this fluid. There are two types of forces acting on the fluid, body forces and surface forces. The body force, like gravity acts on a fluid particle, and is proportional to its mass. It can be expressed as $\int_{\mathcal{V}(t)} \rho(\mathbf{x}, t)\mathbf{g}(\mathbf{x}, t) d\mathbf{x}$, where $\mathbf{g}(\mathbf{x}, t)$ denotes the density of gravity force. The surface forces are the one acting on the boundaries $\partial\mathcal{V}(t)$ of the fluid volume $\mathcal{V}(t)$, and is usually described by the stress tensor $\bar{\boldsymbol{\sigma}}$. The stress consist of pressure and friction forces, which are coming from the interactions between fluid layers sliding one relative to the other. Taking all together, conservation of momentum reads as

$$\frac{d}{dt}\int_{\mathcal{V}(t)} \rho(\mathbf{x},t)\mathbf{v}(\mathbf{x},t)d\mathbf{x} = \int_{\mathcal{V}(t)} \rho(\mathbf{x},t)\mathbf{g}(\mathbf{x},t)d\mathbf{x} + \int_{\partial\mathcal{V}(t)} \vec{n}\cdot\bar{\sigma}(\mathbf{x},t)d\mathbf{a}. \tag{2.11}$$

Applying *Theorem 1* to the left hand side and *Theorem 2* to the second term on the right hand side transforms the above equation to

$$\int_{\mathcal{V}(t)} [\frac{\partial\rho(\mathbf{x},t)\mathbf{v}(\mathbf{x},t)}{\partial t} + \nabla\cdot(\rho(\mathbf{x},t)\mathbf{v}(\mathbf{x},t)\otimes\mathbf{v}(\mathbf{x},t)) - \rho(\mathbf{x},t)\mathbf{g}(\mathbf{x},t) - \nabla\cdot\bar{\sigma}(\mathbf{x},t)]d\mathbf{x} = 0, \tag{2.12}$$

where $\otimes$ stands for dyadic product. Since it holds for every volume $\mathcal{V}(t)$ we get the following differential form of conservation of momentum

$$\frac{\partial\rho(\mathbf{x},t)\mathbf{v}(\mathbf{x},t)}{\partial t} + \nabla\cdot(\rho(\mathbf{x},t)\mathbf{v}(\mathbf{x},t)\otimes\mathbf{v}(\mathbf{x},t)) = \rho(\mathbf{x},t)\mathbf{g}(\mathbf{x},t) + \nabla\cdot\bar{\sigma}(\mathbf{x},t). \tag{2.13}$$

As it was already mentioned the stress $\bar{\boldsymbol{\sigma}}$ consist of the pressure and the friction forces. Since pressure is the force acting on the fluid element in absence of velocity gradients, we can decompose $\bar{\boldsymbol{\sigma}}$ in two components

$$\bar{\boldsymbol{\sigma}} = -p\mathbf{I} + \boldsymbol{\sigma}, \tag{2.14}$$

where p denotes pressure, $\mathbf{I}$ identity operator and $\boldsymbol{\sigma}$ viscous part of the stress tensor. The pressure is defined to be positive, if compression forces are acting on the surface, i.e. if outward normal vector $\vec{n}$ is opposite to the force. Taking that into account and assuming incompressibility of the fluid, we can rewrite equation (2.13) in the following form

$$\rho\frac{\partial\mathbf{v}(\mathbf{x},t)}{\partial t} + (\rho\mathbf{v}(\mathbf{x},t)\cdot\nabla)\mathbf{v}(\mathbf{x},t) = \rho\mathbf{g}(\mathbf{x},t) - \nabla p + \nabla\cdot\boldsymbol{\sigma}(\mathbf{x},t). \tag{2.15}$$

In [32], one can find the proof that stress tensor $\boldsymbol{\sigma}$ exist and is symmetric, i.e. $\boldsymbol{\sigma} = \boldsymbol{\sigma}^{\boldsymbol{T}}$. Viscous stress $\boldsymbol{\sigma}$ is related to the velocity gradients, namely to the rate of deformation, alternatively called rate of strain or strain, tensor $\boldsymbol{\gamma}$. It is defined as

$$\boldsymbol{\gamma} = (\nabla\mathbf{v} + \nabla^T\mathbf{v}) \tag{2.16}$$

and describes how the fluid particle deforms within the flow. A good analysis of it may be found in [18]. The quantity, that measures the magnitude of the rate of strain is the mean shear-rate $\dot{\gamma}$, defined as

$$\dot{\gamma} = \sqrt{\frac{1}{2}[\boldsymbol{\gamma} : \boldsymbol{\gamma}]} = \sqrt{\frac{1}{2}\sum_{ij}\gamma_{ij}\gamma_{ji}}. \tag{2.17}$$

2.2 Newtonian fluids.

The most known fluids, for which the components $\boldsymbol{\sigma}$ of the viscous stress tensor are assumed to depend linearly on the instantaneous values of the strain, are called Newtonian. This assumption leads to the result

$$\boldsymbol{\sigma} = \eta_0(\boldsymbol{\gamma} - \frac{1}{3}\boldsymbol{\delta}(\nabla \cdot \mathbf{v})) + \zeta\boldsymbol{\delta}(\nabla \cdot \mathbf{v})), \tag{2.18}$$

where η_0 is the dynamic viscosity coefficient, ζ is called the second or the bulk viscosity and $\boldsymbol{\delta}$ is the Kronecker delta defined as

$$\delta_{ij} = \begin{cases} 1, i = j \\ 0, i \neq j. \end{cases} \tag{2.19}$$

Justification of above relation may be found in [18]. In equation (2.18), the first term on the right hand side corresponds to deformations without the change in volume, while the second term represents isotropic dilation. For shear flows the dynamic viscosity, also called shear viscosity, η_0 is important, while for compression flows is the bulk viscosity. However, the focus here is on incompressible fluids, for which the above equation (2.18) simplifies to

$$\boldsymbol{\sigma} = \eta_0\boldsymbol{\gamma}. \tag{2.20}$$

This relation together with the equations (2.9, 2.15) give the system of Navier-Stokes equations, that describe the flow of incompressible, isothermal Newtonian fluids.

Viscoelastic fluids are examples of a class of fluids, called non - Newtonian fluids. While the Newtonian fluids are characterized by a constant viscosity (i.e., constant ratio between shear stress and the rate of strain), the non-Newtonian fluids require more complicated constitutive relations in order to close the governing system of equations. In fact, variety of particular models are suggested in the literature for each of the forms of the constitutive relations. Furthermore, the particular models can differ in the way in which they are derived: either postulating an constitutive equation and fitting its coefficient from measurements (like for the generalized Newtonian fluids), or deriving the constitutive equation starting from microscopic phenomena (e.g., reptation theories, molecular considerations, etc.). Now, the constitutive relations can be in the form of a differential, or an integral equation (supplemented by further constitutive relations). For a good review of a most popular differential models used in simulations of viscoelastic models, the interested reader is referred to [32].

2.3 Generalized Newtonian fluids.

In this Section models, that describe the class of generalized Newtonian fluids, are discussed. These fluids may have extremely varying properties and behave in a different way as the Newtonian ones. The viscosity of materials can be influenced by variables such as shear–rate $\dot{\gamma}$, extensional–rate $\dot{\epsilon}$, for example. As soon as the viscosity starts to vary for different values of listed variables, the fluids are referred to the generalized Newtonian ones. In all cases, the relation between stress and strain is local in time, i.e. $\boldsymbol{\sigma} = \eta(\dot{\gamma}, \dot{\epsilon}, p, ...)\boldsymbol{\gamma}$.

2.3.1 Shear viscosity.

From a rheological point of view, the variation of viscosity with the shear-rate is regarded as being the most important. In this content one distinguishes two major classes of fluids, namely shear-thinning and shear-thickening fluids. The first class consists of materials which appear to be Newtonian at low shear-rates. However, after increasing shear-rates, the effective viscosity decreases. One of the reasons for that may, for example, be the macromolecules in suspension in the fluid, which tend to align in the direction of the flow. Typical behavior of viscosity versus shear-rate is given in Figure 2.1. The examples of such fluids are dilute suspension of solids, dilute solutions of high-molecular-weight polymers, liquid polymers and paper pulp. On the other hand some materials, like corn starch suspension, exhibit opposite behavior. Again, for the low shear-rates the relation between stress and strain is linear, however increase of shear-rates results in increase of viscosity. This shear-thickening behavior may result from the fact, that during the flow initially rolled in balls macromolecules start to unroll into long chains. This leads to growth of viscosity. In Figure 2.2 typical behavior of such materials is presented. In order to model shear-thinning or shear-thickening behavior of different materials, various viscosity models have been proposed in the literature. In all of them, the viscosity is expressed as an algebraic equation, which includes parameters that are used to fit given experimental data. From many models, that are reviewed in [32], we choose a very common one, namely Carreau model. It is expressed as

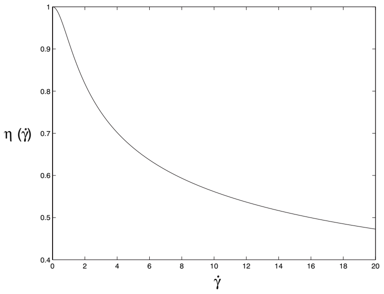

Figure 2.1: *Shear-thinning fluid. Viscosity versus shear-rate.*

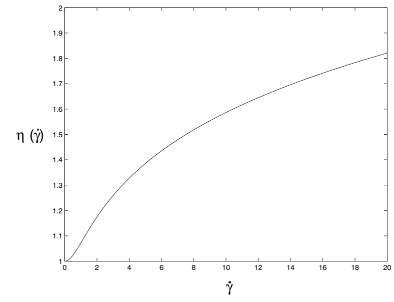

Figure 2.2: *Shear-thickening fluid. Viscosity versus shear-rate.*

$$\eta(\dot{\gamma}) = \eta_0(1 + (Cu\dot{\gamma})^2)^{\frac{(n-1)}{2}}, \tag{2.21}$$

where shear–rate $\dot{\gamma}$ is defined by equation (2.17), η_0 is the zero shear–rate viscosity and Cu, n are the parameters to fit experimental data. For $0 < n < 1$, the above formula models shear–thinning fluid, $n = 1$ corresponds to the Newtonian fluid, and finally $n > 1$ expresses shear–thickening behavior. Second parameter Cu, appearing in (2.21), is used to measure the significance of shear–thinning or shear-thickening. With the above defined viscosity, the stress tensor expressed in equation (2.20), transforms to

$$\boldsymbol{\sigma} = \eta(\dot{\gamma})\boldsymbol{\gamma}. \tag{2.22}$$

This model has showed a good ability in capturing shear–viscosity properties and is still widely used in commercial softwares. However, it is not the only one property of viscoelastic fluids.

2.3.2 Extensional viscosity.

Many complex fluids exhibit different behavior of viscosity in directions parallel and perpendicular to the flow direction. Therefore, it is often necessary to distinguish between extensional and shear viscosities, η_e and η respectively. Fluids, which show decrease of extensional viscosity with increase of extensional strain rate $\dot{\epsilon}$ are called extensional, or tension, thinning, whilst, when η_e grows with increase of $\dot{\epsilon}$, they are called extensional, or tension, thickening. Over many years models like Carreau, that can capture shear behavior of fluids were extensively studied. However, up to the knowledge of the author there is still a lack of simple algebraic models being able to deal with the combined effect of shear–thinning (thickening) and extensional–thickening (thinning). Simple means, that it should have similar structure as the Carreau model, thus could be considered as its natural extension. Moreover, this model should be easily applicable for an arbitrary flow domains. After private communications and discussions with Arnulf Latz [*A.Latz, Fraunhofer ITWM, private* *communications*], such model is proposed in the scope of this thesis. Let us first define extensional strain rate as

$$\dot{\epsilon} = \vec{t} \cdot \gamma \vec{t}, \tag{2.23}$$

where $\vec{t}$ is given by

$$\vec{t} = \frac{\mathbf{v}}{\|\mathbf{v}\|} \tag{2.24}$$

and denotes unit vector parallel to the flow direction. The same definition may be found in [12], where authors discuss how to perform the measurements of elongational viscosity. Now, having defined extensional strain rate $\dot{\epsilon}$, one can also use Carreau model to predict extensional viscosity behavior as

$$\eta_e(\dot{\epsilon}) = \eta_0(1 + (Cu_e\dot{\epsilon})^2)^{\frac{(n_e-1)}{2}} \tag{2.25}$$

with fitting parameters n_e, Cu_e. The first parameter indicates whether the fluid is extensional–thinning ($0 < n_e < 1$), extensional–thickening ($n_e > 1$), or has constant extensional viscosity η_0 ($n_e = 1$). The second parameter Cu_e is used to control the strength of extensional viscosity effects. Next step in modeling is to define the constitutive equation, which should include both, shear $\eta(\dot{\gamma})$ and extensional $\eta_e(\dot{\epsilon})$ viscosities. Therefore, taking into account the fact, that the diagonal terms of rate of strain tensor γ_{ii} represent the rate of elongation of an element of the fluid in respective $i-$ direction, whilst the off-diagonal terms $\gamma_{ij}(i \neq j)$ express deformations of a fluid element (what corresponds to shearing), the stress tensor will now be defined as

$$\sigma_{ij} = \eta(\dot{\gamma})(\gamma_{ij} - \delta_{ij}\gamma_{ii}) + \eta_e(\dot{\epsilon})\delta_{ij}\gamma_{ii}. \tag{2.26}$$

Note, that the shear viscosity η enters the off–diagonal terms of the stress, while the extensional viscosity η_e only the diagonal ones. This simple model includes both shear and extensional effects, modeled by $\eta(\dot{\gamma})$ and $\eta_e(\dot{\epsilon})$ respectively. Therefore, it allows simulations of shear–thinning and extensional–thickening fluids, for example.

2.3.3 Dependence of viscosity on pressure and temperature.

To close the picture of complexity of viscosity behavior, let us discuss its dependence from pressure and temperature, and possible extensions of models given by equations (2.21, 2.25). It is known, that viscosity of many fluids increases exponentially with increasing pressure. At low pressures this effect is negligible, however for high pressures it can be important. The dependence of viscosity on pressure can be modeled by Barus law (see [32]) as

$$\eta(p) = \eta_0 exp(\alpha p), \tag{2.27}$$

where α is material parameter that is estimated by best fitting of experimental data. This may be easily combined with Carreau models adding $exp(\alpha p)$ as a multiplicative factor to (2.21, 2.25). Also variations in temperature can result in significant changes in viscosity. The viscosity of most Newtonian liquids decreases as the temperature increases. This phenomenon is known as temperature-thinning. One can model it by including the temperature dependence in the zero-shear viscosity η_0, which can be expressed by Arrhenius law as

$$\eta_0(T) = a_1 exp(\frac{a_2}{T}), \tag{2.28}$$

where T is the absolute temperature of the fluid and a_1, a_2 are again fitting parameters. Combining all together, the originally presented functions of shear and extensional viscosities (2.21, 2.25) transform to

$$\eta(\dot{\gamma}, T, p) = \eta_0(T)(1 + (Cu\dot{\gamma})^2)^{\frac{(n-1)}{2}} \times exp(\alpha p) \tag{2.29}$$

and

$$\eta_e(\dot{\epsilon}, T, p) = \eta_0(T)(1 + (Cu_e\dot{\epsilon})^2)^{\frac{(n_e-1)}{2}} \times exp(\alpha p) \tag{2.30}$$

respectively.

2.4 Viscoelastic fluids.

Many natural and synthetic fluids are viscoelastic materials, for which the stress of a fluid particle depends not only on the current flow field, as for Newtonian or generalized Newtonian fluids, but also on the history of the deformation experienced by that particle. This can be specified by a differential, or an integral model. In this thesis only integral models are considered.

2.4.1 Integral constitutive equation.

Throughout the thesis constitutive equation of integral type is chosen as the most general, and the most physically adequate approach to predict viscoelastic behavior of the fluid. An attempt to simplify the study of the constitutive (differential and integral) models for viscoelastic fluids is based on introducing a Newtonian component of the viscoelastic tensor, thus conditionally separating the viscous and the elastic part of the stress tensor. This results in

$$\boldsymbol{\tau} = \boldsymbol{\sigma} + \mathbf{T}, \tag{2.31}$$

where $\boldsymbol{\tau}$ denotes total, $\boldsymbol{\sigma}$ - Newtonian, or generalized Newtonian, and $\mathbf{T}$ - extra, or alternatively called elastic, stress tensors, respectively. The stress tensor $\boldsymbol{\sigma}$ is given by equation (2.20, 2.22 or 2.26). The above splitting (2.31) can be interpreted physically as the solvent and elastic contributions to the stress in polymer solutions, or as the stress response associated with the fast ($\boldsymbol{\sigma}$) and slow ($\mathbf{T}$) relaxation modes. Extra stress tensor $\mathbf{T}$ expressed in the integral form is usually formulated in a Lagrangian framework, which means, that it is evaluated at a moving fluid particle. Consider such a fluid particle, that has a position vector $\mathbf{x}$ at the present time t and had a position vector $\mathbf{x}'$ at some past time t'. The deformation gradient tensor $\mathbf{E}$ is defined by

$$E_{ij} = \frac{\partial x_i}{\partial x'_j} \tag{2.32}$$

and expresses displacement of the particle moving from the point $\mathbf{x}'$ to $\mathbf{x}$. Such deformations introduce additional stress, that is coming, for example, from changes of orientation of polymers. However, how the changes of configuration of polymers influence the flow is very material dependent. Therefore, let $f(\mathbf{E}(t,t'))$ be some tensor function, that depends on deformation gradient tensor $\mathbf{E}$ and may take different forms for different fluids. Deformation tensor $\mathbf{E}$ satisfies the evolution equation

$$\frac{D\mathbf{E}(t,t')}{Dt} = \left(\nabla \mathbf{v}(t)\right)^T \mathbf{E}(t,t'). \tag{2.33}$$

To demonstrate this result, one has to take partial time derivative with respect to t' of (i,j)th component of $\mathbf{E}$, as follows

$$\frac{\partial E_{ij}}{\partial t} = \frac{\partial}{\partial t}\frac{\partial x_i}{\partial x'_j} = \frac{\partial v_i}{\partial x'_j} = \frac{\partial v_i}{\partial x_k}\frac{\partial x_k}{\partial x'_j}, \tag{2.34}$$

what after applying summation notation gives equation (2.33). Viscoelastic fluids are somewhere in between purely elastic materials, like rubber, and the Newtonian fluids, like water, for example. Elastic materials remember all the history of its deformations, what means that after applying a force to it, and then releasing, material attains its initial shape again. Newtonian fluid is viscous, and once a force is applied, it dissipates into heat. After removing the force fluid remains unchanged in its deformed shape. Viscoelastic materials possesses characteristic time, called relaxation time, after which the history of deformations is forgotten. Moreover, the younger deformations influence stress more, than the older ones. In the simplest models they are weighted by strain independent memory function $\mu(t,t')$, which takes an exponential form. This leads to integral model of general form

$$\mathbf{T} = \int_{-\infty}^{t} \mu(t,t') f(\mathbf{E}(t,t'))dt', \tag{2.35}$$

where the integral is taken along the past trajectories of the fluid particle. Clearly, for the non-Newtonian fluids the momentum equation can be written as

$$\rho\frac{\partial \mathbf{v}(\mathbf{x},t)}{\partial t} + (\rho\mathbf{v}(\mathbf{x},t)\cdot\nabla)\mathbf{v}(\mathbf{x},t) = \rho\mathbf{g}(\mathbf{x},t) - \nabla p + \nabla\cdot\boldsymbol{\sigma}(\mathbf{x},t) + \nabla\cdot\mathbf{T}(\mathbf{x},t). \quad (2.36)$$

Let us now discuss two possible choices of function $f(\mathbf{E}(t,t'))$ in the case of concentrated and dilute polymer solutions.

2.4.2 Doi-Edwards model.

Let us first discuss the Doi Edwards model (see [9]) that is used to simulate concentrated polymer solutions or polymer melts. Figure 2.3 presents schematic picture of such materials. They consist of highly entangled polymers, indicated by a black curves, that are immersed in Newtonian solvent. It is clear, that modeling has to take into account interactions of different polymers and relation between the polymers and solvent. Doi Edwards model is based on the so called reptation theory. Consider a polymer chain (also called primitive chain) marked in Figure 2.3. Suppose for a moment that other chains are frozen and regarded as fixed obstacles. Thus, one can assume, that the indicated chain lies in a tube-like region formed by surrounding chains (see Figure 2.4). If the polymer chain closed in such a tube moves perpendicularly to its own contour, it drags many surrounded polymers and therfore exhibits a large resistance. On the other hand, the movement of primitive chain along its contour is much easier. It is thus plausible to assume, that the major mode of its dynamics is reptation. Changes of the configuration of the primitive chain is expressed by the so called partial orientation tensor $\mathbf{Q}$

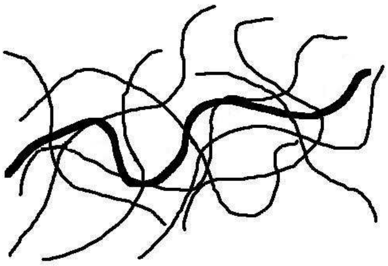

Figure 2.3: *Schematic picture of highly entangled polymer solutions.*

$$\mathbf{Q}(t,t') = \left\langle \frac{(\mathbf{E}(t,t')\mathbf{u}(t'))\otimes(\mathbf{E}(t,t')\mathbf{u}(t'))}{\parallel \mathbf{E}(t,t')\mathbf{u}(t') \parallel} \right\rangle_0 \frac{1}{\langle\parallel \mathbf{E}(t,t')\mathbf{u}(t') \parallel\rangle_0}, \quad (2.37)$$

where the brackets $(\langle\rangle_0)$ on the right–hand side denote an ensemble average over

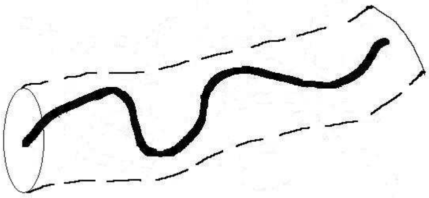

Figure 2.4: *Schematic picture of the tube model.*

unit vectors $\mathbf{u}(t')$ representing the orientation of a tube segment. $\otimes$ stands for the dyadic product and $\mathbf{E}(t,t')$ is the deformation tensor. As it was mentioned, one of the main assumptions in Doi Edwards reptation theory is that polymeric chains move along itself. Roughly speaking, (2.37) means that each tube segment $\mathbf{u}(t')$ of the deformed primitive chain, that initially can move in arbitrary direction, is restricted to remain tangent to the chain after the deformation. Doi Edwards model links microscopic level, on which the model is derived, with the macroscopic flow field using the affine deformation assumption. This means, that the primitive chain is deformed in the same way as the macroscopic deformation. Mathematically, one can express it as $\mathbf{u} \to \mathbf{Eu}$. Changes of the orientation of chains introduce extra stress described by

$$\mathbf{T} = G_e \int_{-\infty}^{t} \mu(t,t')\mathbf{Q}(t,t')dt', \tag{2.38}$$

where G_e is the elastic constant, and

$$\mu(t,t') = \frac{1}{\tau_d} exp\left(-\frac{(t-t')}{\tau_d}\right) \tag{2.39}$$

is the memory function that weights the contributions of the past deformations, which influences the current stress. τ_d is the relaxation time. The latter describes how fast the polymer chain leaves its constraint (tube like region). In general, longer polymers need longer time to relax. One can see, that Doi Edwards model takes the general integral form of constitutive equation, presented in Section 2.4.1, with $f(\mathbf{E}(t,t')) = G_e\mathbf{Q}(\mathbf{E}(t,t'))$. Over years, many additional effects not captured by the original Doi Edwards model (2.38), have been included into the model. These are: fluctuation of the contour length, the diffusive release of constraint by motion of the surrounding chains ([6, 21]), also called "double reptation", convective constraint release in fast flows ([27]), and the chain stretching, which allows the length of the chain to exceed its equilibrium length in fast flows. In this thesis the latter is also considered. For a dynamics of the chain stretch the model proposed by G. Ianniruberto and G. Marrucci ([20]) is chosen here. The stretch of the backbone of chain is controlled by the local force balance of dissipative drag and elastic recovery. In a flow, the extension of the tube around the backbone tends to drag the backbone segment with it. The elasticity of the backbone acts in the opposite direction. At flow larger than the reciprocal Rouse time, polymer chains get stretched. This is modeled by a separate evolution equation for the average stretch ratio $\lambda = \frac{L}{L_0}$, where L is a current tube length (averaged over all chains) and L_0 is its equilibrium value, by

$$\frac{D\lambda}{Dt} = (\nabla \mathbf{v})^T : \mathbf{S}\lambda - \frac{1}{\tau_R}(f\lambda - 1), \tag{2.40}$$

where $\mathbf{S} = \int_{-\infty}^{t} \mu(t,t')\mathbf{Q}(t,t')dt'$. The first term on the right hand side expresses the rate of elongation of a line segment of length ΔL in the tangential direction, i.e. it accounts for an affine deformations. The second describes the relaxation with the Rouse time τ_R taken as a characteristic time. To account for a finite extensibility of the chain, a non–linear factor

$$f = \frac{\lambda_{max} - 1}{\lambda_{max} - \lambda} \tag{2.41}$$

is chosen, where $\lambda_{max} = \frac{L_{max}}{L_0}$ is the maximum stretch ratio. Finally, the original Doi Edwards stress tensor (2.38) transforms to

$$\mathbf{T} = G_e f \lambda^2 \int_{-\infty}^{t} \mu(t,t')\mathbf{Q}(t,t')dt'. \tag{2.42}$$

The quadratic dependence of the stress on λ comes from the assumption of Hooke's law with nonlinear proportionality f, where expression $f\lambda$ denotes the tension of the chain. It is chosen to become unity for the unstretched state $\lambda = 1$, and to become λ for a hypothetical chain of unlimited extensibility ($\lambda_{max} \to \infty$), which enjoys a linear force law. For more details concerning this model we refer to [20].

2.4.3 Oldroyd B model.

Let us now discuss the second class of viscoelastic fluids considered here, namely the class of dilute polymer solutions. Schematic picture of such fluids is presented in Figure 2.5, and shows the distribution of the polymers submerged in the Newtonian solver. Now, contrary to the concentrated polymer solutions, the polymers are not entangled with each other, thus the interactions between different polymers are neglected. One of the models used to describe those kind of fluids is the Oldroyd B model. Historically, it was first proposed by Oldroyd in 1950 and was of differential form. Later, equivalent integral formulation was found. The proof of the equivalence of the integral and differential formulation can be found, for example, in [17]. Polymers are modeled as an elastic dumbbell immersed in the Newtonian solvent and consisting of two beds connected with a spring (see Figure 2.6). The motion of such a single dumbbell is affected by the forces exerted by the spring, Brownian forces due

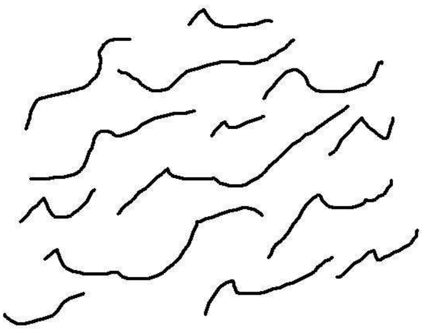

Figure 2.5: *Schematic picture of dilute polymer solutions.*

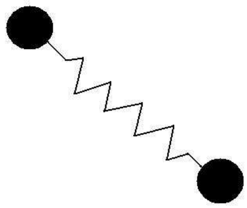

Figure 2.6: *The dumbbell model.*

to the impact on the beads of the solvent molecules, and the drag force coming from the differences of the velocity of the bead and the surrounding medium. Detailed derivation can be found in [32], for example. Such assumption lead to the polymeric stress tensor of the following form

$$\mathbf{T} = \frac{\eta_p}{\tau_{Oldr}^2} \int_{-\infty}^{t} dt' \mu(t,t') \mathbf{B}(t,t'), \tag{2.43}$$

where τ_{Oldr} is the relaxation time, η_p is a polymeric constant,

$$\mu(t,t') = exp\left(-\frac{(t-t')}{\tau_{Oldr}}\right) \tag{2.44}$$

is again the memory function, and finally

$$\mathbf{B}(t,t') = \mathbf{E}(t,t') \cdot \mathbf{E}^T(t,t') \tag{2.45}$$

is the so called Finger strain tensor. The latter describes the changes of the orientation of polymers that further influence the extra stress tensor. It is known, that the above model provides a constant shear viscosity and strain–hardening in extension

([32]). Therefore, combining the generalized Newtonian stress tensor (2.22) with the Oldroyd B integral stress tensor (2.43), one may account for the shear viscosity properties, by fitting $\eta_s(\dot{\gamma})$ from the experimental data, while the Oldroyd B model may account for the elastic effects of the material. The Oldroyd B model takes the general integral constitutive equation form, presented in Section 2.4.1, by choosing the function $f(\mathbf{E}(t,t')) = \frac{\eta_p}{\tau^2_{Oldr}}\mathbf{B}(\mathbf{E}(t,t'))$.

2.5 Summary.

In the balance equations, the stress tensor still has to be specified by the constitutive equation. It obeys different relations for different fluids, thus taking different phenomena into account. In the current Chapter, the balance equations, supplemented by the constitutive equations describing various viscous and viscoelastic fluids, have been presented. The resulting system of differential, or integro–differential equations can be summarized as follows:

The balance equations.

balance of mass:
$\nabla \cdot \mathbf{v} = 0$

balance of momentum:
$\rho\frac{\partial \mathbf{v}}{\partial t} + (\rho\mathbf{v}\cdot\nabla)\mathbf{v} = \rho\mathbf{g} - \nabla p + \nabla\cdot\boldsymbol{\sigma} + \nabla\cdot\mathbf{T}$

The constitutive equations.

Newtonian (viscous) fluids:
$\boldsymbol{\sigma} = \eta_0\boldsymbol{\gamma}, \qquad \mathbf{T} = 0$

generalized Newtonian fluids dealing with shear viscosity properties:
$\boldsymbol{\sigma} = \eta(\dot{\gamma})\boldsymbol{\gamma}, \qquad \mathbf{T} = 0$

generalized Newtonian fluids dealing with shear and extensional viscosity properties:
$\sigma_{ij} = \eta(\dot{\gamma})(\gamma_{ij} - \delta_{ij}\gamma_{ii}) + \eta_e(\dot{\epsilon})\delta_{ij}\gamma_{ii}, \qquad T_{ij} = 0$

non–Newtonian viscoelastic fluids, Oldroyd B model:
$\boldsymbol{\sigma} = \eta_0\boldsymbol{\gamma}, \qquad \mathbf{T} = \frac{\eta_p}{\tau^2_{Oldr}}\int_{-\infty}^{t} dt'\mu(t,t')\mathbf{B}(t,t')$

non–Newtonian viscoelastic fluids, Doi Edwards model:
$\boldsymbol{\sigma} = \eta_0\boldsymbol{\gamma}, \qquad \mathbf{T} = G_e f\lambda^2\int_{-\infty}^{t} \mu(t,t')\mathbf{Q}(t,t')dt'$

The set of equations describing Newtonian fluids is well studied, and the ways to discretize them and further perform the simulations can be found in many CFD books (see for example [13, 15, 45, 50]). However, performing stable and robust simulations of generalized–Newtonian and non–Newtonian fluids is still a challenging task. The viscosity of the former class is modeled by a time and space varying function, as it is, e.g., in Carreau model. This situation needs a careful numerical treatment since, in addition to the coupling between the balance equations, appearing also for the Newtonian fluids, the variations of viscosity result in strong coupling of the momentum equations through their viscous term. This may lead to stability problems. To overcome such problems, we present in this thesis a modification of classical projection–type methods (like SIMPLE, Chorin, etc.), where all the momentum equations are solved coupled (i.e., cross-derivative terms are discretized in implicit way), as well as the methods, where the momentum and the continuity equation are solved together using a proper preconditioner. We show here, that such approaches are much more robust, especially for large viscosity. Due to different algorithms presented in this thesis, different levels of the coupling of the balance equations and easier readability, we introduce some nomenclature: The projection–type methods solved with the momentum equations discretized in the implicit way (also the mixed derivatives), will be called *coupled momentum projection methods*, while the algorithms where all the balance equations are solved at once, *fully coupled methods*.

Chapter 3

Solution of the governing equations.

In this Chapter, we discuss the methods to handle the system of the given equations, presented in Chapter 2, and describing flows of the viscoelastic fluids. First, we give the time discretization and the way to decouple integral and differential parts of the given equations. Next, we investigate two different numerical solution schemes, namely operator splitting approach via pressure correction and direct coupled approach with simultaneous treatment of velocity and pressure. Later, we discuss the discretization method used in simulations, namely finite volume method on collocated grid. At the end of this Chapter, we present special treatment of pressure correction equation for projection type methods, and of divergence and pressure operators for fully coupled method, in order to avoid well known problem of checker-board pressure field.

3.1 Time discretization.

To simulate the flows of viscoelastic fluids, modeled by the integral–type constitutive equation (the general form given by equation (2.35)) one has to solve a strongly coupled system of integro-differential equations. A decoupling approach, based on fractional time step discretization is used within these thesis. However, work on more advanced coupling and decoupling approaches is a part of future research. Shortly, the numerical algorithm reads as follows:

Step 1. Solve the continuity (2.9) and the momentum (2.36) equations taking the polymeric tensor $\mathbf{T}$ in (2.36) as a source term (i.e., from the previous time step calculation).

Step 2. Solve all the equations involved with integral constitutive equation (2.35) using the velocity values obtained at *Step 1.*

Let us now write down the time discretization scheme in more details. The discretization with respect to space variables is done with the finite volume method on cell-centered grid (see, e.g., [15, 23]), what is discussed in Section 3.4. Suppose for a moment, that the continuity (2.9) and the momentum (2.36) equations are discretized in space. Denote by B^T the discretization of the gradient operator, B the discretization of the divergence operator, D_v the discretization of the diffusion (viscous) operator and C_v the discretization of the convection operator. We use superscript $^{n+1}$ to denote values at the new time level and superscript ^{n} to denote the old time values, $\tau = t^{n+1} - t^n$ stands for the time step. With these notations, we get following system to be solved for concentrated polymer solutions

$$\begin{cases} B\mathbf{v}^{n+1} = 0 \\ \rho\frac{\mathbf{v}^{n+1}-\mathbf{v}^n}{\tau} = -B^T p^{n+1} + D_v\mathbf{v}^{n+1} - C_v\mathbf{v}^{n+1} + B\mathbf{T}^n \\ \frac{\mathbf{E}^{n+1}-\mathbf{E}^n}{\tau} = (B^T\mathbf{v}^{n+1})^T\mathbf{E}^n \\ \textit{approximate} \qquad \hat{\mathbf{Q}}^{n+1}(\mathbf{E}^{n+1}) \\ \mathbf{T}^{n+1} = G_e\int_{-\infty}^{t^{n+1}} dt'\mu(t^{n+1},t')\hat{\mathbf{Q}}(t^{n+1},t'), \end{cases}$$

or in the case of dilute polymer solutions

$$\begin{cases} B\mathbf{v}^{n+1} = 0 \\ \rho\frac{\mathbf{v}^{n+1}-\mathbf{v}^n}{\tau} = -B^T p^{n+1} + D_v\mathbf{v}^{n+1} - C_v\mathbf{v}^{n+1} + B\mathbf{T}^n \\ \frac{\mathbf{E}^{n+1}-\mathbf{E}^n}{\tau} = (B^T\mathbf{v}^{n+1})^T\mathbf{E}^n \\ \mathbf{B}^{n+1} = \mathbf{E}^{n+1}\cdot(\mathbf{E}^{n+1})^T \\ \mathbf{T}^{n+1} = \frac{\eta_p}{\tau_{Oldr}^2}\int_{-\infty}^{t^{n+1}} dt'\mu(t^{n+1},t')\mathbf{B}(t^{n+1},t'). \end{cases}$$

Projection type, for example SIMPLE-like (see, e.g., [15]), or coupled algorithms are applied for solving the balance equations at Step 1. In fact, the current algorithm and software are extension of [23], where solution of the Newtonian flow problems are considered. The more interesting part of the algorithm is the non-Newtonian part, i.e. Step 2. Forward Euler scheme for the **E** deformation tensor is shown above, second order Runge-Kutta method is also used for solving its evolution equation (2.33). To calculate extra stress tensor of the general integral–type (2.35), two known methods, namely Backward Lagrangian Particle Method (BLPM) and Deformation Field Method (DFM), are used at this point of our algorithm. A detailed discussion of those methods will be presented and discussed in the forthcoming Chapter 4.

3.2 Projection type methods.

Again, we assume that the continuity (2.9) and the momentum (2.36) equations are discretized in space. Using the same notations as in previous Section we rewrite them in the following form

$$\begin{aligned} &\tfrac{\rho\mathbf{v}^{n+1}}{\tau} + (C_v - D_v)\mathbf{v}^{n+1} + B^T p^{n+1} = \tfrac{\rho\mathbf{v}^{n}}{\tau} + S_v \\ &B\mathbf{v}^{n+1} = 0, \end{aligned} \tag{3.1}$$

where $\mathbf{v} = (v_1, v_2, v_3)$ denotes velocity, p is pressure, ρ the density and S_v represents the body forces like gravity and also includes divergence of extra stress tensor $\mathbf{T}$. The system of equations (2.9, 2.36), that has to be solved at each time step $t = t^{n+1}$, may be written in a matrix form

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}, \tag{3.2}$$

where $f = \frac{\rho\mathbf{v}^n}{\tau} + S_v$ and $A = (\frac{\rho}{\tau}I + C_v - D_v)$. Projection type methods, also called pressure correction methods, or fractional step schemes, are discussed in this Section. Instead of solving simultaneously the velocity an pressure components, the coupled problem (3.2) is divided into simpler equations of the Poisson and the convection-diffusion type. One way to derive such methods for incompressible fluids, using the formalism of Turek [43], is following. Solving the first equation with respect to velocity gives

$$\mathbf{v} = A^{-1}(f - B^T p). \tag{3.3}$$

Substituting into the second one gives an equation for pressure

$$BA^{-1}B^T p = BA^{-1} f. \tag{3.4}$$

Direct solution of above system is not possible since A^{-1} is a full matrix. Therefore an iterative procedure, preconditioned Richardson iteration, has to be applied. It can be expressed as

$$p^{i+1} = p^i - M^{-1}(BA^{-1}B^T p^i - BA^{-1} f). \tag{3.5}$$

Let $\delta p^i = p^{i+1} - p^i$ denote pressure correction at the $i-$th iteration. The above equation is rewritten as

$$M\delta p^i = -(BA^{-1}B^T p^i - BA^{-1} f) = -B(A^{-1}B^T p^i - A^{-1} f) = -B\mathbf{v}^i, \tag{3.6}$$

where $\mathbf{v}^i$ denotes solution of equation (3.3) for given p^i. The preconditioner M should be spectrally close to $BA^{-1}B^T$ and the usual choice of it is

$$M = BH^{-1}B^T, \tag{3.7}$$

where H is a diagonal matrix. Many possible choices were proposed, like $H = diag\{A\}$, which results in well known SIMPLE scheme. Modifications, like $H = diag(part\{A\})$, give different versions of the original SIMPLE algorithm (fractional time step projection method). After solving equation (3.6) new pressure field is calculated through its correction δp^i. However, velocity $\mathbf{v}^i$ will not in general satisfy incompressibility constraint. Therefore, some correction has to be introduced to project it out on divergence free space. It is realized by taking

$$\mathbf{v}^{i+1} = \mathbf{v}^i + H^{-1}B^T\delta p^i. \tag{3.8}$$

As it is discussed by Turek in [43], the projection type methods are preferable in engineering and commercial codes since in many applications, especially for high Reynolds numbers, they seem to provide a fast and robust solutions. These schemes work perfectly for non steady problems with small viscosities (what assures large Reynolds numbers), which require small time step by physical reason. However, they fail often for steady or non steady flows with large viscosity parameter. Moreover, Turek discuss incompressible Newtonian fluids with constant viscosity, for which the matrix A takes a diagonal form

$$A = \begin{pmatrix} A_{11} & 0 & 0 \\ 0 & A_{22} & 0 \\ 0 & 0 & A_{33} \end{pmatrix}. \tag{3.9}$$

Contrary, this thesis deals with the generalized Newtonian fluids. Due to the varying viscosity, matrix A has a full block representation of the form

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix}. \tag{3.10}$$

It shows, that in the case of strong off-diagonal blocks the approximation of A with an diagonal matrix H may not be good enough for a good performance of projection schemes. Moreover, it brings additional coupling of the balance equations. Now, in addition to the coupling of the continuity (2.9) with the momentum (2.36) equations, and the coupling of the latter through the convective term, equations (2.36) are strongly coupled through their viscous term.

3.2.1 Coupled momentum projection algorithm.

The SIMPLE algorithm (Semi-Implicit Method for Pressure Linked Equations) , proposed by Patankar and Spalding [33], is an example of the projection type methods.

This algorithm is widely used presently for solving incompressible Newtonian flows. One way to make an extension to the generalized Newtonian flows, where the matrix A takes the full block representation (3.10), is to discretize the cross-derivative terms in an explicit way. Such approach is very often done in commercial solvers. However, in many cases, especially for large and strongly varying viscosity problems, such treatment of the mixed derivatives may lead to instabilities (what will be shown in the numerical experiments in Chapter 6). One way to overcome such instabilities is to decrease the time step τ used in the simulations. However, this lead to the increase of the computational time, therefore the modified projection type algorithm, which betters the stability of iterative process, is proposed in this Section. We restrict our considerations to 2D case at this point. At each step of an iterative process one has to solve

$$\begin{pmatrix} A_{11}(\mathbf{v}) & A_{12}(\mathbf{v}) & B_1^T \\ A_{21}(\mathbf{v}) & A_{22}(\mathbf{v}) & B_2^T \\ B_1 & B_2 & 0 \end{pmatrix} \begin{pmatrix} v_1^{k+1} \\ v_2^{k+1} \\ p^{k+1}. \end{pmatrix} = \begin{pmatrix} F_1 \\ F_2 \\ 0 \end{pmatrix}, \tag{3.11}$$

where $k+1$ stands for the number of the iteration to be performed. A_{11}, A_{22} denote linear operators, including time, diffusive (without mixed derivatives) and convective terms of the discretized momentum equations. A_{12}, A_{21} denote linear operators including mixed derivatives from the diffusive term. As it was already mentioned, clearly for the Newtonian flows $A_{12} = A_{21} = 0$. $[B_1, B_2]^T$ and $[B_1, B_2]$ stand for the discrete gradient and divergence operators respectively. Now, we will briefly describe coupled momentum projection algorithm following the formalism of Fletcher (see [15]). Each iteration of this method contains the following steps:

(1) $\mathbf{v}^0 = \mathbf{v}^n$, $p^0 = p^n$.
(2) Compute v_1^{*k+1}, v_2^{*k+1} from

$$\begin{pmatrix} A_{11}(\mathbf{v}) & A_{12}(\mathbf{v}) \\ A_{21}(\mathbf{v}) & A_{22}(\mathbf{v}) \end{pmatrix} \begin{pmatrix} v_1^{k+1} \\ v_2^{k+1} \end{pmatrix} = \begin{pmatrix} F_1 \\ F_2 \end{pmatrix}. \tag{3.12}$$

(3) Solve a Poisson-type equation for the pressure correction p';
(4) Calculate velocity corrections $\mathbf{v}' = -H^{-1}B^T p'$;
(5) Correct the velocities $\mathbf{v}^{k+1} = \mathbf{v}^{*k+1} + \alpha_v \mathbf{v}'$ and the pressure $p^{k+1} = p^{*} + \alpha_p p'$.
(6) $\mathbf{v}^{n+1} = \mathbf{v}^{k+1}$, $p^{n+1} = p^{k+1}$.

At the first step, the momentum equations are solved assuming the pressure is known from the preceding iteration. Thus one obtains an initial approximation $\mathbf{v}^*$ for the velocity vector. At the next step, the pressure correction equation (PCE) is formed aiming at satisfying the continuity equation. More precisely, one defines some velocity corrections through $\vec{\mathbf{v}}' = -H^{-1}B^T p'$, and obtains PCE taking discrete divergence from $\mathbf{v}^{k+1} = \mathbf{v}^{*k+1} + \mathbf{v}'$, i.e. $\mathbf{v}^{k+1} = \mathbf{v}^{*k+1} - H^{-1}B^T p'$, where

$$H = diag\{\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}\} \tag{3.13}$$

The obtained PCE looks as follows (see for details [13, 15])

$$B\mathbf{v}^{k+1} = -BH^{-1}B^{T}p' + B\mathbf{v}^{*k+1} = 0. \tag{3.14}$$

The expression $B\mathbf{v}^{*k+1}$ is the so called mass source term. In the case of collocated grids special interpolation schemes (for details see [35]) for the cell face velocities must be used in order to avoid non-physical oscillations (it is discussed in Section 3.4.3). Underrelaxation parameters α_v, α_p are used to improve the convergence of the algorithm (see [13, 23, 33] for details). Note, that modification of the standard projection type methods, like SIMPLE, is done at point (3.12) of the above algorithm. The momentum equations are solved coupled. It is preferable for us due to stability reasons in simulations of the generalized Newtonian fluids with varying viscosity.

3.3 Fully coupled method.

Many viscoelastic fluids have large viscosities. Moreover, as indicated in Section 2.3, contrary to the Newtonian fluids, the viscoelastic materials may also enjoy big variations in shear, or elongational viscosities. This further influences the strength of the off-diagonal blocks of matrix A (given by equation (3.10)). For such cases the use of projection type methods, presented in the previous Section, might not be enough to perform stable calculations. On the other hand, adding elasticity of the fluid, modeled by the integral constitutive equation (2.35), where one has to calculate time integrals, gives restrictions on time step τ chosen in calculation. Too small time step leads to huge memory requirements associated with the integral and also increases significantly the computational time. Therefore, increasing stability of the solver by decreasing the time step τ, what is possible in general, is not recommended in simulations of viscoelastic fluids. To overcome these problems, i.e. to better the stability of the solver and keep both the computational time and the memory needs in reasonable orders, a fully coupled solver has to be used. It should be mentioned at this point, that the fully coupled solvers require much more memory, compered to the segregated ones. Now, the system (3.2) is solved simultaneously for both velocity and pressure components. There are different fully coupled solvers. Some of them work with untransformed system (3.2), others first transform the system (3.2) and after that solve. There are some observations (see, e.g.[43]) that the fully coupled solvers are preferable for steady state solutions and the decoupling techniques being preferable for unsteady problems. However, very few particular cases are completely analyzed and further studies have to be performed in order to confirm or reject above statement. Detailed discussion and further references concerning coupled solvers can be found, for example, in [4, 42]. To speed up calculations, the system (3.2) is supplied with some appropriate preconditioner. The detailed discussion about different preconditioning strategies, for both untransformed and transformed system (3.2), is postponed and will be performed in Chapter 5.

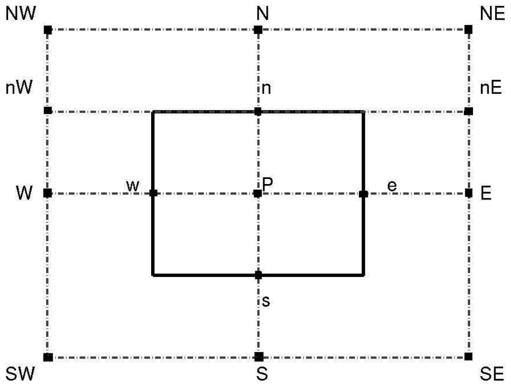

Figure 3.1: *A representative CV.*

3.4 Finite Volume discretization.

Let us now discuss the space discretization used. For simplicity, we restrict ourself to 2D case for all considerations in this Chapter. The extension to 3D case is straightforward. Let us first rewrite equations (2.9, 2.36) in a component–wise manner

$$\frac{\partial}{\partial x_j}(v_j) = 0 \tag{3.15}$$

$$\rho\frac{\partial v_i}{\partial t} + \frac{\partial}{\partial x_j}(\rho v_j v_i) - \frac{\partial}{\partial x_j}(\eta(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i})) = -\frac{\partial p}{\partial x_i} + f_i, \qquad i,j = 1,2. \tag{3.16}$$

Here $\mathbf{v} = (v_1, v_2)^T$ stands for the velocity vector, x_j are Cartesian coordinates, ρ stands for density, p is the pressure, η is the viscosity, that is usually space dependent, and

$$f_i = \frac{\partial}{\partial x_j}T_{i,j}. \tag{3.17}$$

Appropriate boundary conditions complete the system. The summation convention over repeating indices is exploited above. A representative control volume (CV) in an orthogonal grid is drawn on Figure 3.1 - its central node is the point P and the CV index is (i, j). We assume, that the neighbor CVs in a certain space direction have neighbor indices, although this is not obligatory for our consideration. The capital letters W, E, S, N designate the centers of the neighbor CVs in WEST, EAST, SOUTH, and NORTH directions correspondingly. The same (but lower case) letters mark the centers of the CV walls in each direction. NW, NE (SW, SE) stands for the centers of the NORTH (SOUTH) neighbors to the WEST and EAST neighbors of the CV respectively. The letters nW, nE stand for the centers of the north walls of the WEST and EAST neighbors of the CV. The size of the edges of our representative CV are δx and δy: through them one defines the volume δV of the CV, and also the area of the CV faces in each direction (A_w, A_e, A_s, A_n). One also assumes, that the volume integrals are calculated by midpoint cubature, i.e. as a multiplication of the CV nodal value by the volume δV, while the surface integrals are equal to the product of the value in the wall center and the wall area. Thus for the momentum equations (3.16), after applying the Gauss divergence theorem (*Theorem 2* from Section 2.1, page 16), one obtains the following transport equations:

$$\begin{array}{l}
\frac{\partial(\rho v_1)}{\partial t}\delta V+ \\
\Bigg\{\Big[\rho_e v_{1e}(\delta y\ )_e v_{1e} - 2\eta_e\Big(\frac{\partial v_1}{\partial x}\Big)_e(\delta y\ )_e\Big] - \Big[\rho_w v_{1w}(\delta y\ )_w v_{1w} - 2\eta_w\Big(\frac{\partial v_1}{\partial x}\Big)_w(\delta y\ )_w\Big]\Bigg\}+ \\
\Bigg\{\Big[\rho_n v_{2n}(\delta x\ )_n v_{1n} - \eta_n\Big(\frac{\partial v_1}{\partial y}\Big)_n(\delta x\ )_n\Big] - \Big[\rho_s v_{2s}(\delta x\ )_s v_{1s} - \eta_s\Big(\frac{\partial v_1}{\partial y}\Big)_s(\delta x\ )_s\Big]\Bigg\}+ \\
\Bigg\{\eta_n\Big(\frac{\partial v_2}{\partial x}\Big)_n(\delta x\ )_n\Big] - \eta_s\Big(\frac{\partial v_2}{\partial x}\Big)_s(\delta x\ )_s\Big]\Bigg\}+ \\
= S_1\delta V. \\
\frac{\partial(\rho v_2)}{\partial t}\delta V+ \\
\Bigg\{\Big[\rho_e v_{1e}(\delta y\ )_e v_{2e} - \eta_e\Big(\frac{\partial v_2}{\partial x}\Big)_e(\delta y\ )_e\Big] - \Big[\rho_w v_{1w}(\delta y\ )_w v_{2w} - \eta_w\Big(\frac{\partial v_2}{\partial x}\Big)_w(\delta y\ )_w\Big]\Bigg\}+ \\
\Bigg\{\Big[\rho_n v_{2n}(\delta x\ )_n v_{2n} - 2\eta_n\Big(\frac{\partial v_2}{\partial y}\Big)_n(\delta x\ )_n\Big] - \Big[\rho_s v_{2s}(\delta x\ )_s v_{2s} - 2\eta_s\Big(\frac{\partial v_2}{\partial y}\Big)_s(\delta x\ )_s\Big]\Bigg\}+ \\
\Bigg\{\eta_e\Big(\frac{\partial v_1}{\partial y}\Big)_e(\delta y\ )_e\Big] - \eta_w\Big(\frac{\partial v_1}{\partial y}\Big)_w(\delta y\ )_w\Big]\Bigg\}+ \\
= S_2\delta V.
\end{array} \tag{3.18}$$

In the above formulas η (the dynamic viscosity) is a space dependent variable, and $S_1 = -\frac{\partial p}{\partial x_1} + f_1$,$S_2 = -\frac{\partial p}{\partial x_2} + f_2$. Let us first discuss the discretization of the viscous and convective operators appearing in the equation (3.18). The linearization of the momentum equation is of a fixed point type, where the convective terms are written in an equivalent non-divergent form, i.e. $\rho v_j^\star \frac{\partial v_i}{\partial x_j}$ and the velocity components $v_j^\star$ are considered to be known from the previous iteration.

The remaining part of discretization of the system consisting of the continuity (3.15) and the momentum (3.16) equations is discussed later. For projection type methods, this remaining part is the pressure correction equation (for details see Section 3.2) and its discretization is shown in Section 3.4.3. For fully coupled methods, the remaining part consist of discretization of gradient and divergence operators and is discussed in Section 3.4.4.

3.4.1 Discretization of the momentum equations.

The momentum equations (3.16) after an implicit discretization will have the following general form

$$\begin{array}{l}
a_{1P}\phi_{1P} + \sum a_{1nghb}\phi_{1nghb} + \sum a_{2nghb}\phi_{2nghb} = \\
a_{1P}\phi_{1P} + a_{1W}\phi_{1W} + a_{1E}\phi_{1E} + a_{1S}\phi_{1S} + a_{1N}\phi_{1N} \\
+a_{2NW}\phi_{2NW} + a_{2NE}\phi_{2NE} + a_{2W}\phi_{2W} + a_{2E}\phi_{2E}+ \\
a_{2SW}\phi_{2SW} + a_{2SE}\phi_{2SE} + a_{2N}\phi_{2N} + a_{2S}\phi_{2S} = S_P.
\end{array} \tag{3.19}$$

The variables (ϕ_1, ϕ_2) are the unknowns (ϕ_{1P}, ϕ_{1nghb}, ϕ_{2nghb}) at the CV nodal point P and at the nodes of the neighbors of this CV (see Figure 3.1) with the corresponding coefficients a_{1P}, a_{1nghb}, a_{2nghb}. S_P is the point source/sink term for the variable ϕ_1, located at the CV node. The variables ϕ_2 corresponds to the mixed derivatives appearing in (3.18). It is clear that in the first equation of (3.18) ϕ_1 corresponds to v_1 and ϕ_2 corresponds to v_2. In the second equation of (3.18) it is the other way around. The system of linear equations obtained is solved most often by some iterative solver. The size of the system is equal to the dimension number of the considered problem times the number of CV nodes - because one writes such equation for each CV. Further, the values at each internal wall are obtained through a linear interpolation of the neighbor node values, what gives

$$\phi_{1e} = (1-f)\phi_{1P} + f\phi_{1E}, \quad f = \tfrac{x_e - x_P}{x_E - x_P}. \tag{3.20}$$

Let us give the discretization of the convective and diffusive terms for the east ($_e$) wall for a regular orthogonal grids. Let us start with discretization of the convective terms. We consider first order upwind differencing scheme (UDS). Other types of discretization, i.e. higher order schemes, are also implemented but are not discussed here and may, for example, be found in [15]. The convective term associated with the east wall is $(\rho\ v_{1e}\ A_e)\phi_{1e}$. If the CV wall is internal, i.e. the neighbor CV exists, the convective term produces contributions to a_{1P} or a_{1E}

$$\phi_e = \begin{cases} \phi_{1P} & \text{if } v_{1e} > 0 \quad \Rightarrow a_{1P} \longleftarrow (\rho\ v_{1e}\ A_e) \text{ and } a_{1E} \longleftarrow 0 \\ \phi_{1E} & \text{if } v_{1e} < 0 \quad \Rightarrow a_{1P} \longleftarrow 0 \text{ and } a_{1E} \longleftarrow (\rho\ v_{1e}\ A_e) \end{cases}. \tag{3.21}$$

In the case of a Dirichlet boundary wall the value ϕ_{1e} is known from the boundary condition and the contribution is to the free term S_P only: $S_P \longleftarrow (-\rho\ v_{1e}\ A_e)\phi_{1e}$. Second order central difference scheme (CDS) is always used for discretization of the diffusive terms. In case of the internal wall derivatives of the type $\left(\frac{\partial \phi_1}{\partial x}\right)_e$ are approximated as

$$\left(\frac{\partial \phi_1}{\partial x}\right)_e \approx \frac{\phi_{1E} - \phi_{1P}}{x_E - x_P}, \tag{3.22}$$

producing contributions to $a_{1P} \longleftarrow \eta_e\left(\frac{1}{x_E - x_P}\right)A_e$ and to $a_{1E} \longleftarrow -\eta_e\left(\frac{1}{x_E - x_P}\right)A_e$ coefficients. If the east CV wall is a boundary one, first order of approximation is used

$$\left(\frac{\partial \phi_1}{\partial x}\right)_e \approx \frac{\phi_{1e} - \phi_{1P}}{x_e - x_P}. \tag{3.23}$$

Then again one has a contribution to $a_{1P} \longleftarrow \eta_e\left(\frac{1}{x_E - x_P}\right)A_e$ and to $S_P \longleftarrow \eta_e\left(\frac{\phi_{1e}}{x_e - x_P}\right)A_e$ for Dirichlet type of BCs, where ϕ_{1e} is the known boundary value. For Neumann types of BCs the contribution is to the coefficient, associated with the additional unknown in the boundary wall center, instead to S_P directly. In fact, the normal derivative at the east wall is approximated by formula (3.23), thus knowing its value, say ϕ_{bound}, and the value at the center of CV, one can obtain ϕ_{1e} from

$$\phi_{1e} \approx \phi_{1P} + (x_e - x_P)\phi_{bound}. \tag{3.24}$$

Then the contributions looks the same as in the case of Dirichlet BCs, however now, the contribution to the free term S_P uses the wall values calculated from the previous time step iteration.

3.4.2 Discretization of the mixed derivatives.

Second order scheme is also used for the mixed derivative terms. The symmetric discretization, presented in the book of Wesseling (see, [50] and references therein), is chosen. It assures, that when the solution has a certain symmetry, the discrete approximation will also preserve this symmetry. The approximations of $\left(\frac{\partial \phi_2}{\partial x}\right)_n$, that appears in first equation of the system (3.18), is performed in the following manner. In the case of the internal wall

$$\left(\frac{\partial \phi_2}{\partial x}\right)_n \approx \frac{1}{2}\left(\frac{\phi_{2NE} - \phi_{2NW}}{x_{NE} - x_{NW}} + \frac{\phi_{2E} - \phi_{2W}}{x_E - x_W}\right). \tag{3.25}$$

Then the contributions to $a_{2NE}, a_{2NW}, a_{2E}, a_{2W}$ take the following form:

$a_{2NE} \longleftarrow \eta_n\left(\frac{1}{x_{NE}-x_{NW}}\right)A_n$, $a_{2NW} \longleftarrow -\eta_n\left(\frac{1}{x_{NE}-x_{NW}}\right)A_n$,
$a_{2E} \longleftarrow \eta_n\left(\frac{1}{x_E-x_W}\right)A_n$ and $a_{2W} \longleftarrow -\eta_n\left(\frac{1}{x_E-x_W}\right)A_n$.

If the north wall is a boundary one, the discretization looks as follows

$$\left(\frac{\partial \phi_2}{\partial x}\right)_n \approx \frac{1}{2}\left(\frac{\phi_{2nE} - \phi_{2nW}}{x_{nE} - x_{nW}} + \frac{\phi_{2E} - \phi_{2W}}{x_E - x_W}\right). \tag{3.26}$$

We treat the boundary conditions in the same way as in other diffusive terms. Again we have contributions to coefficients a_{2E}, a_{2W} and S_p. For a Dirichlet boundary wall, when ϕ_{2nW} and ϕ_{2nE} are known from the boundary condition, S_p takes the following form

$$S_P \longleftarrow \eta_n\left(\frac{\phi_{2nE}}{x_{NE} - x_{NW}}\right)A_n - \eta_n\left(\frac{\phi_{2nW}}{x_{NE} - x_{NW}}\right)A_n. \tag{3.27}$$

Of course one can discretize the mixed derivatives explicit in time. In this case the contribution to S_p is $S_p \longleftarrow \left(\frac{\partial \phi_2}{\partial x}\right)_n$. However, explicit treatment of those terms may lead to the stability problems for certain class of problems. As it is discussed in [41], in constructing unconditionally stable difference schemes it is necessary to design the schemes, where terms with viscous stresses in one way or another are referred to the upper time–level. Therefore implicit treatment is recommended.

3.4.3 Discretization of the pressure correction equation (PCE).

For projection type methods, presented in Section 3.2, after the solution of momentum equations, the velocity $\mathbf{v}^*$ does not satisfy the continuity equation. Therefore, using this equation one searches corrections of velocity in order the latter to fulfill mass balance. Let us write the corrected velocities as

$$\mathbf{v} = \mathbf{v}^* + \mathbf{v}^{'}, \tag{3.28}$$

where $\mathbf{v}^*$ is the solution of the momentum equations on the current step and $\mathbf{v}^{'}$ is velocity correction that has to be found. Let us discuss derivation of PCE on the example of SIMPLE algorithm. The main assumption on this algorithm is that velocity and pressure corrections are related by

$$v^{'}_{1P} = -\frac{1}{a_{1P}}(p^{'}_e - p^{'}_w)\delta y. \tag{3.29}$$

However, after solution of PCE we obtain the pressure corrections at the CV nodes, but not at the walls, as (3.29) requires. Let us then consider an imaginary CV, having a node at the wall e of basic CV. The east and west walls of this imaginary CV fall then on the points P and E. If we apply the same consideration for this imaginary CV, we obtain the same relation, but written in a way we need it

$$v^{'}_{1e} = -\tfrac{1}{\bar{a}^{v_1}_{P,e}}(p^{'}_E - p^{'}_P)\delta y, \quad \bar{a}^{v_1}_{P,e} = (1-f)a^{v_1}_P + f a^{v_1}_{P,E}, \quad f = \tfrac{x_e - x_P}{x_E - x_P}\ . \tag{3.30}$$

In the latter formula, the coefficient $\bar{a}^{v_1}_P$ of the imaginary CV is linearly approximated by (3.20) using the values a_P of the "representative" CVs with nodes P and E. The upper script ^{v_1} is used to distinguish from which equation of discretized momentum equations a_P comes from. The PCE is derived from the continuity equation (3.15), the expression for the velocity corrections (3.28), and the relations (3.30) between the velocity corrections and the pressure correction. One starts by integrating the continuity equation (3.15) within the representative CV (its volume is $\delta V = \delta x \delta y$)

$$\int\int_{CV} \frac{\partial v_1}{\partial x} dxdy + \int\int_{CV} \frac{\partial v_2}{\partial y} dxdy = 0.$$

Applying the Gauss divergence theorem (*Theorem 2* from Section 2.1, page 16) to the integrals in the left-hand side of the equation above, one obtains

$$\Big(v_{1e}\delta y - v_{1w}\delta y\Big) + \Big(v_{2n}\delta x - v_{2s}\delta x\Big) = 0.$$

The latter relation is valid both for the continuous variables and for the grid-function of the numerical solution for the velocities. Therefore, one can express the velocity components there through their initial approximations and their further corrections, according to (3.28). We get

$$\begin{aligned}&\left(v^{'}_{1e}\delta y - v^{'}_{1w}\delta y\right) + \left(v^{'}_{2n}\delta x - v^{'}_{2s}\delta x\right) = \\ &-\left\{\left(v^{*}_{1e}\delta y - v^{*}_{1w}\delta y\right) + \left(v^{*}_{2n}\delta x - v^{*}_{2s}\delta x\right)\right\}.\end{aligned} \tag{3.31}$$

The expression closed in the contour brackets in the right–hand side of (3.31) is the mass source term. This is the mass disballance within the CV for the velocity approximation, that we have obtained after the solution of the momentum equations. On the other side, the continuity equation (3.15) requires this mass disballance to be zero. Thus, we look for such velocity corrections, which make this mass disballance to vanish. We will further consider in details the calculation of the mass source term. In the following equation we just denote it by MST. The equation itself is obtained from (3.31) by replacing in its left-hand side the velocity corrections by the expressions relating them to the pressure corrections, like (3.30). Thus we obtain

$$\begin{aligned}&\left[-\frac{1}{\bar{a}^{v_1}_{P,e}}(p^{'}_E - p^{'}_P)(\delta y)^2 + \frac{1}{\bar{a}^{v_1}_{P,w}}(p^{'}_P - p^{'}_W)(\delta y)^2\right] + \\ &\left[-\frac{1}{\bar{a}^{v_2}_{P,n}}(p^{'}_N - p^{'}_P)(\delta x)^2 + \frac{1}{\bar{a}^{v_2}_{P,s}}(p^{'}_P - p^{'}_S)(\delta x)^2\right] = -MST.\end{aligned}$$

This equation gives the discretization coefficients for the PCE, which are

$$\begin{aligned}&a_E = -\frac{1}{\bar{a}^{v_1}_{P,e}}(\delta y)^2,\ a_W = -\frac{1}{\bar{a}^{v_1}_{P,w}}(\delta y)^2, \\ &a_N = -\frac{1}{\bar{a}^{v_2}_{P,n}}(\delta x)^2,\ a_S = -\frac{1}{\bar{a}^{v_2}_{P,s}}(\delta x)^2, \\ &a_P = -(a_E + a_W + a_N + a_S).\end{aligned} \tag{3.32}$$

All averaged coefficients in (3.32), coming from the discretization of the momentum equations, are calculated through linear interpolation similar to the one in (3.30). One sees from (3.32) that PCE is a Poisson type equation. We need also to add to the PCE coefficients (3.32) the free term, which is the mass source term taken with a negative sign (see (3.31)). As it was mentioned earlier, the values at the walls are linearly interpolated through the nodal values (see (3.20)). So, the wall values for the velocities, used in (3.31) to calculate the mass source term, are also linearly interpolated. Further in (3.32) we linearly interpolate the values of the a_P coefficients from the momentum equations. If the pressure and the velocity calculations are performed on the same grid, this kind of interpolation creates non-physical oscillations (see e.g. [33]). To overcome the problem, a special kind of interpolation for the wall velocities ([35]) in the mass source term has to be applied. In order to find such cell-face velocity approximations, which - after putting them in the right - hand side of (3.31) - do not cause non-physical oscillations, the requested interpolation should take into account not only the velocity values in the neighboring nodes, but the pressure also. We will derive the expression for the cell-face velocity v_{1e} on the east wall (it is "starred" in (3.31), here we skip the star). We consider the representative CV with a point P as a center and its east neighbor denoted by E. We use our usual notation, only adding the corresponding superscripts ^{P} or ^{E} to designate the CV the corresponding quantity belongs to. Let us start the derivation from the v_1−momentum equations in these CVs, but writing separately in the expressions for the source terms S_P^P and S_P^E those parts, which contain the pressure:

$$v_{1P} = v_P^{1P} = \frac{1}{a_P^P}\Bigg[-\sum a_{nghb}^P v_{1nghb}^P + F_1^P - (p_e^P - p_w^P)\delta y\Bigg], \tag{3.33}$$

$$v_{1E} = v_P^{1E} = \frac{1}{a_P^E}\Bigg[-\sum a_{nghb}^E v_{1nghb}^E + F_1^E - (p_e^E - p_w^E)\delta y\Bigg]. \tag{3.34}$$

We also know that v_{1e} is a linear interpolation (see (3.20)) from v_{1P}^P and v_{1P}^E, i.e.:

$$v_{1e} = (1-f)v_{1P}^P + f v_{1P}^E, \quad f = \tfrac{x_e - x_P}{x_E - x_P}. \tag{3.35}$$

By inserting (3.33) and (3.34) in (3.35), one obtains:

$$\begin{aligned} v_{1e} = (1-f)\Bigg\{\tfrac{1}{a_P^P}\Bigg[-\sum a_{nghb}^P v_{1nghb}^P + F_1^P\Bigg] - \tfrac{\delta y}{a_P^P}(p_e^P - p_w^P)\Bigg\} + \\ f\Bigg\{\tfrac{1}{a_P^E}\Bigg[-\sum a_{nghb}^E v_{1nghb}^E + F_1^E\Bigg] - \tfrac{\delta y}{a_P^E}(p_e^E - p_w^E)\Bigg\}. \end{aligned}$$

Let us write the last equation as

$$\begin{aligned} v_{1e} = (1-f)&\Bigg\{\tfrac{1}{a_P^P}\Bigg[-\sum a_{nghb}^P v_{1nghb}^P + F_1^P\Bigg]\Bigg\} \\ &+f\Bigg\{\tfrac{1}{a_P^E}\Bigg[-\sum a_{nghb}^E v_{1nghb}^E + F_1^E\Bigg]\Bigg\} \\ &-\delta y\Bigg[(1-f)\tfrac{1}{a_P^P}(p_e^P - p_w^P) + f\tfrac{1}{a_P^E}(p_e^E - p_w^E)\Bigg]. \end{aligned} \tag{3.36}$$

Note, that now the last expression in the right–hand side of (3.36) is the linear interpolation averaging (taken at the point e) of the quotient of the pressure difference and the a_P coefficient, i.e

$$(1-f)\frac{1}{a_P^P}(p_e^P - p_w^P) + f\frac{1}{a_P^E}(p_e^E - p_w^E) = \overline{\left(\frac{p_e - p_w}{a_P}\right)_e}.$$

Now, taking the ideas from the staggered grid approach (see [35]), the pressure difference at the east face is replaced by the difference of the pressures taken from the neighboring cell centers, i.e. $\overline{(p_e - p_w)_e} = p_E - p_P$. One can then write the last term in (3.36) as

$$\begin{aligned} -\delta y\Bigg[(1-f)\tfrac{1}{a_P^P}(p_e^P - p_w^P) + f\tfrac{1}{a_P^E}(p_e^E - p_w^E)\Bigg] = \\ -\delta y(p_E - p_P)\overline{\left(\tfrac{1}{a_P}\right)_e} = -\delta y(p_E - p_P)\Bigg[(1-f)\tfrac{1}{a_P^P} + f\tfrac{1}{a_P^E}\Bigg]. \end{aligned} \tag{3.37}$$

In a similar way we transform the first two terms in the right–hand side of (3.36). We have

$$\begin{aligned}&\tfrac{1}{a_P^P}\left[-\sum a^P_{nghb}v^P_{1nghb}+F_1^P\right]=\\&\tfrac{1}{a_P^P}\Big[\underbrace{-\sum a^P_{nghb}v^P_{1nghb}+F_1^P-(p_e^P-p_w^P)\delta y-a_P^P v_{1P}}_{=0,\text{ from (3.33)}}+(p_e^P-p_w^P)\delta y+a_P^P v_{1P}\Big].\end{aligned}$$

So, for the first term in the right–hand side of (3.36), we finally get

$$\frac{1}{a_P^P}\left[-\sum a^P_{nghb}v^P_{1nghb}+F_1^P\right]=\left[v_{1P}^P+\frac{\delta y}{a_P^P}(p_e^P-p_w^P)\right]=\left[v_{1P}+\frac{\delta y}{a_P^P}(p_e^P-p_w^P)\right]. \tag{3.38}$$

For the second term there, following the same sequence of derivations, we write

$$\frac{1}{a_P^E}\left[-\sum a^E_{nghb}v^E_{1nghb}+F_1^E\right]=\left[v_{1P}^E+\frac{\delta y}{a_P^E}(p_e^E-p_w^E)\right]=\left[v_{1E}+\frac{\delta y}{a_P^E}(p_e^E-p_w^E)\right]. \tag{3.39}$$

Now, substituting (3.37), (3.38) and (3.39) in (3.36), we get the required approximation for the east cell-face velocity, i.e.

$$\begin{aligned}v_{1e}=(1-f)\left[v_{1P}+\tfrac{\delta y}{a_P^{v_1,P}}(p_e^P-p_w^P)\right]+f\left[v_{1E}+\tfrac{\delta y}{a_P^{v_1,E}}(p_e^E-p_w^E)\right]\\-\delta y(p_E-p_P)\left[(1-f)\tfrac{1}{a_P^{v_1,P}}+f\tfrac{1}{a_P^{v_1,E}}\right],\qquad f=\tfrac{x_e-x_P}{x_E-x_P}.\end{aligned} \tag{3.40}$$

In (3.40) we have added to the coefficient a_P another upper script ^{v_1} in order to designate the discretization of which momentum equation is referred to. Following the same style of derivation, one obtains similar approximations for each of the cell-face velocities.

3.4.4 Discretization of discrete divergence and gradient operators.

In previous Section 3.4.3 we have presented how to avoid non physical oscillations when one uses projection type methods with collocated grids. In fact central difference approximations for discrete gradient B^T and discrete divergent B operators, based on $2h$ and h respectively, are used. Here h denotes general space sizes, so $h=\delta x, h=\delta y$ for discretization in $x-, y-$, directions respectively. For more detailed description see [13]. In the case of the fully coupled solvers the system of the balance equations (3.15, 3.16), represented in the matrix form (3.2), is solved simultaneously for velocity and pressure components. This means that the discretization of A, B^T and B blocks of (3.2) has to be performed. The way the block A is discretized was already presented in the previous Sections, where discretization of convective and diffusive terms, as well as mixed derivatives were discussed. The remaining part is a discretization of B^T and B blocks. Let us denote by B_P^T and B_P local discrete gradient and divergence operators, where subscript $_P$ indicates that operators are applied locally in a CV with center at point P. As illustrated in [13] double sided discretization given by

$$B_P^T = \begin{pmatrix} \frac{p_E - p_W}{2\delta x} \\ \frac{p_N - p_S}{2\delta y} \end{pmatrix} \tag{3.41}$$

and

$$B_P = \begin{pmatrix} \frac{v_{1E} - v_{1W}}{2\delta x}, \frac{v_{1N} - v_{1S}}{2\delta y} \end{pmatrix} \tag{3.42}$$

may lead to a checkerboard pressure field. This can be avoided by using one sided discretization, forward differences for divergence and backward differences for gradient operators, or the other way around. The latter gives

$$B_P^T = \begin{pmatrix} \frac{p_E - p_P}{\delta x} \\ \frac{p_N - p_P}{\delta y} \end{pmatrix} \tag{3.43}$$

and

$$B_P = \begin{pmatrix} \frac{v_{1P} - v_{1W}}{\delta x}, \frac{v_{2P} - v_{2S}}{\delta y} \end{pmatrix}, \tag{3.44}$$

and is used in our calculations. The weak point of such discretization is that the accuracy of this method is of first order only, what was shown in [16].

Chapter 4

Approximation of the constitutive equation.

In this Chapter, we discuss the approximation techniques of the integral constitutive equation of general type presented in Section 2.4.1. In fact, the use of the integral constitutive equations instead of a differential models has been comparatively rare. The reason for that is certainly not a lack of interest, since often theories of memory fluids yield constitutive equations of integral type (Doi and Edwards [9]), but the numerical complexities associated with them. In [24], a review of different techniques used for solving complex flows of viscoelastic fluids described by the constitutive models of integral type is performed. Here, two methods, also reviewed in [24], are explained: First, a Lagrangian method, namely Backward Lagrangian Particle Method (BLPM), is presented. It combines the computations of Eulerian velocity and pressure fields with the Lagrangian evaluation of the strain history along the particle paths. Second, an alternative method, called Deformation Field Method (DFM), is discussed. In this method, calculation of the extra stress tensor is performed in the Eulerian reference frame. The discussion of a particular discretization of the memory integral, given by equations (2.38, 2.43), in order to decrease the storage requirements and the computational time is presented.

4.1 Backward Lagrangian Particle Method (BLPM).

The theory of viscoelasticity says, that the stress at certain fluid position depends on the history of that fluid element [9]. To handle this a Lagrangian method, the Backward Lagrangian Particle Method (BLPM) [49], has been formulated. The extra stress tensor $\mathbf{T}$, given by the integral constitutive equation (2.38 or 2.43), is calculated along the trajectories of Lagrangian particles that are convected by the flow. Therefore this is the most natural way of calculating such memory integrals. Since a different time levels will be involved in further discussions, before going to the procedure the BLPM follows, let us introduce some nomenclature. Let t denotes the current time of simulation process and t' denotes the past times, i.e. $t' < t$. In BLPM, positions of the particles at current time t are specified a priori, and located at each mesh point, i.e. at the center of each control volume. Then, the particle trajectories leading to these locations are calculated by tracking backward in time. It means that the particle trajectories leading to the centers of control volumes change from one time step to the next, so usually different Lagrangian particles arrive in those grid nodes as time evolves. To describe this process mathematically, let us first denote by $\mathbf{r}(t)$ the initial particle positions from which the backward tracking starts. The particle trajectory is described by the following kinematic equation

$$\frac{D\mathbf{r}}{Dt} = \mathbf{v}(\mathbf{r}, t). \tag{4.1}$$

So, for a given arbitrary particle position $\mathbf{r}(t)$ of the flow domain, at the current time t, the particle position at past times $t' < t$ can be obtained by integration backward in time as

$$\mathbf{r}(t') = \mathbf{r}(t) - \int_{t'}^{t} \mathbf{v}(\mathbf{r}(t), t)dt, \tag{4.2}$$

where $\mathbf{v}(\mathbf{r}(t), t)$ denotes the velocity of particle sampled at $\mathbf{r}(t)$ at time t. Let us write the same procedure in a numerical manner. For that, suppose for a moment, that N_d is the number of backward steps necessary to perform in order to reach the initial particle position (the choice of N_d is discussed in Section 4.5). Next, let $\tau = t^{n+1} - t^n$ denote the time step used in simulations, t^{n+1} will now denote the current time and t^{n-N_d} denote the time at which initial particle position is reached. Note, that since the viscoelastic material do not remember all its deformation history, what was discussed in Section 2.4.1, t^{n-N_d} does not have to be equal to the starting time of simulations $t = 0$. Finally, (4.2) transforms to

$$\mathbf{r}(t^{n-N_d}) = \mathbf{r}(t^{n+1}) - \sum_{i=n-N_d}^{n} \int_{t^i}^{t^{i+1}} \mathbf{v}(\mathbf{r}(t^{i+1}), t^{i+1})dt, \tag{4.3}$$

what corresponds to calculating N_d integrals, where the integral for number $i+1$ uses the velocity $\mathbf{v}(\mathbf{r}(t^{i+1}), t^{i+1})$ taken along the particle trajectory between positions $\mathbf{r}(t^{i+1})$ and $\mathbf{r}(t^i)$. This requires storing all the intermediate N_d velocity fields between the times t^{n-N_d+1} and t^{n+1}. Additionally, at each time step, the array containing of the N_d velocity fields has to be updated in the way, that the oldest velocities are annihilated, the fields 1 ... $N_d - 1$ become fields 2 ... N_d, and the first position is filled with the current velocity field $\mathbf{v}(\mathbf{r}(t^{n+1}), t^{n+1})$. The backward tracking has to be performed for a certain number of the particles in the flow. This number coincides with the number of grid nodes (control volumes). At the initial stages of the flow, when the current time is smaller than the total tracking time τN_d, tracking back is stopped at time $t = 0$. It means that the tracking is performed until reaching the time $t = \min(0, \tau N_d)$. Moreover, it may happen that, due to the numerical errors, the particle leaves the domain boundary. In such cases, since the particle is not allowed to leave the boundary, the tracking back is stopped and one assumes that such particle sticks to the wall at the certain position, and this position is considered as the initial one. Another possibility is, when during the backward tracking inflow is reached. In such cases tracking is stopped and we assume, that new undeformed particles flow into the domain. In both integral constitutive equations, describing the elastic stress of concentrated (2.38) and dilute (2.43) polymer solutions, and presented in Sections 2.4.2 and 2.4.3 respectively, deformation tensor **E** (equation (2.32)) plays a crucial role. In order to approximate memory integrals (2.38 and 2.43), the evolution equation for the deformation tensor **E**, given by the equation (2.33), has to be first solved. Two possible ways to calculate it, in contents of the Lagrangian method, will be now discussed. First, the tracking back is performed the initial particle position $\mathbf{r}(t^{n-N_d})$ is found. One way to calculate **E** is following. After finding the initial position $\mathbf{r}(t^{n-N_d})$, the evolution equation (2.33), which in coordinate–wise manner takes following form (2D description for simplicity)

$$\begin{cases} \frac{DE_{11}(t,t')}{Dt} = \frac{\partial v_1}{\partial x_1}E_{11}(t,t') + \frac{\partial v_1}{\partial x_2}E_{21}(t,t') \\ \frac{DE_{12}(t,t')}{Dt} = \frac{\partial v_1}{\partial x_1}E_{12}(t,t') + \frac{\partial v_1}{\partial x_2}E_{22}(t,t') \\ \frac{DE_{21}(t,t')}{Dt} = \frac{\partial v_2}{\partial x_1}E_{11}(t,t') + \frac{\partial v_2}{\partial x_2}E_{21}(t,t') \\ \frac{DE_{22}(t,t')}{Dt} = \frac{\partial v_2}{\partial x_1}E_{12}(t,t') + \frac{\partial v_2}{\partial x_2}E_{22}(t,t') \end{cases} \tag{4.4}$$

is solved along the particle paths using as a initial condition $\mathbf{E}_{n-N_d}$ taken at $\mathbf{r}(t^{n-N_d})$. The above system of equations is solved using either forward explicit Euler formula or explicit second order Runge-Kutta method. Although we need only the initial condition to be able to calculate deformation **E** tensor, similarly as in backward tracking, we have to store N_d deformation fields, because we need the initial condition at each time step of the simulation. We update those fields in the same way as the velocity fields. The velocity gradients $(\nabla\mathbf{v}(t^i))^T$, that appear in (4.4), can be updated from the velocity fields $\mathbf{v}(t^i)$ or, as in our case, can be treated in the same way as deformation and velocity fields, i.e. calculated at each time step and stored in arrays. This means additional storage, but it safes a lot of time, since the approximation formula needed for all intermediate tracking points (indicated in Figure 4.1) is quite expensive. Let

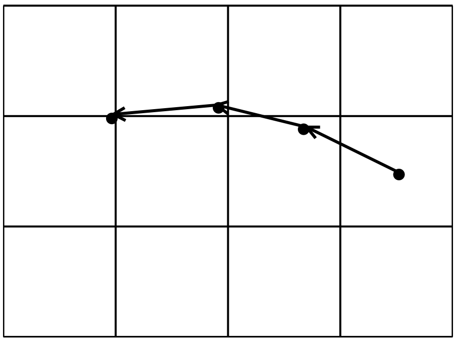

Figure 4.1: *A representative backward particle tracking.*

us now give small modification of the above procedure. In fact, one does not need to calculate the evolution equation (2.33) along all intermediate points of particle path. Since once deformed particle remembers its deformation, it is enough to do the following. First, the evolution equation (2.33) is solved by performing only one step backward. At this point, as an alternative, one can use an equivalent Eulerian form of (2.33)

$$\begin{cases} \frac{\partial E_{11}(x,t,t')}{\partial t} + v_1 \frac{\partial E_{11}(x,t,t')}{\partial x_1} + v_2 \frac{\partial E_{11}(x,t,t')}{\partial x_2} = \frac{\partial v_1}{\partial x_1} E_{11}(x,t,t') + \frac{\partial v_1}{\partial x_2} E_{21}(x,t,t') \\ \frac{\partial E_{12}(x,t,t')}{\partial t} + v_1 \frac{\partial E_{12}(x,t,t')}{\partial x_1} + v_2 \frac{\partial E_{12}(x,t,t')}{\partial x_2} = \frac{\partial v_1}{\partial x_1} E_{12}(x,t,t') + \frac{\partial v_1}{\partial x_2} E_{22}(x,t,t') \\ \frac{\partial E_{21}(x,t,t')}{\partial t} + v_1 \frac{\partial E_{21}(x,t,t')}{\partial x_1} + v_2 \frac{\partial E_{21}(x,t,t')}{\partial x_2} = \frac{\partial v_2}{\partial x_1} E_{11}(x,t,t') + \frac{\partial v_2}{\partial x_2} E_{21}(x,t,t') \\ \frac{\partial E_{22}(x,t,t')}{\partial t} + v_1 \frac{\partial E_{22}(x,t,t')}{\partial x_1} + v_2 \frac{\partial E_{22}(x,t,t')}{\partial x_2} = \frac{\partial v_2}{\partial x_1} E_{12}(x,t,t') + \frac{\partial v_2}{\partial x_2} E_{22}(x,t,t'), \end{cases} \tag{4.5}$$

where x denotes the centers of control volumes. After that backward tracking is performed, as discussed above. Then, forward tracking starts, but despite recalculating the deformation tensor $\mathbf{E}$ at each intermediate point of the particle path, one just collects already known values. This will reduce the computational time, since one does not have to approximate the velocity gradients at those intermediate points and the evolution equation (2.33) for the deformation tensor $\mathbf{E}$ is solved once along each particle path between the time levels $t - \tau$ and t, and not $N_d - 1$ times as in the original method. It should be pointed out, that as it was mentioned in [37], and up to the knowledge of author, no one has ever tried to use the technique of recalculating the particle trajectories at each time step. In [47, 48] authors use BLPM to calculate equation (2.33), however the integral (2.43) is solved with use of Deformation Field Method.

4.2 Deformation Field Method (DFM).

The main assumption of the Deformation Field Method (DFM) [36, 37], is that the deformation history exhibited by the particle is described by a finite number of deformation fields which are convected and deformed by the flow. Such assumption leads to the loss of locality of the stress. Here, the locality means that the stress at certain fluid particle depends only on the previous values evaluated at that fluid element. However, DFM removes the need of recalculating the particle paths, what gives a huge advantage in computational time in comparison with the BLPM. As it was mentioned, one introduces tensor fields $\mathbf{E}(x, t, t')$ where the spatial position x is taken into account. Similar as in equation (4.5) x denotes the centers of control volumes. At each instant of time, the deformation field is created and labeled by the time of creation t'. As time evolves this field is transported and deformed by the flow, obeying equation (4.5). All the information about the history of the flow is now stored in the deformation fields $\mathbf{E}(x, t, t')$ at each grid point x. This allows us to calculate the extra stress tensor, given by the general form (2.35), at the current time t and each grid point of the computational domain x, by integrating the weighted contributions of all deformation fields that have been created so far. For more details we refer interested reader to [36, 37]. After calculating the deformation tensors **E**, obeying the evolution equation (2.33), it is obvious how to calculate Finger strain tensor **B** (2.45) appearing in the integral Oldroyd B model (2.43). More attention has to be paid to the partial orientation tensor **Q** (2.37), which describes the orientations of the polymers in concentrated polymer solutions, and appears in the Doi Edwards integral constitutive equation (2.38).

4.3 Calculation of the partial orientation tensor.

As it was already presented Section 2.4.2, the partial orientation tensor $\mathbf{Q}(x,t,t')$ depends on the deformation tensor $\mathbf{E}(x,t,t')$, either calculated with BLPM, or with DFM respectively. Only the meaning of the coordinate x will be slightly different in the both cases. For BLPM it means the points on particle trajectory, where for DFM x corresponds to grid points. To simplify notation, we introduce two functions

$$\mathbf{g}(x,t) = \frac{\mathbf{E}(x,t,t')\mathbf{u}(t')}{\parallel \mathbf{E}(x,t,t')\mathbf{u}(t') \parallel}, \tag{4.6}$$

$$\mathbf{f}(x,t) = \mathbf{E}(x,t,t')\mathbf{u}(t'). \tag{4.7}$$

In this notation $\mathbf{Q}(x,t,t')$ (equation (2.37)) has the following form

$$\mathbf{Q}(x,t,t') = \langle \mathbf{g}(x,t) \otimes \mathbf{f}(x,t) \rangle_0 \frac{1}{\langle \parallel \mathbf{f}(x,t) \parallel \rangle_0}. \tag{4.8}$$

To evaluate the ensemble average over the isotropic state, M vectors (i.e., $\mathbf{u}_j(t'), j = 1, ..., M.$) are distributed such, that they cover the unit circle as homogeneously as possible. From our experience, it is enough to take $M = 40$. In the discrete notation equations (4.6) and (4.7) transforms to

$$\mathbf{g}_j(x,t) = \frac{\mathbf{E}(x,t,t')\mathbf{u}_j(t')}{\parallel \mathbf{E}(x,t,t')\mathbf{u}_j(t') \parallel}, \tag{4.9}$$

$$\mathbf{f}_j(x,t) = \mathbf{E}(x,t,t')\mathbf{u}_j(t'). \tag{4.10}$$

Finally, the partial orientation tensor **Q**, given by equation (4.8), is written in the form, that is used in simulations as

$$\hat{\mathbf{Q}}(x,t,t') \approx \frac{1}{M} \sum_{j=1}^{M} [\mathbf{g}_j(x,t) \otimes \mathbf{f}_j(x,t)] \cdot \frac{1}{\frac{1}{M}\sum_{j=1}^{M} \parallel \mathbf{f}_j(x,t) \parallel}. \tag{4.11}$$

4.4 Approximation of the extra stress tensor.

Since it has already been presented in previous Sections how to calculate the deformation **E** tensor and the partial orientation **Q** tensor, let us now discuss how to perform the discretization of the memory integral. In the approximation, the general integral constitutive equation, given by equation (2.35), is replaced by a sum as

$$\bar{\mathbf{T}}(x,t) = \sum_{k=0}^{\infty} W_k f(\mathbf{E}_k(x_k,t,t'_k)), \tag{4.12}$$

where $f(\mathbf{E}_k(x_k,t,t'_k)) = G_e\hat{\mathbf{Q}}_k(x_k,t,t'_k)$ and $f(\mathbf{E}_k(x_k,t,t'_k)) = \frac{\eta_p}{\tau_{Oldr}^2}\mathbf{B}_k(x_k,t,t'_k)$ in case of concentrated (without the chain stretch) and dilute polymer solutions respectively. The tensor function f is calculated at all past times t'_k and appropriate points x_k. Again, as discussed in previous Section, in the case of the BLPM these are the points of the particle path, while in the DFM x_k coincide with centers of the control volumes, i.e. $x_k = x$. So, the function $f(\mathbf{E}_k(x_k,t,t'_k))$ contribute to the extra stress tensor $\bar{\mathbf{T}}$ at the current time t and all grid points x. Those contributions are weighted by

$$W_k = exp\left(-\frac{(\Delta_k \cdot \tau)}{\tau_{relax}}\right), k = 1,2,.., \tag{4.13}$$

with a general relaxation time denoted by τ_{relax}. In particular, it denotes the Doi Edwards relaxation time $\tau_{relax} = \tau_d$ in equation (2.38) and the relaxation time $\tau_{relax} = \tau_{Oldr}$ from the integral Oldroyd B model (2.43). Time increment Δ_k indicates that non–uniform time integration is also allowed, and a possible choices of such integration will be discussed in next Section. Note, that the weights W_k decrease when k increase, so it is natural to truncate the infinite summation to the value, say N_d. Then, inserting the truncated approximation to (4.12) gives

$$\bar{\mathbf{T}}(x,t) = \sum_{k=0}^{N_d-1} W_k f(\mathbf{E}_k(x_k,t,t'_k)). \tag{4.14}$$

The weights W_k are position independent, so they can be calculated in advance of the simulation, and they have to be calculated only once. Finally, if $f(\mathbf{E}_k(x_k,t,t'_k))$ is bounded and the truncation (4.14) is performed at the time that is several times larger than the relaxation time τ_{relax}, then the error made by truncation should be small since the memory function decays exponentially.

4.5 Non–uniform discretization of the memory integral.

In this Section we will present and discuss the approximation of the integral constitutive equation (4.14) based on a non–uniform time increment. To calculate the approximation of the memory integral (2.35), one would ideally have to know all the deformation fields $\mathbf{E}(x_k, t, t'_k)$ for all past times $t' \leqslant t$. Then, the most naturally would be to use a uniform time increment $\Delta_k = 1$ in the approximation (4.14) of the integral constitutive equation, which coincide with the time step used in simulations. However, in many cases viscoelastic fluids consist of long polymers, what results in large relaxation times. For such polymers the above mentioned uniform time discretization ($\Delta_k = 1$) is not suitable due to the memory requirements (too many deformation fields would have to be stored) and the computational time needed to calculate those approximations. Therefore, following the ideas introduced by Peters et al. in [37],

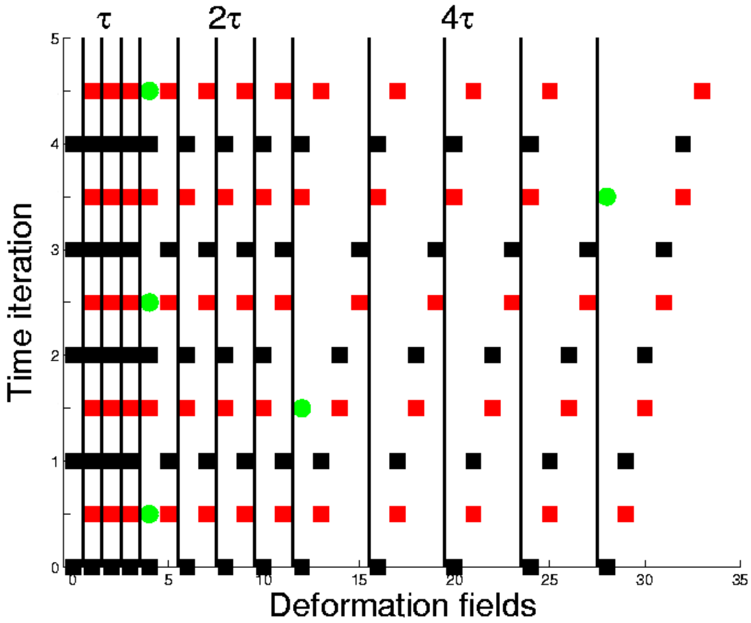

Figure 4.2: *Figure presents which deformation fields are annihilated at appropriate time iteration. Black squares denote deformations used in approximation of integral constitutive equation, red one denote shifted deformation fields after time step and green circles denote annihilated deformations, which later are used to store the youngest one.*

a non–uniform time increment is performed. Now, the past times $t' \leqslant t$ are divided into I intervals with increasing time increment. Each interval $i = 0, .., I-1$ contains a number of N_i fields with a time step $2^i\tau$. Such discretization allows to keep the total number of deformation fields that have to be stored (N_d) in a reasonable orders. It also allows to span the memory of viscoelastic fluids possessing long relaxation times. In Figure 4.2, a schematic picture shows how it is realized. Assume for a moment, that each interval contains of 4 deformations, as indicated in Figure 4.2. In general, one can choose an arbitrary number of deformation tensors in each such interval. For approximation of integral (4.14), it is enough to have only one deformation per interval. This means, that in the situation when two deformations fall into one interval, the older remains and the younger is annihilated. The latter is then used to store the youngest deformation $\mathbf{E}_0$. In our approach the history fluid exhibits is divided in 10 intervals. Each interval contains of 10 deformation fields with equidistant time step. Therefore the number of stored deformation tensors per grid node $N_d = 100$. The time step for the youngest deformations is the same as the time step used in the calculations. Then the time steps for the older fields are obtained by doubling the time steps from the previous intervals. Since the weights decreases for older deformation fields, the error introduced with such treatment of the integral should be relatively small. To say more definite about this error, we will also perform the simulations with one equidistant time step used for all deformations, which is equal to the time used in the simulations. The results obtained by such uniform and non–uniform discretization of the memory integral will be compared. It is clear that both, calculation time and memory requirements are more demanding in the former case. Taking for example $\tau_{relax} = 2$ and time step $\tau = 10^{-3}$, one would have to store $N_d = 10^4$ deformations in the case of uniform time discretization, while in the case of non–uniform time discretization $N_d = 10^2$ spans the same history.

4.6 Calculation of the chain stretch.

The above discussions concern two particular methods (BLPM and DFM) one can use in calculating approximations to the memory integrals of the general form (2.35), that has been presented in Section 2.4.1. Therefore, it can be applied to all models of such general form, also to the ones considered in this thesis, the integral Doi Edwards and the integral Oldroyd B models. Since extension of the original Doi Edwards model, by introducing the stretch of chains, is also considered here, let us finally discuss how to solve the evolution equation for the stretch parameter λ. Its evolution equation (2.40), presented in Section 2.4.2, is solved via the Backward Lagrangian Particle Method, discussed in Section 4.1. At every grid point and every time step of simulations, the trajectory of the particle that arrives at those points is predicted, by integrating equation (4.1) one step τ in time. Then, at the starting points of the trajectory the initialization of $(\nabla \mathbf{v})^T$, $\mathbf{S}$ and λ is performed, by approximation of the appropriate values stored at the nodal grid points and the corresponding time level $t - \tau$ (it means that one step backward is performed). Then, the ordinary differential equation (2.40) is integrated along the particle trajectories to obtain the solution at the nodal points and current time t. At this point, similarly as in calculation of $\mathbf{E}$ deformation tensor, one can solve equation (2.40) in an Eulerian reference frame. Equation (2.40) has to be first transformed by splitting material derivative into partial derivative and convective term, i.e. $\frac{D\lambda}{Dt} = \frac{\partial \lambda}{\partial t} + (\mathbf{v} \cdot \nabla)\lambda$. Finally, the approximation of the extra stress tensor (4.14), according to equation (2.42), is given by

$$\bar{\mathbf{T}}(x,t) = f\lambda^2 \sum_{k=0}^{\infty} W_k f(\mathbf{E}_k(x_k, t, t'_k)). \tag{4.15}$$

4.7 Few words about additional storage and approximation used in BLPM.

Our code is organized in the way, that we store all the values needed to calculate the continuity (3.15) and the momentum (3.16) equations, i.e. velocities (v_1, v_2), pressure (p), extra stress tensor $(\mathbf{T})$ at the center of control volumes. In Section 4.1 and Section 4.2 we have discussed how BLPM and DFM are performed. In both cases N_d number of deformation tensors $(\mathbf{E}_{1,..,N_d})$ have to be stored at each grid node. In the case of BLPM additional storage of the velocities $((v_1, v_2)_{1,..,N_d})$ is needed to be able to perform backward and forward tracking of the particles. At this point one has to approximate velocities at the intermediate points of tracking, what is indicated in Figure 4.1. Also we have to approximate the values of collected along the particle trajectories deformation fields exhibited by the particle. Having all the quantities stored at the centers of control volumes, first we use bilinear interpolation to obtain proper values at the corners of control volumes, and then we apply bilinear approximation, using those values, to get the intermediate velocities and deformations. In 3D case trilinear interpolations and trilinear approximations are used instead.

Chapter 5

Preconditioning techniques for the saddle point problems.

Chapter 5 concerns the solution techniques for the coupled system of partial differential equations, which arise from implicit discretization of the governing equations presented in Chapter 2. In particular, the main focus here is on generalized–Newtonian fluids. After discretization and linearization of the given equations, one gets a two by two block system (3.2), already presented in Section 3.2, called saddle point problem. It is written in the following form

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}.$$

As already indicated in Section 3.1, the block matrix B^T denotes the discrete gradient operator, B the discrete divergence operator. The block matrix A consists of contributions from discretization of the time derivative, as well as from the viscous and the convective terms from the momentum equations. Contrary to Newtonian fluids, where the momentum equations are coupled through the convective term and pressure, in the case of the generalized Newtonian fluids additional strong coupling through the viscous term occur. Since the given system of equations describing such fluids is too complex to perform an analysis of the solution techniques, some simplifications have to be done. First, since most of the viscoelastic materials have large viscosities and the slow flow regimes are considered here, one can drop the convective terms from the momentum equations. It means, that describing viscosity, for example, by the Carreau model (see Section 2.3 and the explanation of this model there), in the limiting case for very slow flows, i.e. $\dot{\gamma} \rightarrow 0$, the given equations become the Stokes equations. Moreover, we assume that A is symmetric and positive definite. Further, let block B has a full rank and is conjugated with B^T. However, the above assumptions are in general not true, in fact A may unsymmetric, and block B may posses rank deficiency. To be effective in solving the above system, good preconditioning techniques are required. First, we will present an analysis of a block diagonal preconditioner to A, in the context of the coupled momentum projection type method presented in Section 3.2.1. In particular, an interesting question for us is how the mixed derivatives appearing in simulations of the generalized-Newtonian fluids can influence the process of iterative solution. Next, different preconditioners for unreduced, transformed and untransformed, system (3.2) are given and discussed. In the considerations below we restrict ourself to 2D case (3D situation is a simple extension).

5.1 Preconditioners for coupled momentum projection method.

The convergence rate of iterative methods depends on spectral properties of the coefficient matrix. The idea of applying a preconditioner is to transform a linear system into one that gives the same solution, but has better spectral properties at the same time. For instance, if D approximates the coefficient matrix A in some way, the transformed system

$$D^{-1}Ax = D^{-1}b \tag{5.1}$$

has the same solution as the original system $Ax = b$, but the spectral properties of its coefficient matrix $D^{-1}A$ may be more favorable. In projection–type methods, presented in Sections 3.2 and 3.2.1, preconditioners are applied at two places of the projection–type algorithm. First preconditioner is applied to the matrix A (from equation (3.2)), when solving intermediate velocity field, and second one for discretized Poisson–like operator in pressure correction equation. Moreover, one can solve the momentum equations iteratively for each velocity component separately by decoupling them, i.e. by discretizing the cross derivatives explicitely. In such case, the preconditioners are applied to each diagonal subblock of matrix A. When the Stokes equations with constant viscosity (i.e. for Newtonian fluids) are considered, the block matrix A from equation (3.2) takes the following form

$$A = \begin{pmatrix} A_{11} & 0 \\ 0 & A_{22} \end{pmatrix}, \tag{5.2}$$

and preconditioner of it can be written as a block diagonal one

$$D = \begin{pmatrix} D_{11} & 0 \\ 0 & D_{22} \end{pmatrix}, \tag{5.3}$$

where D_{11}, D_{22} are preconditioners to blocks A_{11}, A_{22} respectively. Choosing sufficiently close D_{11}, D_{22} to A_{11}, A_{22}, one can cluster eigenvalues of matrix A around the unity number. Since many viscoelastic fluids exhibit variations of viscosity (see Section 2.3), let us first analyze the influence of the terms corresponding to mixed derivatives. As it was already discussed in Chapter 3, in such cases matrix A takes the full block form

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \tag{5.4}$$

where off–diagonal blocks A_{12}, A_{21} correspond to terms $\frac{\partial v_1}{\partial x_2}, \frac{\partial v_2}{\partial x_1}$ from the momentum equations (3.16). To see the influence of those blocks, let us first assume that blocks A_{12}, A_{21} are nonsingular, and again block diagonal preconditioner (5.3) to A is applied. Further, we assume that $[x, y] \neq 0$ is an eigenvector and ξ is the corresponding eigenvalue. Following the approach from [4], for the analysis of the preconditioner (5.3) we must analyze the eigenvalues of

$$D^{-1}A = \begin{pmatrix} D_{11} & 0 \\ 0 & D_{22} \end{pmatrix}^{-1} \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \tag{5.5}$$

what is equivalent to

$$\begin{pmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \xi \begin{pmatrix} x \\ y \end{pmatrix}, \tag{5.6}$$

where $\tilde{A}_{ij} = D_{ii}^{-1} A_{ij}, i, j = 1, 2$. A computation shows, that

$$\begin{cases} \tilde{A}_{11}x + \tilde{A}_{12}y = \xi x \\ \tilde{A}_{21}x + \tilde{A}_{22}y = \xi y \end{cases} \Leftrightarrow$$

$$\begin{cases} \tilde{A}_{11}x + \tilde{A}_{12}y = \xi x \\ x = \xi \tilde{A}_{21}^{-1} y - \tilde{A}_{21}^{-1}\tilde{A}_{22}y. \end{cases}$$

Substituting second equation into the first one, we obtain

$$\xi \tilde{A}_{11}\tilde{A}_{21}^{-1}y - \tilde{A}_{11}\tilde{A}_{21}^{-1}\tilde{A}_{22}y + \tilde{A}_{12}y = \xi^2 \tilde{A}_{21}^{-1}y - \xi \tilde{A}_{21}^{-1}\tilde{A}_{22}y. \tag{5.7}$$

Multiplying above equation form left by $y^T \tilde{A}_{21}$ we obtain

$$\xi^2 \parallel y \parallel^2 - \xi y^T \left(\tilde{A}_{22} + \tilde{A}_{21}\tilde{A}_{11}\tilde{A}_{21}^{-1}\right) y + y^T \left(\tilde{A}_{21}\tilde{A}_{11}\tilde{A}_{21}^{-1}\tilde{A}_{22}\right) y - y^T \left(\tilde{A}_{21}\tilde{A}_{12}\right) y = 0. \tag{5.8}$$

Above results are put together in a theorem.

Theorem 3 *Let* $D = \begin{pmatrix} D_{11} & 0 \\ 0 & D_{22} \end{pmatrix}$ *be a preconditioner to* $A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$ *that comes form discretization of the momentum equations (3.16). Then the eigenvalues of* $D^{-1}A$ *satisfy* $\xi = \frac{a \pm \sqrt{a^2 - 4(b-c)}}{2}$ *where* $a = \frac{y^T\left(\tilde{A}_{22} + \tilde{A}_{21}\tilde{A}_{11}\tilde{A}_{21}^{-1}\right)y}{\|y\|^2}$, $b = \frac{y^T\left(\tilde{A}_{21}\tilde{A}_{11}\tilde{A}_{21}^{-1}\tilde{A}_{22}\right)y}{\|y\|^2}$, $c = \frac{y^T\left(\tilde{A}_{21}\tilde{A}_{12}\right)y}{\|y\|^2}$ *and* $[x, y]$ *is an eigenvector of* $D^{-1}A$.

■

The proof is following from the above derivations. Above theorem shows, that even the choice $D_{11} = A_{11}$ and $D_{22} = A_{22}$ will not cluster eigenvalues around unity. In such case $a = 2, b = 1$ and $\xi = 1 \pm \sqrt{c}$, and it shows influence of the off–diagonal blocks on eigenvalue distribution.

However, at this point appears the question if such block diagonal preconditioners are good enough for the generalized Newtonian problems. To answer this question let us first define, following [41], the standard Hilbert space $\mathcal{H}(\Omega)$ with the dot product and norm

$$(u, v) = \int_\Omega u(s)v(s)ds, \qquad \|u\| = (u, u)^{\frac{1}{2}}.$$

For the 2D vectors $\mathbf{u}$,$\mathbf{v}$ the Hilbert space $\mathcal{H}_2 = \mathcal{H} \otimes \mathcal{H}$ with the dot product

$$(\mathbf{u}, \mathbf{v}) = (u_1, v_1) + (u_2, v_2)$$

is introduced. Since we are interested here in understanding the influence of cross derivatives on iterative process, let us now analyze the divergence operator $D_v\mathbf{v} = -\nabla \cdot \boldsymbol{\sigma}$ (we use similar notation as in Sections 3.1 and 3.2) on the set of vector functions $\mathbf{v}$ equal zero on $\partial\Omega$. For simplicity, we assume that $\boldsymbol{\sigma}$ represents the Carreau constitutive equation (2.22). Note, that operator D_v can be written in the matrix form

$$D_v = \begin{pmatrix} \mathcal{N}_{11} & \mathcal{N}_{12} \\ \mathcal{N}_{21} & \mathcal{N}_{22} \end{pmatrix},$$

where

$$\begin{aligned} \mathcal{N}_{11}v_1 &= -\tfrac{\partial}{\partial x_1}(2\nu\tfrac{\partial v_1}{\partial x_1}) - \tfrac{\partial}{\partial x_2}(\nu\tfrac{\partial v_1}{\partial x_2}) \\ \mathcal{N}_{12}v_2 &= -\tfrac{\partial}{\partial x_2}(\nu\tfrac{\partial v_2}{\partial x_1}) \\ \mathcal{N}_{21}v_1 &= -\tfrac{\partial}{\partial x_1}(\nu\tfrac{\partial v_1}{\partial x_2}) \\ \mathcal{N}_{22}v_2 &= -\tfrac{\partial}{\partial x_1}(\nu\tfrac{\partial v_2}{\partial x_1}) - \tfrac{\partial}{\partial x_2}(2\nu\tfrac{\partial v_2}{\partial x_2}) \end{aligned}$$

and $\nu = \frac{\eta(\dot\gamma)}{\rho}$ denotes the kinematic viscosity. In [41] authors have showed, that $(D_v\mathbf{v}, \mathbf{u}) = (\mathbf{v}, D_v\mathbf{u})$, i.e. operator $D_v = D_v^* > 0$ is positive and self–adjoint. Moreover, they have showed that operator D_v and Laplace–like operator $L\mathbf{v} = -\nabla\cdot(\nu\nabla\mathbf{v})$, which can be written as

$$L = \begin{pmatrix} \mathcal{L}_{11} & 0 \\ 0 & \mathcal{L}_{22} \end{pmatrix},$$

where

$$\begin{aligned} \mathcal{L}_{11}v_1 &= -\tfrac{\partial}{\partial x_1}(\nu\tfrac{\partial v_1}{\partial x_1}) - \tfrac{\partial}{\partial x_2}(\nu\tfrac{\partial v_1}{\partial x_2}) \\ \mathcal{L}_{22}v_2 &= -\tfrac{\partial}{\partial x_1}(\nu\tfrac{\partial v_2}{\partial x_1}) - \tfrac{\partial}{\partial x_2}(\nu\tfrac{\partial v_2}{\partial x_2}), \end{aligned}$$

are energy equivalent, i.e. $\nu(L\mathbf{v}, \mathbf{v}) \leq (D_v\mathbf{v}, \mathbf{v}) \leq 2\nu(L\mathbf{v}, \mathbf{v})$. This result shows, that the block diagonal preconditioner (5.3) applied to the system (5.4), arising after linearization and discretization of the momentum equations (3.16), is a good choice.

Let us illustrate the above results in a following numerical experiment. The perfect case to see the influence of the off–diagonal blocks of matrix (5.4) on the iterative procedure would be to choose such a test fluid, for which the coefficients coming from the discretization of the viscous operator and entering the diagonal blocks A_{11}, A_{22}

are multiplied by a constant value (η_0), and the coefficients of the off-diagonal blocks A_{12}, A_{21} changes ($\eta(\dot{\gamma})$) for different fluid parameters. To be close to such perfect situation, we choose an artificial fluid, described by a stress–strain relation (2.26), where the elongational viscosity η_e remains constant and the shear viscosity η changes. The values of the first enter the diagonal blocks of matrix A and the values of the latter the off–diagonal ones. However, one has to point out that some of the values coming from the cross derivatives enter also the diagonal blocks. In this experiment, we choose the zero shear–rate viscosity $\eta_0 = 100$ and the fluid parameters appearing in (2.26) as $Cu_e = 1, n_e = 1, Cu = 1$ and the only parameter that changes is n. As already discussed in Section 2.3, the parameters Cu, n and Cu_e, n_e are used to fit experimental data of shear and extensional properties of the fluid respectively. Figure 5.1 shows, that increase of n form $n = 1.01$ (low shear–thickening) to $n = 1.6$ (high shear–thickening) results in larger spread of the eigenvalues distribution. It is what one should expect, since choosing $n = 1.6$ results in much higher values of $\eta_s(\dot{\gamma})$ in the regions where $\dot{\gamma} > 0$ comparing with $n = 1.01$, thus increasing the strength of the A_{12}, A_{21} blocks. The maximum value of η in the latter case ($n = 1.6$) is 5 times larger than in the former one ($n = 1.01$). On the other hand, one can see, that such big increase of the shear viscosity η does not lead to significant dispersion of the eigenvalue distribution, what should be expected after estimations given in [41].

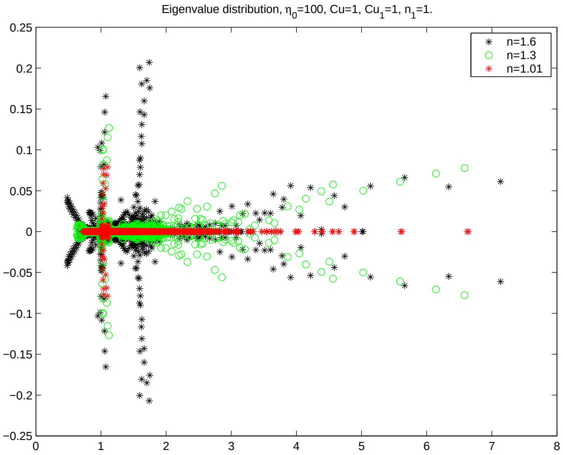

Figure 5.1: *Eigenvalue distribution for block matrix A preconditioned with a block diagonal preconditioner.*

As a preconditioner for a coefficient matrix $B\{diag(A)\}^{-1}B^T$, resulting form the discretized pressure correction equation (3.14), one can for example choose an incomplete LU factorization (ILU) or its relaxed form ($RILU(\alpha)$ with relaxation parameter $0 < \alpha < 1$). For an overview of different preconditioners like ILU, $RILU(\alpha)$, $MILU$, $Jacobi$, SOR, etc. we refer interested reader to [39]. Keeping above results in mind, let us now discuss some preconditioning techniques for unreduced saddle point problem (3.2).

5.2 Preconditioners for untransformed fully coupled system.

During years, many different preconditioning techniques for saddle point problems have been developed. These are, for example, block *LU* preconditioners (see.[26]), right oriented triangular preconditioners ([10, 42]), block Gauss-Seidel preconditioner (as well as its symmetrized form), congruence transformation, block indefinite triangular preconditioner ([4]), and others. In general (see [5]) applying a preconditioned iterative method for solving a system of equations is equivalent to applying a non–preconditioned method for solving the transformed system. The examples of such approach can be found in [4, 10, 42]. Moreover in [42] it is stated, that left, right, or double sided preconditioners give almost identical results. Several preconditioners for (3.2) are carefully analyzed in [4]. However, as already mentioned all those preconditioners were analyzed for Newtonian flow problems (i.e. possessing constant viscosity). Since we would like to compare different preconditioner techniques (for untransformed and transformed system (3.2)) for variable viscosity fluids later on, we have chosen two preconditioners applied to the untransformed system (3.2), and presented in [4], the block Gauss-Seidel and the indefinite block triangular ones.

5.2.1 Block Gauss–Seidel preconditioner to untransformed saddle-point problem.

The block Gauss-Seidel preconditioner to untransformed system (3.2) takes the form

$$\mathcal{D}_{GS} = \begin{pmatrix} D_1 & 0 \\ B & D_2 \end{pmatrix}, \tag{5.9}$$

where D_1 and D_2 are symmetric, positive definite preconditioners to A and $BD_1^{-1}B^T$ respectively. The analysis of the eigenvalues of

$$\begin{pmatrix} D_1 & 0 \\ B & D_2 \end{pmatrix}^{-1} \begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} = \begin{pmatrix} D_1^{-1}A & D_1^{-1}B^T \\ D_2^{-1}B(I - D_1^{-1}A) & -D_2^{-1}BD_1^{-1}B^T \end{pmatrix} \tag{5.10}$$

is collected in following theorem.

Theorem 4 *Let* $\mathcal{D}_{GS} = \begin{pmatrix} D_1 & 0 \\ B & D_2 \end{pmatrix}$ *be a preconditioner to* $\mathcal{L} = \begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix}$. *Then the eigenvalues of* $\mathcal{D}_{GS}^{-1}\mathcal{L}$ *satisfy* $\xi = 1 + \frac{1}{2}(a + b^2 - 1) \pm \sqrt{(a + b^2 - 1)^2 + 4a}$, *where* $a = \frac{\tilde{x}^T(\tilde{A}-I)\tilde{x}}{\|\tilde{x}\|^2}$, $b = \frac{\|\tilde{B}\tilde{x}\|}{\|\tilde{x}\|}$, $\tilde{x} = D_1^{-\frac{1}{2}}x$, $\tilde{A} = D_1^{-\frac{1}{2}}AD_1^{-\frac{1}{2}}$, $\tilde{B} = D_2^{-\frac{1}{2}}BD_1^{-\frac{1}{2}}$ *and [x,y] is an eigenvector of* $\mathcal{D}_{GS}^{-1}\mathcal{L}$.

■

Proof can be found in [4]. Let us look more close at equation (5.10). If one chooses D_1 close to A, the left lower block tends to a zero block, as well as upper left block to the identity matrix. Now taking D_2 close to $BD_1^{-1}B^T$ drives lower right block to identity one. All this means that eigenvalues are clustered around unity number. In fact, if $D_1 = A$ then $a = 0$ and eigenvalues $\xi = 1 + \left\{ \begin{array}{l} b^2 - 1 \\ 0 \end{array} \right.$ are real and equal the unit number and the eigenvalues of $D_2^{-1}BD_1^{-1}B^T$, which are positive. Further if $D_2 = BD_1^{-1}B^T$ then $b = 1$. Clustering occurs also in the case when $|a|$ is small. However, when b is small, which happens for a nearly rank deficient matrix B, eigenvalues ξ take values close to $1 + a$ and 0. This means that preconditioned matrix is nearly singular.

5.2.2 Indefinite block triangular preconditioner to untransformed saddle-point problem.

Second preconditioner, the indefinite block triangular, take the same saddle point form as the given matrix (3.2)

$$\mathcal{D}_{IT} = \begin{pmatrix} D_1 & B^T \\ B & 0 \end{pmatrix}, \tag{5.11}$$

where D_1 approximates block A in some sense. For the analysis of above preconditioner we will use following theorem.

Theorem 5 *Let B, C, E be real matrices of order $n \times m, m \times m$ and $n \times n$ respectively, where B has a full rank ($= m$), C is positive semi-definite and E is symmetric. Then the eigenvalues of the generalized eigenvalue problem*

$$\gamma \begin{pmatrix} I & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} E & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}, \qquad |x| + |y| \neq 0 \tag{5.12}$$

where $x \in \mathbb{C}^n$ and $y \in \mathbb{C}^m$, satisfy:
(a) $\gamma = \frac{x^T E x}{x^T (I + B^T C^{-1} B) x}$, if $Ex \neq 0$ and C is positive definite.
(b) $\gamma = 0$ if and only if $Ex = 0, x \neq 0$ and the dimension of the eigenvector space corresponding to the zero eigenvalue is $m + q$, where $q = dim\{ker(E)\}$.
(c) the nonzero eigenvalues are contained in the interval
$min\{0, \xi_{min}(E)\} \leq \gamma \leq \xi_{max}(E)$.

■

Proof of above theorem is given in [4]. In the case when $C = 0$, there holds that for $\gamma \neq 0$, $\gamma = \frac{x^T E x}{x^T x}$ and corresponding eigenvector is $\begin{bmatrix} x \\ \frac{1}{\gamma}(BB^T)^{-1}BEx \end{bmatrix}$. Now, if D_1 is a symmetric and positive definite preconditioner to A, we can write the system (3.2) preconditioned by (5.11) in the following way

$$\xi \begin{pmatrix} D_1 & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \Leftrightarrow$$

$$\gamma \begin{pmatrix} D_1 & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} A - D_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix},$$

where $\gamma = \xi - 1$. Further, the above can be transformed to

$$\gamma \begin{pmatrix} I & \tilde{B}^T \\ \tilde{B} & 0 \end{pmatrix} \begin{pmatrix} \tilde{x} \\ y \end{pmatrix} = \begin{pmatrix} \tilde{E} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \tilde{x} \\ y \end{pmatrix},$$

where $\tilde{E} = D_1^{-\frac{1}{2}} A D_1^{-\frac{1}{2}} - I$, $\tilde{B} = B D_1^{-\frac{1}{2}}$ and $\tilde{x} = D_1^{-\frac{1}{2}} x$. Since above problem has the same form as (5.12) the analysis from *Theorem 5* is applicable.

5.3 Preconditioners for transformed fully coupled system.

Let us now discuss a two–stage approach. At the first stage system (3.2) is transformed using a matrix like above mentioned preconditioners. In the second stage, instead of using unpreconditioned iterative method, we use preconditioned one. For the transformation we use matrix of the following form

$$\begin{pmatrix} H^{-1} & 0 \\ -BH^{-1} & I \end{pmatrix}, \tag{5.13}$$

where H is some preconditioner to A and

$$\begin{pmatrix} H^{-1} & 0 \\ -BH^{-1} & I \end{pmatrix} \begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} = \begin{pmatrix} H^{-1}A & H^{-1}B^T \\ B(I - H^{-1}A) & -BH^{-1}B^T \end{pmatrix}.$$

The aim of such transformation is to obtain "good" blocks at the main diagonal of the transformed system. It is clear, that choice $H = A$ leads to upper block triangular system with identity operator I on left upper position. However, the operator $BA^{-1}B^T$ will have a full form in this case. Moreover, since we want to apply preconditioner to transformed system later on, the computational effort at this stage should not be big. Therefore similar to the projection-like algorithms, we select

$$H = diag\{A\}, \tag{5.14}$$

or alternatively $H = diag\{part(A)\}$. Thus, we obtain transformed system

$$\bar{\mathcal{L}} = \begin{pmatrix} \bar{A} & \bar{B}^T \\ B(I - \bar{A}) & -B\bar{B}^T \end{pmatrix}, \tag{5.15}$$

where $\bar{A} = H^{-1}A$ and $\bar{B}^T = H^{-1}B^T$. Let us now present and analyze three different preconditioners applied to (5.15). Let D_1 and D_2 denote the nonsingular preconditioners to $\bar{A}$ and $-BD_1^{-1}\bar{B}^T$ respectively.

5.3.1 Block diagonal preconditioner to transformed saddle-point problem.

Let us first consider a block diagonal preconditioner to $\bar{\mathcal{L}}$

$$\bar{\mathcal{D}}_D^{-1} = \begin{pmatrix} D_1^{-1} & 0 \\ 0 & D_2^{-1} \end{pmatrix} \tag{5.16}$$

and assume that ξ is an eigenvalue and $[x, y] \neq 0$ corresponding eigenvector. Let us analyze the eigenvalues of a matrix

$$\begin{pmatrix} D_1^{-1} & 0 \\ 0 & D_2^{-1} \end{pmatrix} \begin{pmatrix} \bar{A} & \bar{B}^T \\ B(I-\bar{A}) & -B\bar{B}^T \end{pmatrix} = \begin{pmatrix} D_1^{-1}\bar{A} & D_1^{-1}\bar{B}^T \\ D_2^{-1}B(I-\bar{A}) & -D_2^{-1}B\bar{B}^T \end{pmatrix},$$

what leads to the equation

$$\begin{pmatrix} D_1^{-1}\bar{A} & D_1^{-1}\bar{B}^T \\ D_2^{-1}B(I-\bar{A}) & -D_2^{-1}B\bar{B}^T \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \xi \begin{pmatrix} x \\ y \end{pmatrix}.$$

This is equivalent to

$$\begin{cases} D_1^{-1}\bar{A}x + D_1^{-1}\bar{B}^T y = \xi x \\ D_2^{-1}B(I-\bar{A})x - D_2^{-1}B\bar{B}^T y = \xi y \end{cases}.$$

Multiplying the first equation from left by $D_2^{-1}BD_1$ and summing up with the second one gives

$$\xi y = D_2^{-1}Bx - \xi D_2^{-1}BD_1 x,$$

what further leads to

$$\xi^2 x - \xi(D_1^{-1}\bar{A}x - D_1^{-1}\bar{B}^T D_2^{-1}BD_1 x) - D_1^{-1}\bar{B}^T D_2^{-1}Bx = 0.$$

Let us now multiply above equation form left by $x^T B^T B$, what gives

$$\xi^2\|Bx\|^2 - \xi x^T B^T(BD_1^{-1}\bar{A}x - BD_1^{-1}\bar{B}^T D_2^{-1}BD_1 x) - x^T B^T(BD_1^{-1}\bar{B}^T D_2^{-1})Bx = 0.$$

We collect above results in theorem.

Theorem 6 *Let* $\bar{\mathcal{D}}_D^{-1} = \begin{pmatrix} D_1^{-1} & 0 \\ 0 & D_2^{-1} \end{pmatrix}$ *be a preconditioner to the transformed system* $\bar{\mathcal{L}}$ *(from equation 5.15). Then the eigenvalue of* $\bar{\mathcal{D}}_D^{-1}\bar{\mathcal{L}}$ *satisfy* $\xi = \frac{(a-b)\pm\sqrt{(a-b)^2+4c}}{2}$, *where* $a = \frac{x^T B^T BD_1^{-1}\bar{A}x}{\|Bx\|^2}$, $b = \frac{x^T B^T(BD_1^{-1}\bar{B}^T D_2^{-1}BD_1)x}{\|Bx\|^2}$, $c = \frac{x^T B^T(BD_1^{-1}\bar{B}^T D_2^{-1})Bx}{\|Bx\|^2}$ *and [x,y] is an eigenvector of* $\bar{\mathcal{D}}_D^{-1}\mathcal{L}$.

■

The proof is following from the above derivations. Suppose now, that we have taken $H = A$ at the first step during transformation. Then $D_1 = I$, $a = 1$, $b = c$ and eigenvalues can be expressed as $\xi = \frac{(1-c)\pm|(1+c)|}{2}$. Moreover, if one chooses D_2 to be symmetric and positive definite, the eigenvalues of $-BD_1^{-1}\bar{B}^TD_2^{-1}$ are positive and $\xi = \begin{cases} 1 \\ -c \end{cases}$. Therefore, choosing D_2 sufficiently close to the Schur complement clusters eigenvalues around unity number. In practice, however, this is not possible since A^{-1} is a full matrix, thus finding good preconditioner to the Schur complement is too costly. Above result shows, that for the choice (5.14) of H during the transformation, the block diagonal preconditioner (5.16) has a certain limit, i.e. it is not able to cluster eigenvalues around unity number even for a perfect preconditioners $D_1 = \bar{A}$ and $D_2 = -B\bar{A}^{-1}\bar{B}^T$ (since $a = 1$, $c = -1$, $b = \frac{x^TB^TBD_1x}{\|Bx\|^2}$). To illustrate that, let us make a numerical experiment for artificial shear–thinning fluid with a parameters $\eta_0 = 100$, $Cu = 1$ and $n = 0.2$ (for details concerning model parameters see Section 2.3). During transformation $H = diag(A)$ and later we have chosen $D_1 = \bar{A}$ and $D_2 = -B\bar{A}^{-1}\bar{B}^T$. Eigenvalue distribution for such a choice is shown in Figure 5.2,

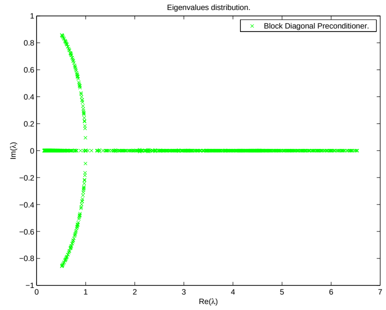

Figure 5.2: *Eigenvalue distribution for optimal choice of block diagonal preconditioner* $\bar{\mathcal{D}}_D$ *to transformed system* $\bar{\mathcal{L}}$.

where $max\{\mathcal{R}e\} \approx 6.5, min\{\mathcal{R}e\} \approx 0.15, max\{\mathcal{I}m\} \approx 0.86, min\{\mathcal{I}m\} \approx -0.86,$.

5.3.2 Block lower triangular preconditioner to transformed saddle-point problem.

Next preconditioner to be considered is a block lower triangular one to $\bar{\mathcal{L}}$ of the form

$$\bar{\mathcal{D}}_{LT}^{-1} = \begin{pmatrix} D_1^{-1} & 0 \\ B(I-\bar{A}) & D_2^{-1} \end{pmatrix}. \tag{5.17}$$

Similarly as in previous case we will analyze eigenvalues of the matrix

$$\bar{\mathcal{D}}_{LT}^{-1}\bar{\mathcal{L}} = \begin{pmatrix} D_1^{-1}\bar{A} & D_1^{-1}\bar{B}^T \\ D_2^{-1}B(I-\bar{A}) + B(I-\bar{A})\bar{A} & -D_2^{-1}B\bar{B}^T + B(I-\bar{A})\bar{B}^T \end{pmatrix}.$$

Now, we write the eigenvalue equation in a bit different way as

$$\begin{pmatrix} \bar{A} & \bar{B}^T \\ B(I-\bar{A}) & -B\bar{B}^T \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \xi \begin{pmatrix} D_1 & 0 \\ -D_2 B(I-\bar{A})D_1 & D_2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

Further simple algebraic transformations result in a quadratic equation for eigenvalue ξ as

$$\xi^2 x - \xi(D_1^{-1}\bar{A}x + D_1^{-1}\bar{B}^T B(I-\bar{A})D_1 x - D_1^{-1}\bar{B}^T D_2^{-1} B D_1 x) - D_1^{-1}\bar{B}^T D_2^{-1} B x = 0,$$

which after multiplication form left by $x^T B^T B$ transforms to

$$\xi^2 - \xi\left(\frac{x^T B^T B D_1^{-1}\bar{A}x}{\|Bx\|^2} + \frac{x^T B^T B D_1^{-1}\bar{B}^T B(I-\bar{A})D_1 x}{\|Bx\|^2} - \frac{x^T B^T (B D_1^{-1}\bar{B}^T D_2^{-1}) B D_1 x}{\|Bx\|^2}\right)$$

$$-\frac{x^T B^T (B D_1^{-1}\bar{B}^T D_2^{-1}) B x}{\|Bx\|^2} = 0.$$

We put together above results in another theorem.

Theorem 7 *Let* $\bar{\mathcal{D}}_{LT}^{-1} = \begin{pmatrix} D_1^{-1} & 0 \\ B(I-\bar{A}) & D_2^{-1} \end{pmatrix}$ *be a preconditioner to the transformed system* $\bar{\mathcal{L}}$ *(from equation 5.15). Then the eigenvalue of* $\bar{\mathcal{D}}_{LT}^{-1}\bar{\mathcal{L}}$ *satisfy* $\xi = \frac{(a+b-c)\pm\sqrt{(a+b-c)^2+4d}}{2}$, *where* $a = \frac{x^T B^T B D_1^{-1}\bar{A}x}{\|Bx\|^2}$, $b = \frac{x^T B^T (B D_1^{-1}\bar{B}^T B(I-\bar{A})D_1)x}{\|Bx\|^2}$, $c = \frac{x^T B^T (B D_1^{-1}\bar{B}^T D_2^{-1}) D_1 B x}{\|Bx\|^2}$, $d = \frac{x^T B^T (B D_1^{-1}\bar{B}^T D_2^{-1}) B x}{\|Bx\|^2}$ *and [x,y] is an eigenvector of* $\bar{\mathcal{D}}_{LT}^{-1}\mathcal{L}$.

■

For a proof see the derivations above. If again one has taken $H = A$ during transformation, then $D_1 = I$, $a = 1$, $b = 0$, $c = d$ and eigenvalues are the same as in block diagonal preconditioner $\xi = \frac{(1-c)\pm|(1+c)|}{2}$ and the same discussion as before is applicable. Moreover, in second extreme case, when H follows equation (5.14) and both D_1 and D_2 are chosen in optimal way (i.e. $D_1 = \bar{A}$, $D_2 = -B\bar{A}^{-1}\bar{B}^T$), the block lower triangular preconditioner $\bar{\mathcal{D}}_{LT}$ returns the same results as block diagonal one. This is confirmed in the same numerical experiment as performed for diagonal preconditioner $\bar{\mathcal{D}}_D$ (see Figure 5.3), but applying $\bar{\mathcal{D}}_{LT}$ as a preconditioner to $\bar{\mathcal{L}}$.

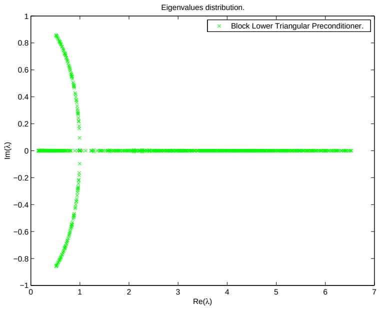

Figure 5.3: *Eigenvalue distribution for optimal choice of block lower triangular preconditioner* $\bar{\mathcal{D}}_{LT}$ *to transformed system* $\bar{\mathcal{L}}$.

5.3.3 Block Gauss–Seidel preconditioner to transformed saddle-point problem.

The last preconditioner discussed in this thesis, the block Gauss-Seidel applied to the transformed system (5.15), removes above presented limitation. To motivate such preconditioning strategy, let us write the system (5.15) in the block LU factorization as

$$\begin{pmatrix} \bar{A} & \bar{B}^T \\ B(I-\bar{A}) & -B\bar{B}^T \end{pmatrix} = \begin{pmatrix} \bar{A} & 0 \\ B(I-\bar{A}) & -B\bar{A}^{-1}\bar{B}^T \end{pmatrix} \begin{pmatrix} I & \bar{A}^{-1}\bar{B}^T \\ 0 & I \end{pmatrix}. \tag{5.18}$$

It shows, that applying from left the block Gauss-Seidel preconditioner to (5.15), which is written in following form

$$\bar{\mathcal{D}}_{GS}^{-1} = \begin{pmatrix} D_1 & 0 \\ B(I-\bar{A}) & D_2 \end{pmatrix}^{-1} = \begin{pmatrix} D_1^{-1} & 0 \\ -D_2^{-1}B(I-\bar{A})D_1^{-1} & D_2^{-1} \end{pmatrix}, \tag{5.19}$$

where D_1 and D_2 are sufficiently close to $\bar{A}$ and $-BD_1^{-1}\bar{B}^T$ respectively, will lead to eigenvalue clustering. To confirm that, let us again analyze the eigenvalue problem of the form

$$\begin{pmatrix} \bar{A} & \bar{B}^T \\ B(I-\bar{A}) & -B\bar{B}^T \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \xi \begin{pmatrix} D_1 & 0 \\ B(I-\bar{A}) & D_2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix},$$

what is equivalent to

$$\left\{ \begin{array}{r} \bar{A}x + \bar{B}^T y = \xi D_1 x \\ B(I-\bar{A})x - B\bar{B}^T y = \xi B(I-\bar{A})x + \xi D_2 y \end{array} \right. . \tag{5.20}$$

Multiplying first equation from left by B and summing up with second one gives

$$\xi y = D_2^{-1}Bx - \xi(D_2^{-1}Bx + D_2^{-1}B(D_1 - \bar{A})x).$$

Now, multiplying again first equation of (5.20) form left by ξD_1^{-1} and further using the above result gives

$$\xi^2 x = \xi D_1^{-1}\bar{A}x + D_1^{-1}\bar{B}^T D_2^{-1}Bx - \xi(D_1^{-1}\bar{B}^T D_2^{-1}Bx + D_1^{-1}\bar{B}^T D_2^{-1}B(D_1 - \bar{A})x).$$

Finally, we transform above equation by multiplying from left by $x^T B^T B$ to

$$\xi^2 - \xi\frac{x^T B^T B D_1^{-1}\bar{A}x}{\|Bx\|^2} - \frac{x^T B^T B D_1^{-1}\bar{B}^T D_2^{-1}Bx}{\|Bx\|^2} +$$

$$\xi\left(\frac{x^T B^T B D_1^{-1}\bar{B}^T D_2^{-1}Bx}{\|Bx\|^2} + \frac{x^T B^T B D_1^{-1}\bar{B}^T D_2^{-1}B(D_1 - \bar{A})x}{\|Bx\|^2}\right) = 0.$$

Following theorem gather above results.

Theorem 8 *Let* $\bar{\mathcal{D}}_{GS}^{-1} = \begin{pmatrix} D_1 & 0 \\ B(I-\bar{A}) & D_2 \end{pmatrix}^{-1}$ *be preconditioner to transformed system* $\bar{\mathcal{L}}$ *(equation 5.15). Then the eigenvalue of* $\bar{\mathcal{D}}_{GS}^{-1}\bar{\mathcal{L}}$ *satisfy*
$\xi = \frac{(a-b-c)\pm\sqrt{(a-b-c)^2+4b}}{2}$, *where* $a = \frac{x^T B^T B D_1^{-1}\bar{A}x}{\|Bx\|^2}$, $b = \frac{x^T B^T (B D_1^{-1}\bar{B}^T D_2^{-1})Bx}{\|Bx\|^2}$,
$c = \frac{x^T B^T (B D_1^{-1}\bar{B}^T D_2^{-1})B(D_1-\bar{A})x}{\|Bx\|^2}$ *and [x,y] is an eigenvector of* $\bar{\mathcal{D}}_{GS}^{-1}\mathcal{L}$.

■

Again the proof is contained in the derivations above. The advantage of such preconditioner is that, independent from the choice of H during the transformation step, the clustering of eigenvalues around unity number is possible. It comes for the fact, that if one chooses $D_1 \approx \bar{A}$, $D_2 \approx -BD_1^{-1}\bar{B}^T$ then $a \approx 1$, $b \approx -1$, $c \approx 0$ and $\xi \approx 1$. Moreover if $D_1 = \bar{A}$ the eigenvalues satisfy $\xi = \begin{cases} 1 \\ -b \end{cases}$ and, similarly to $\mathcal{D}_{GS}$ preconditioner presented in Section 5.2.1, are real and equal one and $-b$. The latter is positive and taking sufficient close approximation D_2 to the Schur complement $(-BA^{-1}B^T)$ result in $b \approx -1$. Again, we repeat numerical experiment for shear–thinning fluid with the same fluid parameters as previously, i.e. $\eta_0 = 100$, $Cu = 1$ and $n = 0.2$. Result of applying block Gauss–Seidel preconditioner $\bar{\mathcal{D}}_{GS}$ to $\bar{\mathcal{L}}$ with an optimal choice for $D_1 = \bar{A}$ and $D_2 = -BD_1^{-1}\bar{B}^T$ is shown in Figure 5.4. One can observe that eigenvalues are tightly clustered around unity number, where $\mathcal{R}e = 1, max\{\mathcal{I}m\} \approx 4.6 \cdot 10^{-6}, min\{\mathcal{I}m\} \approx -4.6 \cdot 10^{-6}$, what confirms previous discussion. All eigenvalue distribution plots, i.e. Figures 5.1, 5.2 , 5.3 and 5.4, were performed using *MATLAB*.

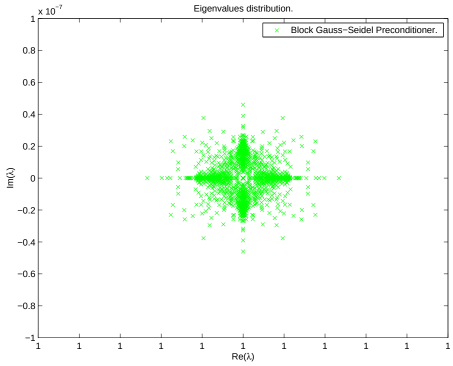

Figure 5.4: *Eigenvalue distribution for optimal choice of block Gauss–Seidel preconditioner* $\bar{\mathcal{D}}_{GS}$ *to transformed system* $\bar{\mathcal{L}}$.

Chapter 6

Numerical results.

In this Chapter, we present results obtained from numerical simulations of viscoelastic fluids. As already discussed in previous Chapters, viscoelastic fluids consist of both viscous and elastic properties. Therefore, first we present results of generalized Newtonian fluids, where we study dependence of viscosity variations on the flow. In particular, we focus on both shear and elongational viscosity properties of the fluid. First, we present the influence of shear–thinning viscosity properties on the flow in a planar contraction domain. Since such fluids were already extensively studied, it is a good opportunity to validate our work. Then, we show the influence of extensional–thickening on the flow. In this case, we observe very good agreement with experiments. Another interesting case study are the so called Boger fluids, alternatively named constant–viscosity elastic fluids. These are examples of dilute polymer solutions, and are modeled by the integral Oldroyd B constitutive equation These liquids helps in understanding the elasticity effects, artificially setting viscosity to a constant value. Simulations show very good qualitative and quantitative agreement with experiments, and results obtained by differential Oldroyd B counterpart, respectively. Next, we present results from simulations of concentrated polymer solutions, modeled by the integral Doi Edwards constitutive equation. We achieve very good agreement with experimental data in prediction of first normal stress difference and shear viscosity, defined as the ratio of a shear stress to a shear–rate. Finally, we present performance of different iterative solvers, presented and discussed in the previous Chapter 5, with respect to time steps, geometries and deviation of the fluid from Newtonian flow behavior.

6.1 Simulations of shear–thinning fluids.

There is much interest in understanding the flow of non-Newtonian fluids. However, since many different effects, like shear and elongational rate dependent viscosities, as well as elastic contribution influence the flow, it is important to look at the varying viscosity problems separately. Moreover, most of the viscoelastic fluids exhibit shear–thinning behavior. Therefore, in current Section the simulation results of the generalized Newtonian, shear–thinning, fluids are presented. To investigate the in-

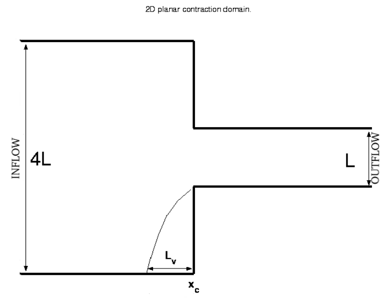

Figure 6.1: *2D planar contraction domain.*

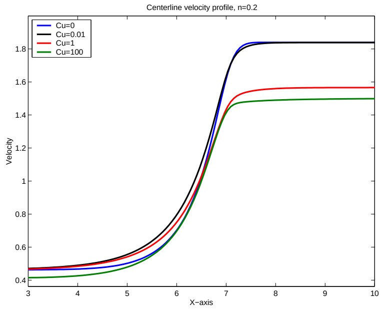

Figure 6.2: *The effect of varying Cu number on the calculated centerline velocity for n=0.2, Carreau model.*

teraction of shear-thinning, similarly as in [25], the Carreau model parameters Cu and n will be varied (detailed description and explanation of the above model parameters is performed in Section 2.3). The result will be discussed in terms of the centerline velocity profile, axial velocity profile in the plane of contraction and the upstream vortex size. As a test geometry we use 4:1 planar contraction domain shown in Figure 6.1. To exclude the inertial effects we choose the flow with Reynolds number $Re = 10^{-2}$ defined as $Re = \frac{\rho \bar{U} L}{\eta_0}$, where $\rho = 1[\frac{g}{cm^3}]$ stands for fluid density, L is the width of the upstream channel, $\eta_0 = 100[Pas]$ is a zero–shear viscosity and $\bar{U}$ is av-

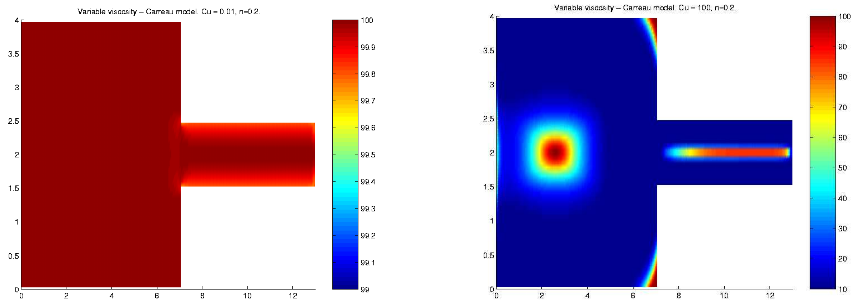

Figure 6.3: *Carreau viscosity plots for* $n = 0.2$, $Cu = 0.01$ *(left figure),* $Cu = 100$ *(right figure).*

erage upstream channel velocity. Note, that for shear–thinning fluids local Reynolds numbers can be higher, especially near to the walls where higher shear–rates occur and decrease local viscosities even couple of order of magnitude. First, we examine the changes of the centerline velocity profiles for high shear–thinning fluid ($n = 0.2$) when changing Cu number. Results are presented in Figure 6.2. For $Cu = 0.01$ the axial velocity attain almost the same maximum as in Newtonian case ($Cu = 0$). This should be expected, since for such low Carreau number the shear–rate required for an onset of shear–thinning is almost not present (see left plot from Figure 6.3). Carreau fluid, however, exhibits less steep axial velocity profile compared with the Newtonian one. Further increase of Carreau Cu number results in decrease of centerline velocity profiles. At $Cu = 100$, the flow is dominated by the power-law region of the viscosity function (see right plot from Figure 6.3), and further increase of Cu number does not change the velocity profile. Next, decrease of power-law index n at constant Cu has quantitatively the same effect as increasing Cu at constant n. It is shown in Figure 6.4. The reason for such behavior is that both combinations lead to more shear–thinning. The effect of increasing Cu number, or decreasing power–law index n on axial velocity profile at the contraction $x = x_c$ is shown in Figure 6.5. In both cases more shear–thinning fluids flatter fully-developed velocity profile in downstream channel. Again, neither larger Cu nor smaller n do further change

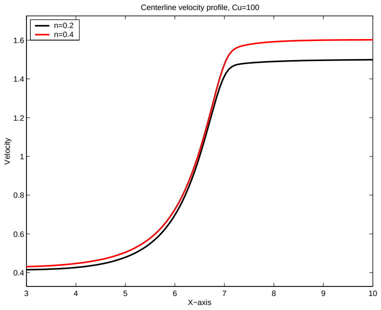

Figure 6.4: *The effect of varying n number on the calculated centerline velocity for $Cu = 100$, Carreau model.*

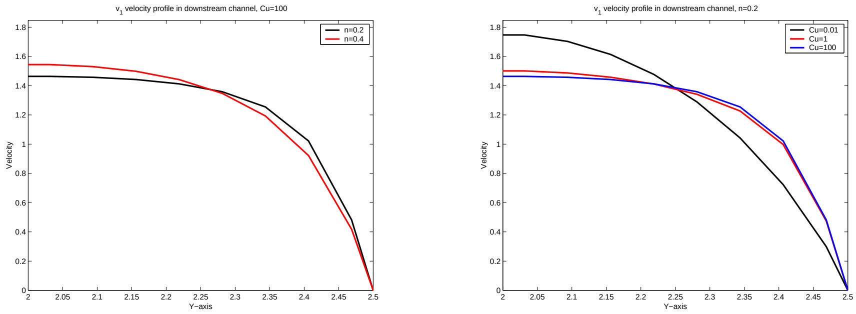

Figure 6.5: *The effect of decreasing n with $Cu = 100$ (left figure) and increasing Cu with $n = 0.2$ (right figure) on axial velocity profile at the contraction $x = x_c$.*

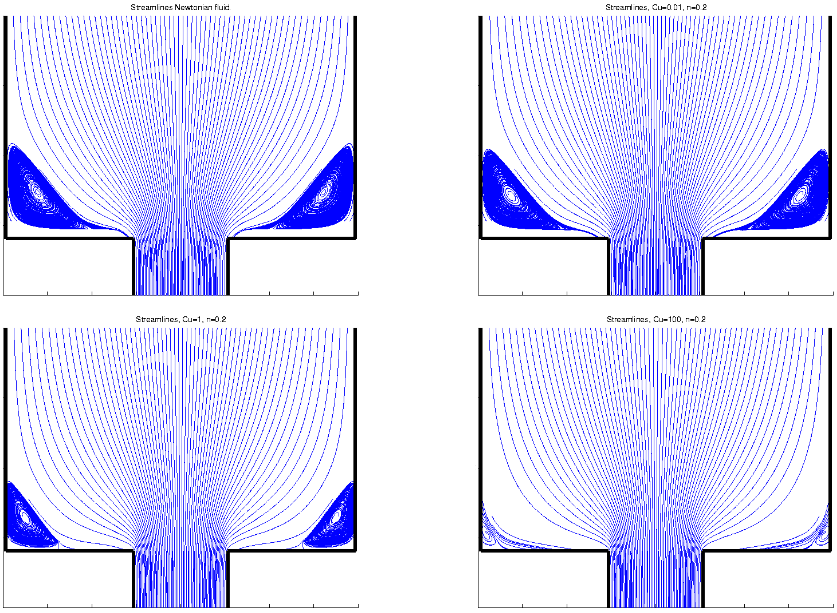

Figure 6.6: *Streamline plots for Newtonian fluid (upper left) and shear–thinning fluids of the same zero–shear rate viscosity for* $n = 0.2$ *and* $Cu = 0.01$ *(upper right),* $Cu = 1$ *(lower left),*$Cu = 100$ *(lower right).*

velocity profiles. Another well known fact is that shear–thinning decreases the sizes of the upstream vortices compared to the Newtonian fluids of the same zero–shear rate viscosity. The more shear–thinning fluid is, the smaller vortices it exhibits. These behavior is reflected in Figure 6.6. All the results presented in this Section are in very good agreement with the one showed in [25]. However, such results are contrary to the large vortices observed experimentally in viscoelastic fluids (see for example [7, 8, 11, 30]). One reason for it might be, that no elastic effects are considered. Another can be connected with a lack of extensional viscosity properties.

6.2 Extensional viscosity effect.

Many viscoelastic fluids exhibits much larger vortices in contraction flows compared with the Newtonian ones. During years, many experimental and numerical studies were performed to understand the reasons for such formation of vortices. First major work concerning the growth of vortices was presented by Cable and Boger (see [52] and references therein). In all tested fluids there, the vortices were always present and were increasing with higher flow rate. Authors have correlated such behavior with the elasticity of the fluid. The elasticity was expressed by Weissenberg number $We = \tau_{relax}\frac{\bar{U}}{L}$, where $\bar{U}$, L are average velocity and width of the downstream channel, respectively. τ_{relax} is a relaxation time that characterize the fluid. The observations made in [52] were, that fluids with grater elasticity exhibited larger vortices. Similar results were obtained by Nuygen and Boger in [30]. However, it was not supported by the work of White and Kondo [53] in experimental study of polymer melts. They have shown that low density polyethylene and polystyrene exhibit vortices but high density polyethylene, which is both elastic and shear–thinning, does not. They went on to say that the origins of vortices may be due to the development of large extensional stresses in the entry region and that vortices are formed as a stress relief mechanism. It is suggested, that only materials with an extensional viscosity that increases with extensional rate exhibit vortex enhancement, whereas no, or only small, vortices are present for the materials where extensional viscosity decreases or remains constant with elongational rate. It was confirmed in a paper of White and Baird [51], where two polymers, low density polyethylene and polystyrene, were considered. At about the same Weissenberg number the first polymer exhibited vortices and its growth while the second one did not. The difference, however, was in extensional flow properties. Vortex growth was associated with unbounded extensional growth in time at fixed extensional rate, whereas absence of vortices was associated with bounded extensional stress growth. To validate the performance of the model (2.26), presented in Section 2.3.2 and being able to distinguish between shear and extensional properties of the fluid, we choose shear–thinning and extensional–thickening liquid with the fluid parameters: $Cu = 1, n = 0.8, Cu_e = 1, n_e = 2$ and zero-shear viscosity

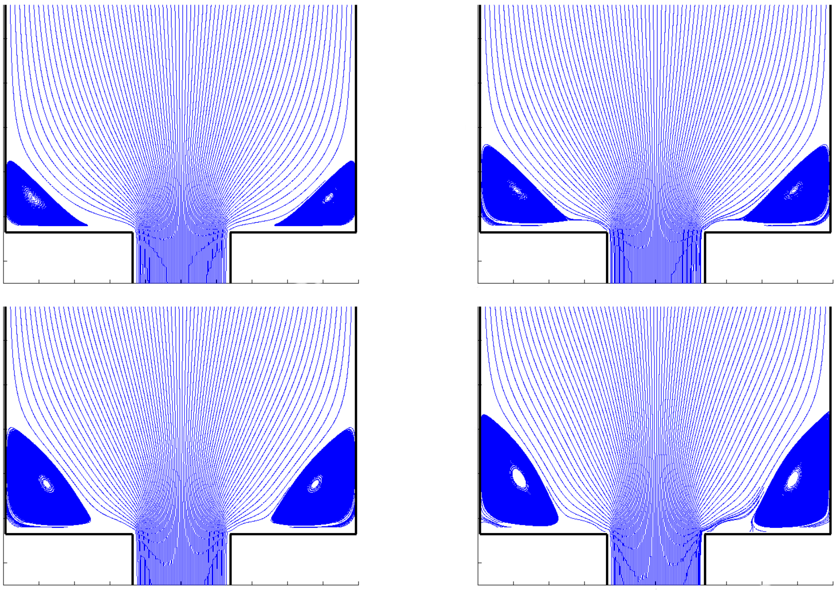

Figure 6.7: *Streamline plots for Newtonian fluid (upper left) and shear–thinning extensional–thickening fluids of the same zero–shear rate viscosity* $\eta_0 = 10[Pas]$ *for* $Cu = 1, n = 0.8, Cu_e = 1, n_e = 2$. *Inflow velocity is increased for upper right, lower left and lower right figure respectively.*

$\eta_0 = 10[Pas]$. Let us first consider 2D planar 4:1 contraction geometry (Figure 6.1). As it was discussed in previous Section, shear–thinning itself decreases the vortices for higher flow rates (see Figure 6.6). However, adding extensional–thickening into the model changes this situation. As indicated in Figure 6.7, increase of the flow rate

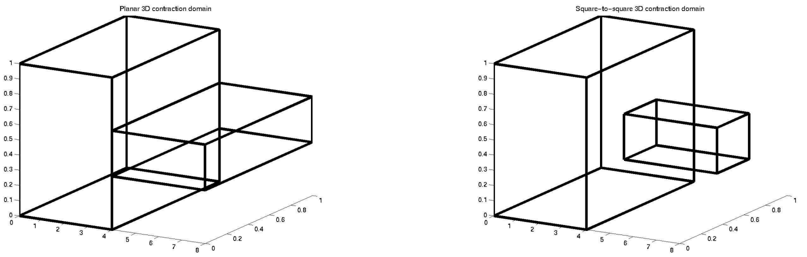
Planar 3D contraction domain
Square–to–square 3D contraction domain

Figure 6.8: *3D planar contraction (left figure) and 3D square–to–square contraction (right figure) geometries.*

does not decrease the vortices. In fact, they increase in size and the vortex center shifts toward the upstream channel. Moreover, the cell boundary changes its shape from concave for Newtonian fluid (upper left plot in Figure 6.7) to convex one (lower right plot in Figure 6.7). The situation becomes even more interesting in 3D planar contraction flows. Experimentalists have examined the effect of geometry on the formation of vortices.

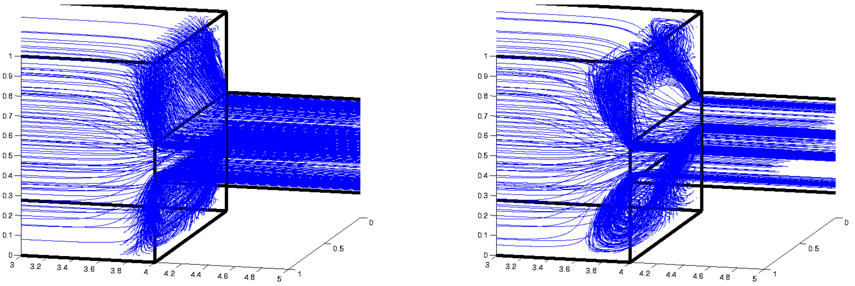

Figure 6.9: *3D planar contraction geometry. Streamline plots, Newtonian fluid (left figure) and extensional–thickening fluid (right figure).*

The most common test problems considered in the literature are planar and axisymmetric entry flows. For the Newtonian fluids the results show little, or no vortices at the corners for both cases. The observations show, that vortices in planar contraction, if appear, are much smaller than in axisymmetric domain. Similar situation happens in square–to–square geometry, which resembles axisymmetric one. Therefore, after Evans and Walters [11], as a test geometries 7:1 planar contraction and 7:1 square–to–square contraction domains, showed in Figure 6.8, are chosen. In Figure 6.9 streamlines plots in 3D planar contraction of the Newtonian (left plot) and the generalized Newtonian (right plot) fluids, at the same flow rate, are given. For the Newtonian fluid no vortices are noticeable, whereas for the latter one, that exhibits both shear–thinning and extensional–thickening, the vortices are clearly visible. The streamlines plots for the same test fluid in square–to–square geometry is shown in Figure 6.11. Again, the Newtonian fluids exhibit no vortices, contrary to the generalized Newtonian one. Moreover, the vortices are much larger comparing with 3D planar contraction case, what is in a very good qualitative agreement with

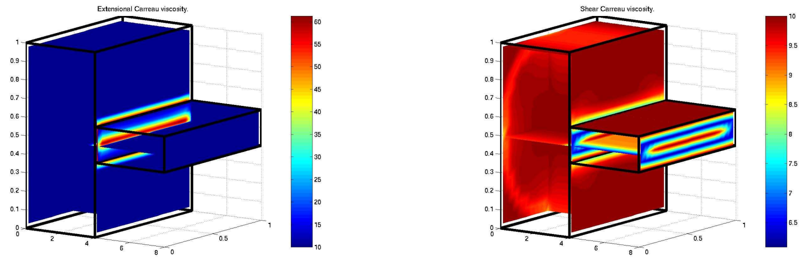

Figure 6.10: *3D planar contraction geometry. Extensional viscosity (left figure) and shear viscosity (right figure).*

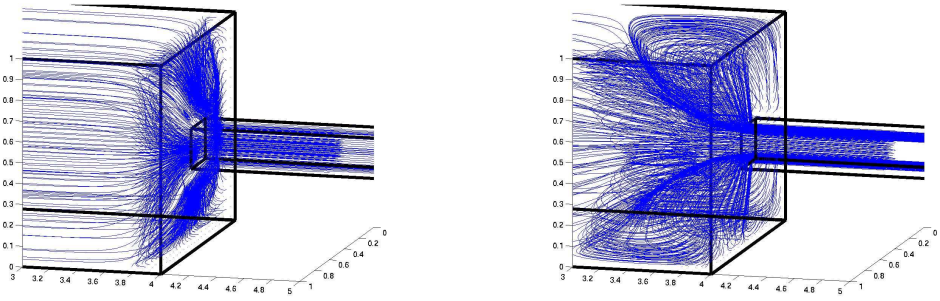

Figure 6.11: *3D square–to–square contraction geometry. Streamline plots, Newtonian fluid (left figure) and extensional–thickening fluid (right figure).*

experiments. Such behavior is correlated with much higher growth of the extensional stress in 3D square–to–square contraction, what is shown in Figures 6.10 and 6.12, where extensional $\eta_e(\dot{\epsilon})$ and shear $\eta(\dot{\gamma})$ viscosities are given. Moreover, for constant $\eta_e(\dot{\epsilon})$, or extensional–thinning fluids no growth of the vortices was detected. Above results may indicate the statement made by White and Kondo in [53], that only materials with clearly strong increasing of the elongational viscosities as a function of extensional rate exhibit vortices.

6.3 Simulations of viscoelastic fluids.

Current Section concerns the results obtained from simulations of dilute and concentrated polymer solutions, where the integral Oldroyd B and the integral Doi Edwards models are used respectively. First, we present comparison for Boger fluids, against experiments and numerical simulations performed by equivalent differential Oldroyd B model, in two and three dimensional contraction domains. Next, we present comparison between two methods used in approximation of the integral constitutive equations: Backward Lagrangian Particle Method (BLPM) and Deformation Field Method (DFM), discussed in Sections 4.1 and 4.2 respectively. Further, the simulation results of concentrated polymer solutions are shown. We obtain very good predictions of

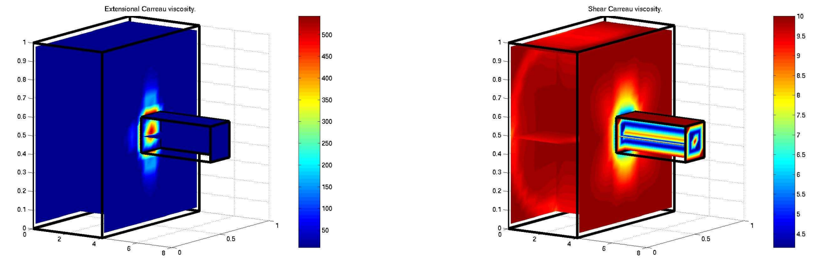

Figure 6.12: *3D square–to–square contraction geometry. Extensional viscosity (left figure) and shear viscosity (right figure).*

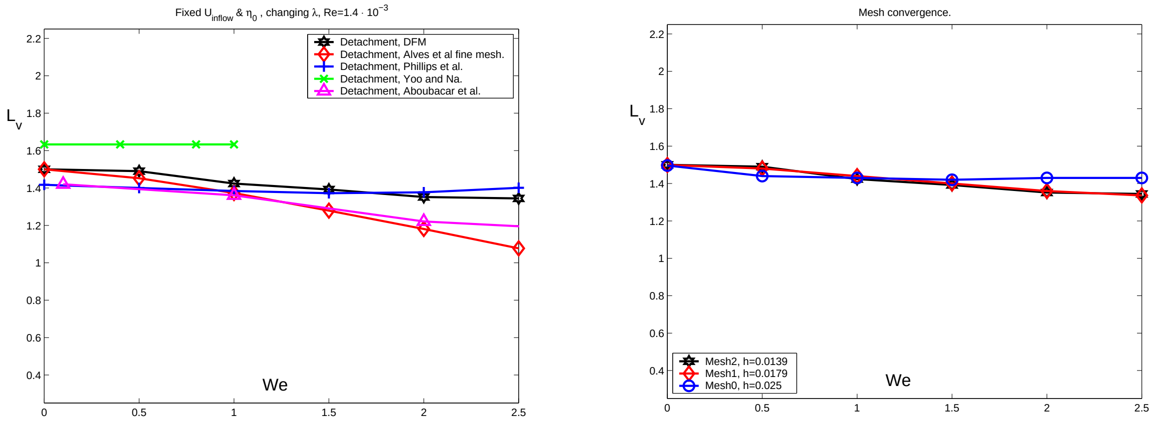

Figure 6.13: *Reattachement lengths against We numbers (left figure). Reattachement lengths against We numbers for different meshes (right figure).*

first normal stress difference and shear viscosity in simulations of polystyrene solution. Similar as in [46], adding chain length fluctuation to the original Doi Edwards model results in growing of the vortices in contraction flows at high chain stretch flow regimes.

6.3.1 Oldroyd B constitutive equation.

2D simulations.

Numerical computations are performed on regular Cartesian grid in 4:1 2D planar contraction domain (see Figure 6.1) with $L = 0.5$. We use the same space grid steps in all coordinates directions, $\delta x = \delta y$. In all simulation, the time step used is $\tau = 5 \cdot 10^{-3}$. The calculations have been performed for a range of Weissenberg numbers, which is defined as $We = \tau_{Oldr} \cdot \frac{\bar{U}}{L}$, where τ_{Oldr} is the relaxation time, $\bar{U}$ is the mean outflow velocity and L is the outflow channel width. To be consistent with experiments, the Weissenberg number was increased by increasing the inflow velocity (thus increasing $\bar{U}$) and keeping the ratio We/Re, where Re is the Reynolds number, constant. We also perform comparison with the simulations obtained in [1, 2, 3, 28, 38], where equivalent differential Oldroyd B model is used. In this case, we keep Re number constant and the change of the We number is due to the change of the relaxation time τ_{Oldr}. To quantify the size of the vortices we use the reattachement length L_v (see Figure 6.1). First, let us present quantitative comparison with the results obtained by differential counterpart.

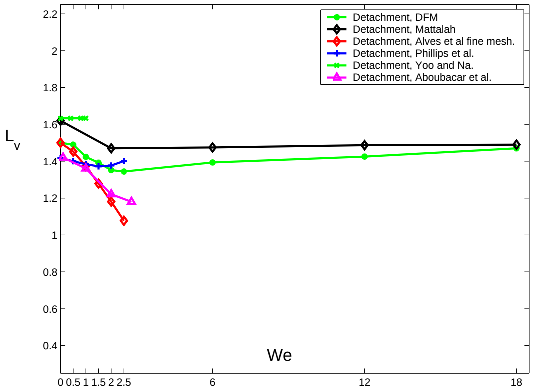

Figure 6.14: *Reattachement lengths against high* We *numbers.*

We compare our work with the results presented in [1, 2, 3, 28, 38]. To be consistent with those findings we have chosen the following fluid parameters: $\rho = 1[\frac{g}{cm^3}]$, $\eta_s = 1[Pa\ s]$ and $\eta_p = 8[Pa\ s]$. Time discretization of the integral constitutive equation is performed via Deformation Field Method (DFM), with non–equidistant time step used (detailed discussion see in Section 4.5). Comparison of Backward Lagrangian Particle Method (BLPM) and DFM, with equidistant time step used in approximation of the integral model, will be shown later. The simulations are performed for constant $Re = 10^{-3}$. We change We number by changing relaxation time τ_{Oldr}. The number of stored deformation fields in this case is $N_d = 100$. We observe decrease in the reattachement lengths as We is increased, what is consistent with [1, 2, 3, 28, 38]. Results for the vortex behavior are presented in Figure 6.13. The left figure shows comparison of the vortex behavior with the previously published values. As it was pointed out in [3], there is dispersion in these results, however our findings fall in the range of listed data. The right figure shows the mesh convergence results for three consecutive refined meshes with space grid steps Mesh0 –

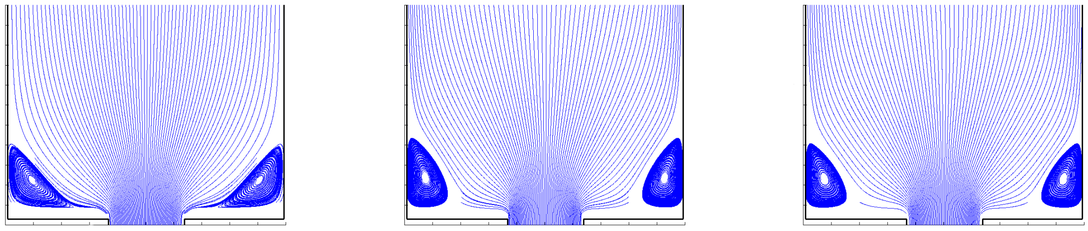

Figure 6.15: *We=0 (left figure), We=0.5 (center figure), We=1 (right figure).*

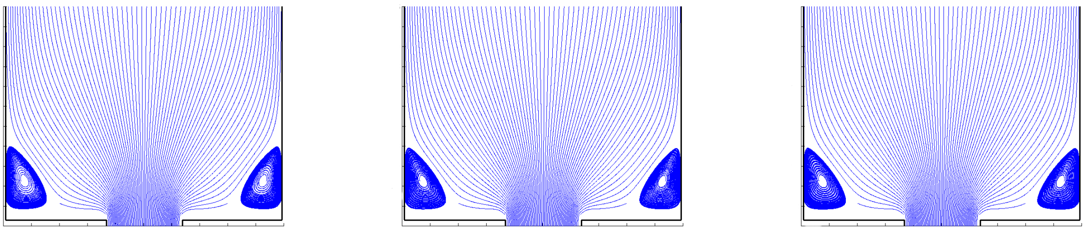

Figure 6.16: *We=1.5 (left figure), We=2 (center figure), We=2.5 (right figure).*

$\delta x = \delta y = 0.025$, Mesh1 – $\delta x = \delta y = 0.0179$, Mesh2 – $\delta x = \delta y = 0.0139$ respectively. Results for the coarsest mesh differ slightly from the finer and finest meshes, however the reattachement lengths in Mesh1 and Mesh2 almost coincide. In Figures 6.15 – 6.16 we show the streamlines for different We numbers. We would like to point out, that we have no problems with performing calculation for high We numbers. In Figure 6.14 we show comparison of the reattachement lengths with [28]. The reattachement lengths are a bit lower in our simulations, however for $We = 18$ results coincide. We have also checked that simulations are not time–dependent. In Table 6.1 we present reattachement lengths at $We = 1.5$ numbers obtained for three different time steps τ on the finest mesh. The change of time steps did not lead to any discernible change in the results.

We	1.5		
τ	10^{-3}	$2.5 \cdot 10^{-3}$	$5 \cdot 10^{-3}$
L_v	1.368	1.378	1.392

Table 6.1: *Time convergence for integral Oldroyd B model. Reattachement length* L_v *against time step* τ *at* $We = 1.5$.

Simulation of Boger fluid – B64.

In this subsection we present comparison of our results with experimental observations for Boger B64 fluid. The model parameters for this fluid, taken from [11], are: $\rho = 1.3[\frac{g}{cm^3}]$,$\tau_{Oldr} = 0.05[s]$, $\eta_s = 1.25[Pa\ \ s]$, $\eta_p = 1[Pa\ \ s]$. Now, according to the experiments both Re and We numbers are changed by increasing the inflow velocity. Good qualitative agreement with the experiments, namely growth of the vortices is observed. In Figure 6.17, we show the streamline contours of our simulations for different We and Re numbers. Starting from $We = 0$ and $L_v = 1.5$ (Newtonian case), we have noticed that the reattachement length increases up to $L = 2.47$ for $We = 0.2$. At the same time, at the $We = 0$ the cell boundary is concave with the vortex center near the corner of the upstream tube. When we increase the flow rate, the vortex center shifts towards the tube entrance, while the cell boundary changes its shape from concave (Newtonian fluid) to convex one. Another important issue is the difference in the pressure drop in comparison with the Newtonian fluid of the same viscosity. S.Nigen and K.Walters (see [31]) show in their experimental work that as long as the vortices do not grow, it is almost impossible to distinguish between pressure drop/flow rate data for Boger and Newtonian fluid. The situation changes when the vortices start to grow. Then, higher pressure drop is seen in the case of Boger fluid. We also predict this behavior, what is illustrated in Figure 6.18. The data, similarly as in [2], are scaled by $\frac{\eta_t}{L} \cdot U_{REF}$, where $\eta_t = \eta_s + \eta_p$ and U_{REF} is the mean outflow velocity at $We = 1$. Summarizing, a very good qualitative agreement with experimentalists (see [7, 8, 11, 30, 31]) is achieved.

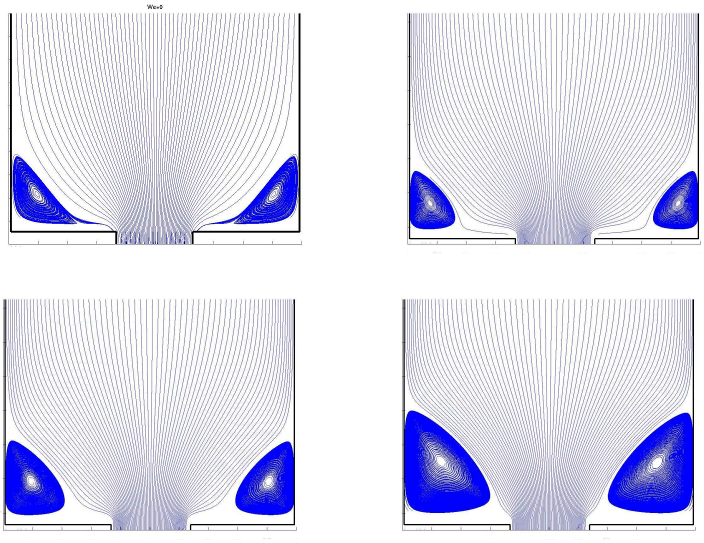

Figure 6.17: DFM simulations.$We = 0$ (upper left figure), $We = 0.05$, $Re = 0.0125$ (upper right figure), $We = 0.1$, $Re = 0.025$ (lower left figure), $We = 0.2$, $Re = 0.05$ (lower right figure).

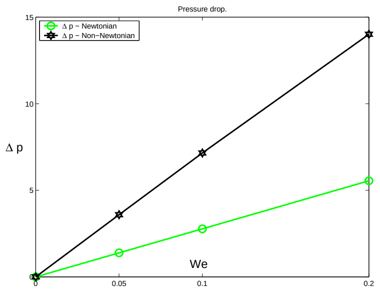

Figure 6.18: *Pressure drops of the Newtonian (green color) and Boger (black color) fluids against* We *numbers. The Newtonian pressure drops were sampled at the appropriate mean outflow velocities* $\bar{U} = \frac{We \cdot L}{\tau_{Oldr}}$

Comparison of DFM and BLPM.

Very interesting questions concerning the time discretization of the integral constitutive equation (2.38 or 2.43) has already appeared in Section 4.5. The first question, that appeared there was, how the non–equidistant time step used in approximation of (2.38 or 2.43), discussed in Section 4.5, influences the results. The second one, also took up in Section 4.5, was connected with the loss of locality, when Deformation Field Method is used in calculation of the extra stress tensor. To give an answer to the first question, we have performed simulations of the viscoelastic fluid with $\rho = 1[\frac{g}{cm^3}]$, $\eta_s = 1[Pa\ s]$ and $\eta_p = 8[Pa\ s]$, however, the equidistant time step was used in approximation of the time integral (2.43). The comparison with non–equidistant time step approximation of (2.43) was performed for various Weissenberg numbers ($We = 0.5, 1, 1.5, 2, 2.5$), where the change of We number was due to the change of the relaxation time τ_{Oldr}. In the case of equidistant time step approximation, at $We = 2.5$ we had to store $N_d = 1200$ deformation fields, contrary to $N_d = 100$ in the non–equidistant case, to be able to span more than 90% of the fluid history. This ended up in a drastic growth of the memory requirements. For all We numbers, the relative differences in velocities and pressure fields, resulting from the simulations with equidistant and non–equidistant approximation of the memory integral (2.43), were less than one percent. In Backward Lagrangian Particle Method extra stress

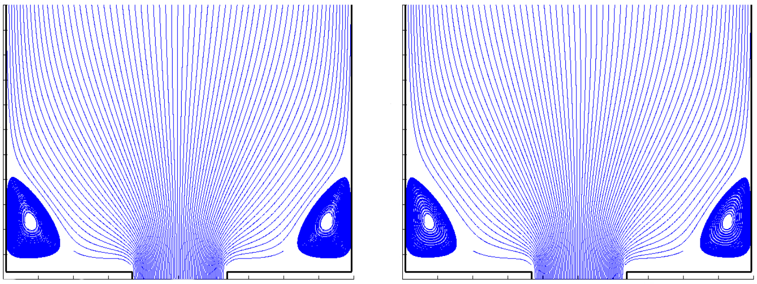

Figure 6.19: *Deformation Field Method (left figure), Backward Lagrangian Particle Method (right figure) at* $We = 1$.

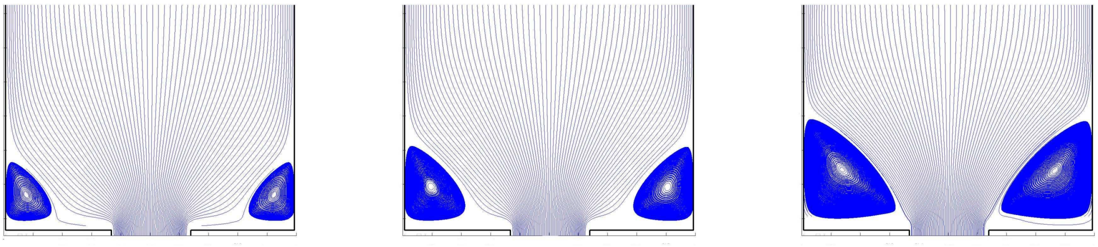

Figure 6.20: *BLPM simulations of Boger B64 fluid.* $We = 0.05$, $Re = 0.0125$ *(left figure),* $We = 0.1$, $Re = 0.025$ *(center figure),* $We = 0.2$, $Re = 0.05$ *(right figure).*

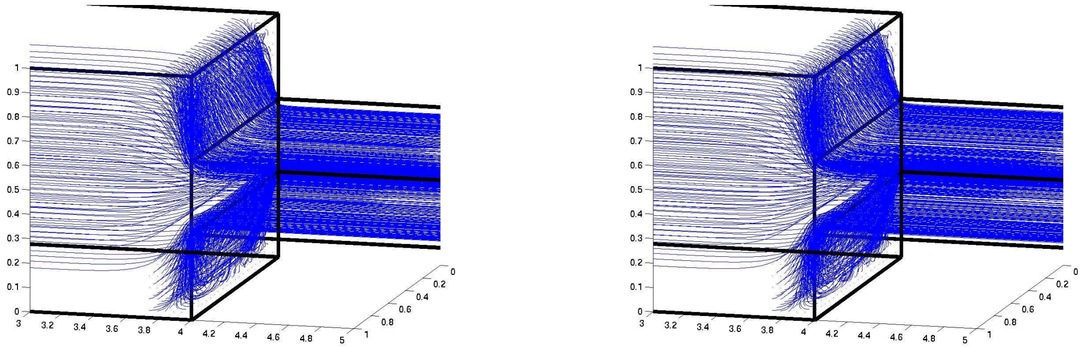

Figure 6.21: *Streamlines contours in 3D planar 4:1 contraction domains. Newtonian flow (left figure) and Boger B64 flow (right figure).*

is calculated along the particle trajectories. This assures that the calculations are local, i.e. the stress is carried by the particle. Performing both DFM and BLPM has answered the second question posed at the beginning of this subsection. Comparison were performed for two different viscoelastic fluids. The first one, described at the beginning of Section 6.3.1, with $\rho = 1[\frac{g}{cm^3}]$, $\eta_s = 1[Pa\ s]$, $\eta_p = 8[Pa\ s]$ and relative large relaxation time $\tau_{Oldr} = 1$, and second one (B64), also discussed in Section 6.3.1, with fluid parameters $\rho = 1.3[\frac{g}{cm^3}]$, $\eta_s = 1.25[Pa\ s]$, $\eta_p = 1[Pa\ s]$ and small relaxation time $\tau_{Oldr} = 0.05$. In Figure 6.19, we show streamlines contours for the first fluid at $We = 1$. Both methods give almost identical vortex behavior. Figure 6.20 shows streamline contours from simulations of Boger B64 fluid, when BLPM was used. Again, no differences to Figure 6.17 has been noticed. In all the calculations the maximum relative difference in velocities was less 4%, and the maximum relative difference in pressure was less than 3%. This results bring us to the conclusion, that both methods perform similar results, however BLPM is much more memory and time consuming. Computational time needed to perform BLPM for $\tau_{Oldr} = 1$ and final time $T = 10$ was about 24 hours, while DFM performs the same calculations in about 1 hour.

3D simulations.

In this subsection, we will show that the influence of the geometry on the vortex activity of constant viscosity–elastic (Boger) fluids is reflected in our simulations, when the integral version of the Oldroyd B model is used. The computational domains, planar 3D contraction and square–to–square contraction, correspond to the ones chosen in the experimental work by R.E Evans and K. Walters [11]. The geometries are shown in Figure 6.8. In both cases the computational domain is closed in the cuboid of the size $[0,8] \times [0,1] \times [0,1]$ in $x-, y-, z-$ directions respectively. The contraction wall is located along $x = 4$. Again, we use a regular Cartesian grid with space steps $\delta x = \delta y = \delta z = 0.05$. The Weissenberg number is now defined as $We = \tau_{Oldr} \cdot \frac{\bar{U}}{L}$. However, to be consistent with the results presented in [11], $\bar{U}$ denotes now mean

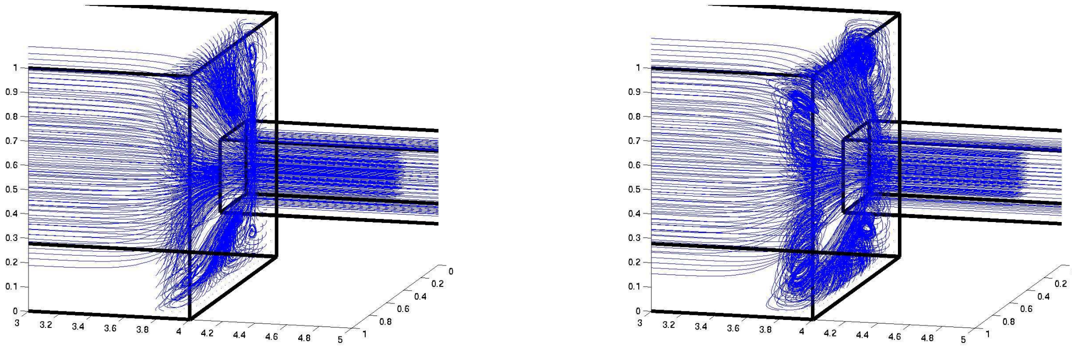

Figure 6.22: *Streamlines contours in 3D square–to–square 4:1 contraction domains. Newtonian flow (left figure) and Boger B64 flow (right figure).*

velocity of the upper section of the domains and L is the width of it.

Simulation of Boger fluid – B64.

The experiments shows that, for Boger fluids, it is as difficult to avoid enhancement of the vortices in square-to-square contractions, as it is to observe them in 3D planar contractions. The vortices in planar contraction, if visible, are reduced in size and intensity. Fluid parameters for Boger B64 fluid are: $\rho = 1.3[\frac{g}{cm^3}]$,$\tau_{Oldr} = 0.05[s]$, $\eta_s = 1.25[Pa\ s]$, $\eta_p = 1[Pa\ s]$. The experiments showed no vortex activity in planar contraction. This is in agreement with our findings as shown in Figure 6.21 for contraction ratio 4:1. The situation changes in square-to-square contraction domain. In this case we observe, in agreement with experiment, vortex activity. In Fig. 6.22 clear vortices are visible, contrary to the Newtonian case. Both calculations were done for $We = 0.039$ which is in the range of the values reported in [11].

6.3.2 Doi Edwards constitutive equation.

In current subsection we present results obtained form simulations of concentrated polymer solutions. Two nearly monodisperse solutions, with an experimental data available in [6, 19] and [46], are taken into considerations. The extra stress tensor is modeled by the most successful tube model described by the time integral constitutive equation (2.38), introduced by Doi and Edwards (in 1986) , as well as by equation (2.42) where the possible stretch of polymers is included. Both models are evaluated in rheometrical and complex flows. It is well known, that viscoelastic fluids, contrary to the Newtonian one, exhibit non zero first normal stress difference $N_1 = T_{xx} - T_{yy}$ in steady state–state shear flows. In [6], Bhattacharjee et al. have published experimental data in both shear and extension for a 10% solution of a $3.9 \cdot 10^6$ molecular weight polystyrene in diethyl phthalate. In addition, they have derived the values for the Doi Edwards model parameters, which are: $\rho = 1[\frac{g}{cm^3}]$, $G_e = 3083[Pa]$, $\tau_d = 8.61[s]$, $\tau_R = 0.282[s]$, $\lambda_{max} = 13.6$ and $\eta_0 = 4570[Pa\ s]$.

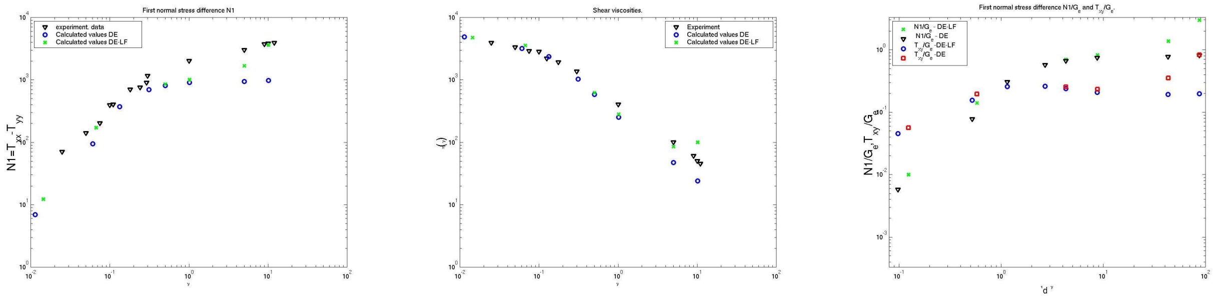

Figure 6.23: *Rheological results for polystyrene solution. Left figure presents comparison of* $N_1 = T_{xx} - T_{yy}$ *between the experimental data (black triangles) given in [6] and calculated results (blue circle - original Doi Edwards model, green star - Doi Edwards with chain stretch) against shear–rate* $\dot{\gamma}$*. Central figure presents comparison of the shear viscosity, defined as* $\eta(\dot{\gamma_{xy}}) = \frac{T_{xy}}{\dot{\gamma_{xy}}}$*, against shear–rate* $\dot{\gamma_{xy}}$ *between experimental measurements and calculated results. Right figure shows calculated data of* N_1/G_e *and* T_{xy}/G_e *against* $\tau_d\dot{\gamma}$*.*

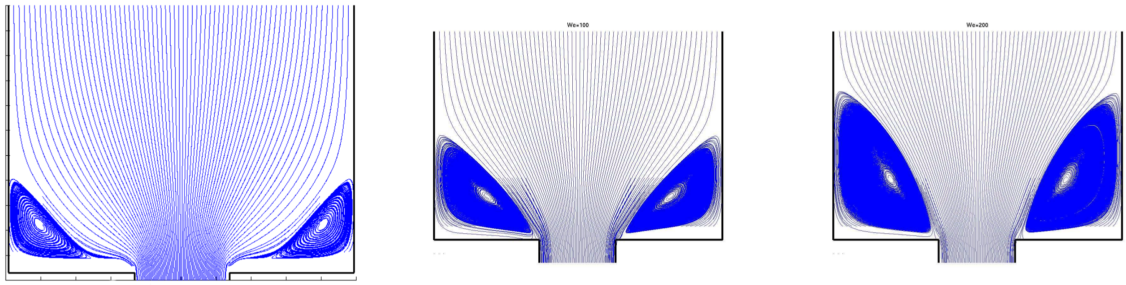

Figure 6.24: *Streamlines plots for polystyrene solution at* $We = 0$ *(left figure),* $We = 100$ *(center figure) and* $We = 200$ *(right figure).*

For a numerical comparison of simulation results and rheological data of the first normal stress difference N_1, as well as the shear viscosity defined as $\eta(\dot{\gamma}_{xy}) = \frac{T_{xy}}{\dot{\gamma}_{xy}}$, we choose the plane Couette flow. Therefore, as a test geometry the channel with upper wall moving at a constant velocity and static lower wall is taken. At the inflow linear velocity profile, known analytically form the Newtonian Couette flows, is imposed. At the outflow zero Neumann condition for velocities is satisfied. Note, that in these comparison equations (3.15), (3.16) supplemented by models (2.38) or (2.42) are solved, and the simulations are performed till the steady state is reached. For the comparison, similarly as in [6], the modulus G_e is scaled in such a way, that the zero–shear viscosity of the model coincides with the zero–shear viscosity of the data, i.e. $\eta_0 = 4570[Pa\ s]$. In Figure 6.23 comparison between experimental measurements and the values obtained from the simulations, sampled at the middle point of the channel, and thus being at less influenced by the boundaries, are shown. Left plot shows comparison of the first normal stress difference $N_1 = T_{xx} - T_{yy}$. At small shear–rates, where stretch of the chains brings no significant effects, both integral

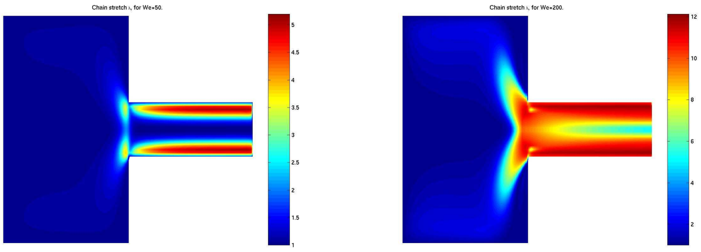

Figure 6.25: *Polystyrene solution, the chain stretch at* $We = 50$ *(left figure) with* $max(\lambda) = 5.096$ *and at* $We = 200$ *(right figure) with* $max(\lambda) = 12.096$

models show almost the same results, as expected. However, for stretch dominated regimes results start to differ. At such regimes, the original Doi Edwards model (2.38) gives lower predictions than experimental values, while the Doi Edwards model with the chain stretch (2.42) exhibits much higher values of N_1. At about $\dot{\gamma}_{xy} = 10$, experimental and calculated values coincide. The same observations are showed in [6, 46], where authors have used differential approximation to the integral models used here. Central plot from Figure 6.23 presents comparison of the shear viscosity sampled experimentally and calculated through the formula $\eta(\dot{\gamma}_{xy}) = \frac{T_{xy}}{\dot{\gamma}_{xy}}$. Again, very good qualitative agreement with experiment is achieved. Right plot from Figure 6.23

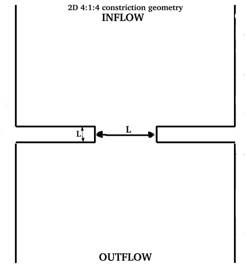

Figure 6.26: *2D planar contraction–expansion domain.*

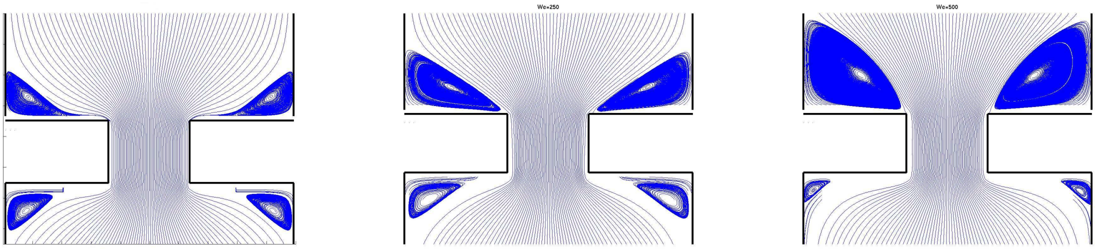

Figure 6.27: *Streamlines plots for polystyrene solution in 4:1:4 constriction at* $We = 0$ *(left figure),* $We = 250$ *(center figure) and* $We = 500$ *(right figure).*

presents the values of N_1/G_e and T_{xy}/G_e against $\tau_d\dot{\gamma}$ obtained from our simulations.

Unfortunately, experimental results for shear T_{xy} stress for considered polystyrene solution are not avaliable, however the tendency of both curves agrees qualitatively with the results of various different polymer solutions shown, for example, in [19, 20, 29, 34]. We conclude, that both models are in very good qualitative agreement with experiments in the range of shear–rates for which experimental data are available.

In the simulations of polystyrene solution in 4:1 planar contraction domain, given in Figure 6.1, a stable calculations could be performed for high Weissenberg numbers, up to $We = \mathcal{O}(10^2)$. Now, the Weissenberg number is defined as $We = \frac{\tau_d \cdot \bar{U}}{L}$, where τ_d is relaxation time, $\bar{U}$ average upstream velocity and L the width of the upstream channel. Figure 6.24 displays the streamlines for various We numbers. No vortex activity has been noticed when original Doi Edwards models was used, and the vortices are of the same size as for Newtonian fluid with the same viscosity ($We = 0$). However, the situation changes when the chain stretch is considered in addition. At the flows, where high stretch regimes are not achieved, the vortex activity coincide with the ones obtained by the original Doi Edwards model. However, high stretching of the chains, shown in Figure 6.25, introduce differences in vortices. At $We = 200$, the vortices are much bigger than in the Newtonian case. Similarly as for Boger fluid B64, the vortex center shifts towards the tube entrance, while the cell boundary changes its shape from concave to convex one. Unfortunately, there are no numerical simulations of considered here polystyrene solution, published for 4:1 planar contraction geometry. Therefore no comparison can be performed.

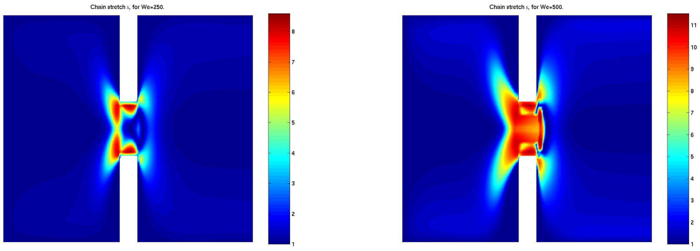

Figure 6.28: *Polystyrene solution, the chain stretch at* $We = 250$ *(left figure) with* $max(\lambda) = 8.494$ *and at* $We = 500$ *(right figure) with* $max(\lambda) = 11.436$

However in [46], P. Wapperom and R. Keunings have presented simulation results of this polystyrene solution in 4:1:4 axisymmetric sudden contraction–expansion geometry. This gives us a chance for further justification of our simulations. One has to point out, that there are some differences in both simulations. P. Wapperom and R. Keunings use differential approximation to integral Doi Edwards model with additional effects of chain length fluctuation, convective constraint release and intrinsic friction of the chain. The last two effects are not included in our case. Moreover, they use geometry with rounded corners, thus avoiding singularity points, contrary to the simulations presented here (see Figure 6.26). In Figure 6.27, we show streamline contour plots near constriction. No vortex activity is noticeable as long as the high stretch regimes are not reached (up to $We = 100$), and the vortices are of the same

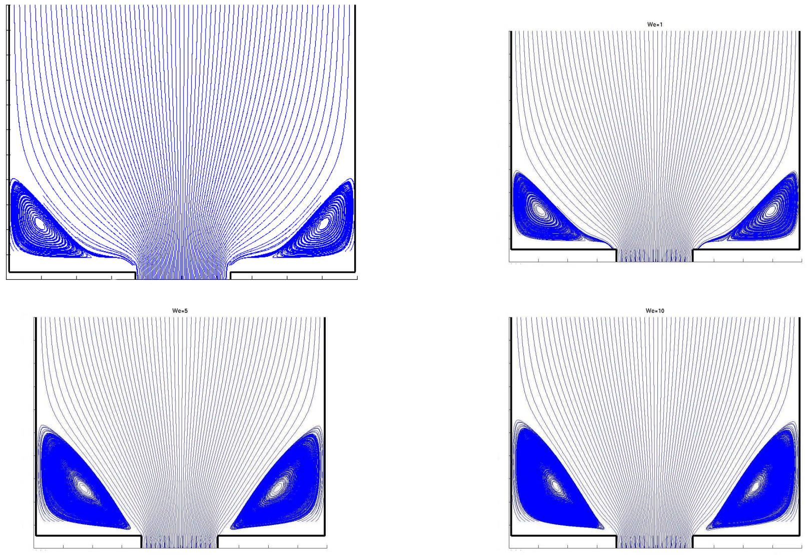

Figure 6.29: *Streamline plots for polybutadiene solution at* $We = 0$ *(upper left),* $We = 1$ *(upper right),* $We = 5$ *(lower left) and* $We = 10$ *(lower right).*

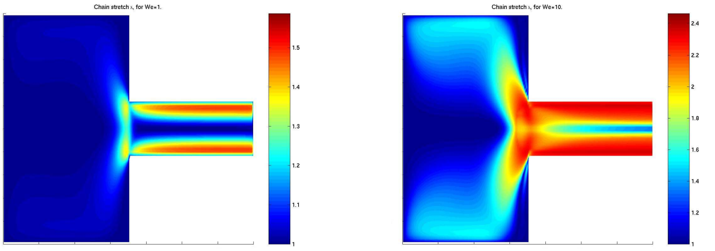

Figure 6.30: *Polybutadiene solution, the chain stretch at* $We = 1$ *(left figure) with* $max(\lambda) = 1.487$ *and at* $We = 10$ *(right figure) with* $max(\lambda) = 2.363$

order as in the Newtonian case. This is what one should expected, since similar behavior was found in [46], where vortices were decreasing with higher We numbers. As indicated in [46], such decreasing of vortices happens at convected constraint release dominated regimes, which is not considered here. In [46], authors could perform stable steady–state simulations up to $We = 50$. In our simulations, we could achieved stable solutions up to $We = \mathcal{O}(10^3)$. Figure 6.28 shows the chain stretch for various We numbers. It is seen there, that the high stretch regimes appear for much larger We numbers than achieved in [46]. Similar as in 4:1 contraction geometry, those high stretch regimes are correlated with the vortex growth (see Figure 6.27). To be consistent with the definition of We used in [46], despite τ_d we have used equilibrium relaxation time $\tau_{eq} = \tau_d + \tau_R$, and half of the width of the upstream channel L. In 4:1:4 constriction geometry $\bar{U}$ denotes average velocity in smaller chanel at constriction.

In [19], G. Ianniruberto and G. Marrucci have identified parameter values for

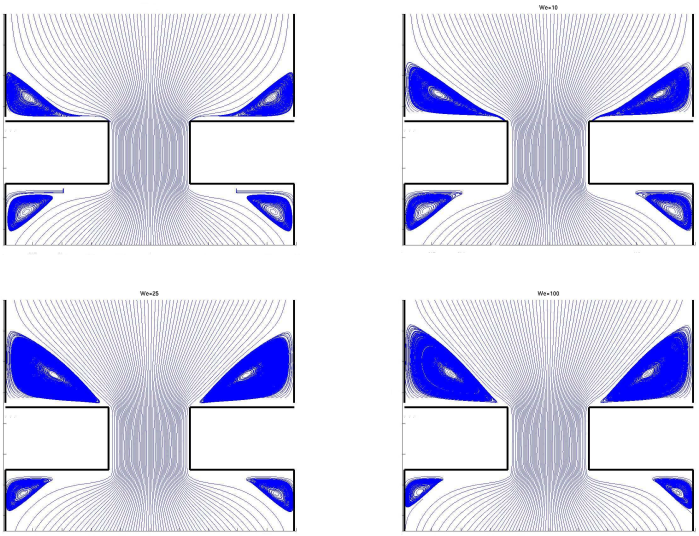

Figure 6.31: *Streamlines plots for polybutadiene solution in 4:1:4 constriction at* $We = 0$ *(upper left),* $We = 10$ *(upper right),* $We = 25$ *(lower left) and* $We = 100$ *(lower right).*

a nearly monodisperse 7.5% polybutadiene solution of $3.5 \cdot 10^5$ molecular weight in hydrocarbon oil. These are: $\rho = 1[\frac{g}{cm^3}]$, $G_e = 9000[Pa]$, $\tau_d = 0.75[s]$, $\tau_R = 0.5[s]$, $\lambda_{max} = 2.5$ and $\eta_0 = 1750[Pa\ s]$. First, we present simulation results of this fluid obtained in 4:1 planar contraction geometry. For the constitutive equation we have chosen the integral Doi Edwards model supplemented by chain stretch λ. In Figure 6.29, the streamlines at different We numbers are displayed. We observe clear growth of the vortices with increase of We numbers. Such growth of the vortices is correlated with high chain stretch regimes. The use of original Doi Edwards model does not give such results. In such case, the vortices do not change and are of the size of Newtonian fluid with the same viscosity. In Figure 6.30, we present the chain stretch at various Weissenberg numbers. High stretch regime is already obtained for $We = 10$ with the maximum value $max(\lambda) = 2.363$. Comparing Figures 6.25 with 6.30 we can see, that the chains start to stretch in downstream channel for both, polystyrene and polybutadiene, solutions. Later, further increase of We number results in stretching of chains also in the upstream channel.

Similar observation are obtained in 4:1:4 constriction geometry. Again, growth of the vortices at the contraction wall, when increasing Weissenberg number, is predicted. It is displayed in Figure 6.31. The same effect is reported in [46], however there authors could obtain stable steady state results up to $We = 15$. Using integral Doi Edwards model allows to preform stable simulations for much higher Weissenberg numbers, up to $We = \mathcal{O}(100)$. The contours of the chain stretch λ are shown in Figure 6.32. At Weissenberg numbers between 25 and 100 polymer chains become highly stretch (up to $max(\lambda) = 2.42$ at $We = 100$) in the constriction region. As the We number is increased, this region extend further upstream and downstream, and the

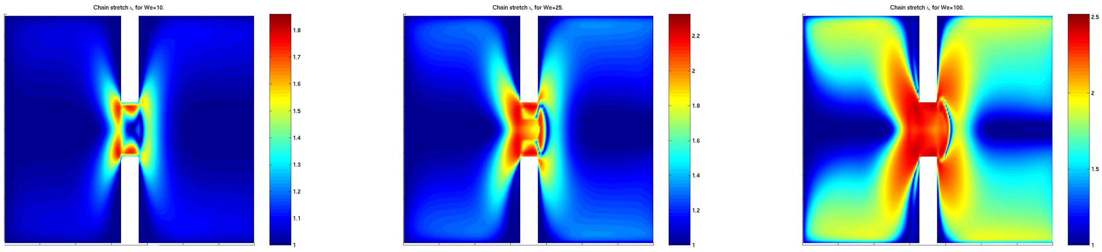

Figure 6.32: *Chain stretch* λ *for polystyrene solution in 4:1:4 constriction at* $We = 10$ *with* $max(\lambda) = 1.76$ *(left figure),* $We = 25$ *with* $max(\lambda) = 2.226$ *(center figure) and* $We = 100$ *with* $max(\lambda) = 2.42$ *(right figure).*

chain stretch is already significant far away from the constriction. Moreover, almost full chain stretch is attained at $We = 100$.

6.4 Performance of iterative solvers.

Current subsection concerns the performance of different solvers applied in the simulations of liquid polymer flows. As already discussed in Chapter 2 the viscoelastic material do not obey Newtonian behavior, thus it is necessary to use specific material models in order to close the system of governing (continuity (3.15) and momentum (3.16)) equations. In particular, we focus here on the generalized–Newtonian fluids, modeled by the shear–dependent viscosity Carreau constitutive equation (2.21), or its generalization adding extensional–viscosity properties (see equation (2.25)). As it was already discussed such variable viscosity models lead to strong coupling of the momentum equations (3.16) through the viscous terms, which is not a case for the Newtonian fluids. An unsteady problem, starting simulations from the liquid being at rest and calculating till reaching the steady state, is solved. During such process the viscosity changes couple of orders of magnitudes. At each time step the continuity (3.15) and the momentum (3.16) equations have to be solved. In Chapter 5 analytical results concerning different algorithms and preconditioning techniques, that can be used in simulations, are presented. Before going to further discussion, let us first recall some nomenclature that will be used in next considerations. We will denote by *SIMPLE–like projection method* the projection method, discussed in Section 3.2, where the mixed derivatives appearing in the discretization of the momentum equations are treated in an explicit manner, i.e. taken from the previous time step iteration. Next, *coupled–momentum projection method* denotes the projection method, presented in details in Section 3.2.1, where all terms from the viscous operator are discretized in an implicit manner. Finally, *fully coupled method* denotes the method, where the coupled momentum and continuity equations are solved together.

Comparison of different solution algorithms

First, let us compare the performance of three solution algorithms listed above, i.e. SIMPLE–like projection method, coupled momentum projection method and fully coupled method, in terms of stability of calculations that could be achieved. For that, the following numerical experiment was performed. Three fluids with different viscosity properties, namely shear–thinning (with fluid parameters $Cu = 1.0$, $n = 0.2$), shear–thinning and extensional–thickening ($Cu = 1.0$, $n = 0.2$, $Cu_e = 1.0$, $n_e = 0.4$), and finally shear–thickening and extensional–thinning ($Cu = 1.0$, $n = 1.1$, $Cu_e = 1.0$, $n_e = 0.4$), were selected. The simulations were performed in planar 4:1 contraction domain (Figure 6.1) with the time step $\tau = 10^{-3}$. The only parameter that was varied was the zero shear–rate viscosity η_0. In all the cases the flow was such, that the mean velocity in the smaller outflow channel $\bar{U} = 10[\frac{cm}{s}]$ (if the steady state was reached). In Tables 6.2, 6.3, 6.4 the simulations result of the listed above fluids are presented, where we indicate for which zero shear–rate viscosity η_0 the corresponding solution techniques converged (denoted by Con.) and gave steady state solution, or diverged (denoted by Div.) and no solution was obtained. In all the cases, as expected, the most stable behavior was obtained by the coupled momentum projection method and the fully coupled method. For all range of the zero shear–rate viscosities, that has been taken into considerations, stable steady–state solutions were reached. The least stable was the SIMPLE–like solution algorithm. As expected, explicit discretization of the mixed derivatives has led to stability problems. The stable simulations could be obtained for all considered η_0 only in the case of the shear–thinning and extensional–thickening fluid. However, for shear–thinning fluid (Table 6.2), and for shear–thickening and extensional–thinning one (Table 6.4), stable calculations were obtained up to $\eta_0 = 10$ and $\eta_0 = 1$, respectively. This can be explained from the fact, that for the first fluid the extensional viscosities that enter diagonal blocks of the viscous operator $\eta_e \geq \eta_0$ (since the fluid is extensional–thickening) and the shear viscosities $\eta_s \leq \eta_0$ enter the off–diagonal ones. This results in stronger block diagonal dominance of the viscous operator, which is not the case for the last two remaining fluids. We have also observed, that for large viscosities ($\eta_0 \geq$

η_0	1	10	50	100	500	1000	2500	5000	7500
SIMPLE like	Con.	Con.	Div.	Div.	Div.	Div.	Div.	Div.	Div.
Coupled momentum	Con.	Con.	Con.	Con.	Con.	Con.	Con.	Con.	Con.
Fully coupled	Con.	Con.	Con.	Con.	Con.	Con.	Con.	Con.	Con.

Table 6.2: *Stability performance of three different solution techniques for shear–thinning fluid, modeled by the Carreau constitutive equation (2.22), with the fluid parameters* $Cu_s = 1.0$*,* $n_s = 0.4$*.*

η_0	1	10	50	100	500	1000	2500	5000	7500
SIMPLE like	Con.	Con.	Con.	Con.	Con.	Con.	Con.	Con.	Con.
Coupled momentum	Con.	Con.	Con.	Con.	Con.	Con.	Con.	Con.	Con.
Fully coupled	Con.	Con.	Con.	Con.	Con.	Con.	Con.	Con.	Con.

Table 6.3: *Stability performance of three different solution techniques for shear–thinning and extensional–thickening fluid, modeled by the extension of the Carreau constitutive equation (2.26), with the fluid parameters* $Cu_s = 1.0$, $n_s = 0.4$, $Cu_e = 1.0$, $n_e = 1.1$.

η_0	1	10	50	100	500	1000	2500	5000	7500
SIMPLE like	Con.	Div.	Div.	Div.	Div.	Div.	Div.	Div.	Div.
Coupled momentum	Con.	Con.	Con.	Con.	Con.	Con.	Con.	Con.	Con.
Fully coupled	Con.	Con.	Con.	Con.	Con.	Con.	Con.	Con.	Con.

Table 6.4: *Stability performance of three different solution techniques for shear–thickening and extensional–thinning fluid, modeled by the extension of the Carreau constitutive equation (2.26), with the fluid parameters* $Cu_s = 1.0$, $n_s = 1.1$, $Cu_e = 1.0$, $n_e = 0.4$.

5000) the convergence rate of the projection type methods (if they have converged) was very poor, and high number of the non–linear iterations had to be performed in order to get satisfactory result. It was not the case for the fully coupled method. However, as discussed in Section 3.3, this method needs more memory compared to the projection–type methods. Moreover, it takes more time to solve once coupled momentum and continuity equations, than performing once the whole projection–type procedure, i.e. predicting velocities, calculating pressure corrections, correcting pressure and velocities (for detailed description see Section 3.2.1). Therefore, fully coupled methods require good preconditioning techniques in order to reduce the time needed to perform the simulations.

Performance of the preconditioning techniques for fully coupled method

The numerical experiments are performed on two simple test geometries, a channel with a contraction and expansion in the middle and a 2:1 planar contraction (see Figure 6.33). We will consider the fully coupled solver only, supplemented by different preconditioning strategies discussed already in Chapter 5. We consider five different

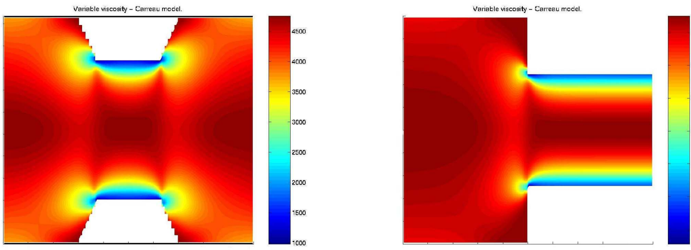

Figure 6.33: *The variable viscosities in Venturi domain (left plot) and 2:1 planar contraction domain (right plot).*

preconditioners. Two of them are applied to the untransformed system, given by the equation (3.2), of the form

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix},$$

arising after linearization and discretization of coupled continuity and momentum equations. These are the block Gauss–Seidel $\mathcal{D}_{GS}$ (denote by **UBGS**, for details see Section 5.2.1) and the indefinite block triangular $\mathcal{D}_{IBT}$ (**UIBT**, discussed in Section 5.2.2). The three remaining are applied to first transformed system (3.2). These preconditioners are the block diagonal $\bar{\mathcal{D}}_D$ (**BD**), the block lower triangular $\bar{\mathcal{D}}_{LT}$ (**BLT**) and the block Gauss–Seidel $\bar{\mathcal{D}}_{GS}$ (**BGS**). The discussion concerning those preconditioners and the way the system (3.2) is transformed was already presented in Section 5.3. All preconditioners used in simulations are collected in Table 6.5. For a linear solver two iterative methods are used, namely BiCGstab (bi–conjugate gradient method, stabilized) and GMRES(m) (generalized minimal residual method, where m denotes the number of sequences used). The performance of the iterative solvers is investigated with respect to grid size and the time step used in simulations. More precise information about the tests being performed is given in the following Table 6.6. As a test fluid we have chosen a 10% solution of a $3.9 \cdot 10^6$ molecular weight polystyrene in diethyl phthalate. For a constitutive relation, the generalized Newtonian Carreau viscosity model, with fluid parameters $\rho = 1[\frac{g}{cm^3}]$, $\eta_0 = 4570[Pa\ s]$, $Cu = 10.5$, $n = 0.04$ that fits the experimentally measured shear–thinning viscosity of this fluid, has been chosen. At the inflow a parabolic velocity profile, with it's maximum value given in the Table 6.6, is imposed. Such flow conditions lead to large variations in viscosity, what is shown in Figure 6.33. The variants for the blocks D_1 and D_2 in the preconditioners, given in Table 6.5, that have been used are following. D_1 is a preconditioner to the block matrix A (or transformed matrix $\bar{A}$, for details see Section 5.3). As it is discussed in Section 5.1 it is enough to use a block diagonal one. Therefore, two such block diagonal preconditioners are considered, namely block ILU, where incomplete LU factorizations are performed for diagonal blocks, denoted by $BILU_A$ and block $RILU$ (relaxed block ILU factorization for various β) denoted by $BRILU_A(\beta)$. D_2 is a preconditioner to the Schur complement, that will be de-

UBGS	$\mathcal{D}_{GS}^{-1} = \begin{pmatrix} D_1 & 0 \\ B & D_2 \end{pmatrix}^{-1}$
UBIT	$\mathcal{D}_{GS}^{-1} = \begin{pmatrix} D_1 & B^T \\ B & 0 \end{pmatrix}^{-1}$
BD	$\bar{\mathcal{D}}_{D}^{-1} = \begin{pmatrix} D_1^{-1} & 0 \\ 0 & D_2^{-1} \end{pmatrix}$
BLT	$\bar{\mathcal{D}}_{LT}^{-1} = \begin{pmatrix} D_1^{-1} & 0 \\ B(I-\bar{A}) & D_2^{-1} \end{pmatrix}$
BGS	$\bar{\mathcal{D}}_{GS}^{-1} = \begin{pmatrix} D_1 & 0 \\ B(I-\bar{A}) & D_2 \end{pmatrix}^{-1}$

Table 6.5: *Preconditioners to the fully coupled method.*

Domain	Contraction - expansion	2:1 planar contraction
Grid	Fine : 88704 fluid cells Coarse: 9856 fluid cells	Fine : 89280 fluid cells Coarse: 9920 fluid cells
Accuracy	$\epsilon = 10^{-6}$	$\epsilon = 10^{-6}$
Time step	Fine : 10^{-5} Coarse: 10^{-2}	Fine : 10^{-5} Coarse: 10^{-2}
Inflow velocity	$U_{inf} = 0.1[\frac{cm}{s}]$	$U_{inf} = 0.05[\frac{cm}{s}]$

Table 6.6: *The set up for the performed tests in order to check different preconditioning techniques applied to the fully coupled method*

noted here by $\Lambda = BD_1^{-1}B^T$ (or alternatively $\Lambda = BD_1^{-1}\bar{B}^T$). Since finding D_1^{-1} is computationally costly, in the numerical simulations it is replaced by $\{diag(D_1)\}^{-1}$. For preconditioners to Λ, we have chosen the incomplete LU factorization ILU_Λ, relaxed incomplete LU factorization $RILU_\Lambda(\beta)$, and finally, the block Λ has also been inverted with use of Jacobi iterations preconditioned by $RILU_\Lambda(\beta)$. During the transformation of the system (3.2), the transformation matrix $H = (diag\{part(A)\})$ (for details see equation (5.13)). In fact, similar idea as in relaxed incomplete LU factorizations is used here. Diagonal matrix H is constructed by subtracting from the diagonal entries of A the weighted sum of its row entries. It can be mathematically written as $H = (diag(A) - \beta Ae)$, where $e = [1, 1, 1..., 1]^T$ and β is the weight. Note, that taking $\beta = 0$ result in $H = (diag(A))$. In all the simulations, we have used such a matrix H with $\beta = 0.65$. Increase or decrease of β did not improve the results.

Below, several tables with the results from simulations are presented. First, we

BiCGstab	TRANSFORMED						UNTRANSFORMED			
	BGS		**BD**		**BLT**		**UBGS**		**UBIT**	
$\tau = 10^{-2}$	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU
$BRILU_A(0.8)$ $RILU_\Lambda(0.65)$	168	5.1	> 999	—	> 999	—	474	9.52	> 999	—
$BRILU_A(0.8)$ $Jaccob_\Lambda(8)$	80	4.27	> 999	—	> 999	—	521	22.93	> 999	—
$BILU_A$ $RILU_\Lambda(0.65)$	273	8.58	> 999	—	> 999	—	515	13.71	> 999	—
$BRILU_A(0.8)$ ILU_Λ	273	8.76	> 999	—	> 999	—	> 999	—	> 999	—
$BILU_A$ $Jaccob_\Lambda(8)$	131	7.54	> 999	—	> 999	—	434	21	> 999	—

Table 6.7: *Number of iterations and CPU times for contraction expansion geometry on coarse grid and coarse time step. BiCGstab iterative method used.*

show five tables (Table 6.7, 6.8, 6.9, 6.10, 6.11) concerning simulations in contraction–expansion geometry (see left plot from Figure 6.33). The iterative procedure was stopped if the accuracy, presented in Table 6.6, has not been achieved within 999 steps. For a coarse grid the best results were achieved when the block Gauss–Seidel preconditioner (**BGS**), applied to the transformed system (3.2), has been used. For a coarse time step $\tau = 10^{-2}$, the best result was obtained with GMRES(100) solver (Table 6.8) with block $BILU_A$ and $Jaccob_\Lambda(8)$ preconditioners. The same number of iterations was achieved by BiCGstab (Table 6.7) with block $BRILU_A(0.8)$ and $Jaccob_\Lambda(8)$ preconditioners, however it needed more computational CPU time to perform calculations. This is coming from more frequent action of the preconditioner in the latter case. Neither the change of the relaxation parameter β nor increase or decrease of the number of Jacobi iterations improved the results. However, the situation changes when fine time step is chosen. In such situation BiCGstab (Table 6.9) performs better, in terms of iteration number and computational CPU time, for all considered block preconditioners D_1 and D_2 comparing with GMRES(100) (Table 6.10). Moreover, **BGS** preconditioner was the only one, from all considered here, that allowed us to reach the imposed accuracy on the fine grid. In Table 6.11 the results are presented. Again, while for coarse time step the best results obtained by BiCGstab and GMRES(100) with block $RILU_\Lambda(0.65)$ and $Jaccob_\Lambda(8)$ were quite similar, BiCGstab has performed much better when the fine time step $\tau = 10^{-5}$ was chosen.

Similar results have been obtained in 2:1 planar contraction geometry (Tables 6.12, 6.13, 6.14, 6.15). For coarse grid, the best performance was achieved by BiCGstab with **BGS** and block $BRILU_A(0.8)$ and $RILU_\Lambda(0.65)$ preconditioners for both coarse and fine time steps. All the rest preconditioners, namely the block diagonal (**BD**), the block lower triangular (**BLT**), applied to the transformed system (3.2), as well as the

GMRES(100)	TRANSFORMED						UNTRANSFORMED			
	BGS		**BD**		**BLT**		**UBGS**		**UBIT**	
$\tau = 10^{-2}$	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU
$BRILU_A(0.8)$ $RILU_\Lambda(0.65)$	280	11.05	570	18.93	570	21.27	> 999	—	704	30.05
$BRILU_A(0.8)$ $Jaccob_\Lambda(8)$	82	3.94	256	11.08	256	12.19	785	36.32	> 290	28.03
$BILU_A$ $RILU_\Lambda(0.65)$	290	12.17	> 999	—	> 999	—	> 999	—	241	10.57
$BRILU_A(0.8)$ ILU_Λ	300	12.37	575	19.83	575	22.17	> 999	—	474	21.26
$BILU_A$ $Jaccob_\Lambda(8)$	80	3.9	191	8.77	191	9.58	549	26.73	151	15.03

Table 6.8: *Number of iterations and CPU times for contraction expansion geometry on coarse grid and coarse time step. GMRES(100) iterative method used.*

ones applied to the untransformed system (3.2) – the block Gauss–Seidel (**UBGS**) and the block indefinite triangular (**UBIT**), needed at least twice more iterations to achieve imposed accuracy compared with already discussed **BGS**.

Moreover, for both geometries, when coarse grid with fine time step was chosen, in most of the cases **BLT** was performing less iterations than **BD**, when BiCGstab was used (Table 6.9, 6.14). Similar observations were done in [22], where **BLT** showed better results than **BD** for all variants of preconditioners applied to subblocks of the transformed system. However, the simulations there were performed with staggered grid arrangement, contrary to collocated one used here, and for different domain. Therefore the influence of the discretization on the iterative solver can not be provided now, however it is planned for a future work. In the case of GMRES(100) iterative solver (not presented in [22]) both **BD** and **BLT** perform the same number of iterations (Table 6.8, 6.10, 6.13, 6.15). The first needs less computational time since less work has to be done when it's applied. Also GMRES(100) performed better than BiCGstab when coarse time step was chosen.

In Table 6.16 the comparison between GMRES(100) and BiCGstab applied on a fine grid is shown. Again, only (**BGS**) preconditioner applied to the transformed system (3.2) allowed to reach imposed accuracy. GMRES(100) performed better when coarse time step was taken, contrary to the simulation with the fine time step, where BiCGstab was a better choice.

Let us now discuss the performance of the preconditioners applied to the untransformed system (3.2). Only results obtained on the coarse grids are presented, since on the fine grid neither **UBGS** nor **UBIT** could achieve imposed accuracy within 999 iterations. When the coarse time step was chosen, the block Gauss–Seidel (**UBGS**) preconditioner performed better with BiCGstab solver, contrary to the indefinite block triangular (**UBIT**) one, which has showed better results with

BiCGstab	TRANSFORMED						UNTRANSFORMED			
	BGS		**BD**		**BLT**		**UBGS**		**UBIT**	
$\tau = 10^{-5}$	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU
$BRILU_A(0.8)$ $RILU_\Lambda(0.65)$	49	1.57	125	2.49	116	3.19	250	5.35	179	6.42
$BRILU_A(0.8)$ $Jaccob_\Lambda(8)$	35	1.93	763	31.3	$>$ 999	—	224	9.71	76	11.28
$BILU_A$ $RILU_\Lambda(0.65)$	78	2.78	172	3.75	179	5.37	274	7.23	74	3.19
$BRILU_A(0.8)$ ILU_Λ	53	1.68	90	1.83	89	2.52	216	4.85	127	5.34
$BILU_A$ $Jaccob_\Lambda(8)$	40	2.45	658	28.8	152	7.87	179	8.76	54	8.49

Table 6.9: *Number of iterations and CPU times for contraction expansion geometry on coarse grid and fine time step. BiCGstab iterative method used.*

GMRES(100) (see Tables 6.7, 6.8, 6.12 and 6.13). For a fine time step used the best result was obtained by BiCGstab iterative method with **UBIT** and block $BILU_A$ and $RILU_\Lambda(0.65)$ preconditioners. However, still **UBIT** needed at least twice more CPU time than **BGS** with $BRILU_A(0.8)$ and $RILU_\Lambda(0.65)$ preconditioners.

Since in many applications the number of gid nodes is of order $\mathcal{O}(10^6)$, the use of GMRES with long sequences is not suitable due to memory limitations. Therefore, let us finally discuss the results obtained by GMRES(10) and presented in Table 6.17. Calculations were done with use of block Gauss–Seidel preconditioner (**BGS**), applied to the transformed system (3.2), on a coarse grid with a fine time step $\tau = 10^{-5}$. It can be seen, that decreasing 10 times the number of sequences results in, at least, 5 times more iteration number needed to perform in order to reach the imposed accuracy. Moreover, one needs at least 4 times more the CPU time to reach that accuracy. Note, that GMRES method is preferred in [4, 42], however there more academic examples are considered and such long sequences can be used. Also results presented here show, that the GMRES with long sequences is a good choice when relatively small (in terms of the number of grid points) geometries are considered.

6.5 Summary.

The simulation results of two classes of: generalized Newtonian and non–Newtonian fluids, as well as performance of different solution techniques applied to the first class, have been presented in this Chapter.

First, very good qualitative comparison of shear–thinning fluid, modeled by Carreau viscosity, is shown. Next, the results of anisotropic viscosity model are given. It is shown, that extensional–thickening and shear–thinning fluids can predict growth

GMRES(100)	TRANSFORMED						UNTRANSFORMED			
	BGS		**BD**		**BLT**		**UBGS**		**UBIT**	
$\tau = 10^{-5}$	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU
$BRILU_A(0.8)$ $RILU_\Lambda(0.65)$	83	3.05	210	6.98	210	7.83	893	31.78	155	6.07
$BRILU_A(0.8)$ $Jaccob_\Lambda(8)$	55	2.28	143	5.86	143	6.52	288	13.28	103	10.31
$BILU_A$ $RILU_\Lambda(0.65)$	128	5.08	388	13.56	388	13.91	864	32.63	155	6.67
$BRILU_A(0.8)$ ILU_Λ	86	3.24	193	6.57	193	7.46	792	28.88	107	4.86
$BILU_A$ $Jaccob_\Lambda(8)$	59	2.63	145	6.28	145	6.85	238	11.27	73	7.23

Table 6.10: *Number of iterations and CPU times for contraction expansion geometry on coarse grid and fine time step. GMRES(100) iterative method used.*

of the vortices in various contraction domains, what is also observed in experiments.

Then, simulation results of dilute polymer solutions modeled by integral Oldroyd B model are presented. Both quantitative and qualitative agreement with the results obtained by differential counterpart and experimental observations, respectively, are obtained.

Concentrated polymer solutions are modeled by integral Doi Edwards constitutive equation with possible chain stretch. Very good quantitative agreement with experimentally measured data of first normal stress difference and shear viscosity for polystyrene solution is achieved. Also qualitative agreement with the results obtained by differential approximation to Doi Edwards model in constriction geometry is shown.

Both time integral constitutive equations allow to perform stable simulations for much higher Weissenberg numbers compared with differential ones.

Growth of the vortices is always associated with much higher pressure drop than for Newtonian fluid with the same viscosity.

Finally, the performance of projection type methods and fully coupled method in simulations of generalized Newtonian fluids has been checked. It is shown, that explicit discretization of the cross derivatives from momentum equations may lead to stability problems. To avoid it, implicit discretization is recommended.

The performance of different preconditioners (applied to the fully coupled method) and iterative methods was studied for two geometries (2:1 planar contraction and contraction expansion domains), on coarse and fine grid, using different time steps. The best results are obtained with block Gauss–Seidel preconditioner applied to transformed system of discretized and linearized continuity and momentum equations (5.15). Only using this preconditioner allowed us to reach imposed accuracy on a fine grid within 999 steps.

	BGS preconditioner			
	BiCGstab		GMRES(100)	
	$\tau = 10^{-2}$	$\tau = 10^{-5}$	$\tau = 10^{-2}$	$\tau = 10^{-5}$
$BRILU_A(0.8)$ $RILU_\Lambda(0.65)$	$>$ 999 iter. —	249 iter. 85 sec.	$>$ 999 iter. —	560 iter. 227.52 sec.
$BRILU_A(0.8)$ $Jaccob_\Lambda(8)$	387 iter. 229.32 sec.	219 iter. 140.06 sec.	392 iter. 214.38 sec.	285 iter. 154.45 sec.
$BILU_A$ $RILU_\Lambda(0.65)$	$>$ 999 iter. —	$>$ 999 iter. —	$>$ 999 iter. —	994 iter. 438.79 sec.
$BRILU_A(0.8)$ ILU_Λ	$>$ 999 iter. —	$>$ 999 iter —	$>$ 999 iter. —	$>$ 999 iter. —
$BILU_A$ $Jaccob_\Lambda(8)$	842 iter. 551.29 sec.	765 iter. 265.83 sec.	686 iter. 396.69 sec.	$>$ 999 iter. —

Table 6.11: *Number of iterations and CPU times for contraction expansion geometry on fine grid and fine time step. BiCGstab against GMRES(100) iterative method used.*

BiCGstab performed better than GMRES(100) on a fine grid, when fine time step was chosen, while, for a coarse time step the performance was similar.

Block lower triangular and block diagonal preconditioners, applied to the transformed system (5.15), performed the same number of iterations when GMRES(100) was used.

On coarse grid and coarse time step chosen, the block indefinite triangular preconditioner, applied to the untransformed system (3.2), preformed better with GMRES(100) than BiCGstab. It was contrary to the block Gauss–Seidel (also applied to (3.2)), which showed better behavior with BiCGstab solver.

Choosing GMRES(10) solver with short number of sequences, which is necessary in industrial applications (domains of order 10^6 number of grid points) due to memory requirements, needed high number of iteration steps to converge.

BiCGstab	TRANSFORMED						UNTRANSFORMED			
	BGS		**BD**		**BLT**		**UBGS**		**UBIT**	
$\tau = 10^{-2}$	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU
$BRILU_A(0.8)$ $RILU_\Lambda(0.65)$	119	3.58	> 999	—	> 999	—	381	7.71	> 999	—
$BRILU_A(0.8)$ $Jaccob_\Lambda(8)$	62	3.13	> 999	—	> 999	—	470	19.51	> 999	—
$BILU_A$ $RILU_\Lambda(0.65)$	282	9.58	> 999	—	> 999	—	395	9.85	191	7.8
$BRILU_A(0.8)$ ILU_Λ	148	4.48	> 999	—	> 999	—	317	6.85	> 999	—
$BILU_A$ $Jaccob_\Lambda(8)$	115	6.32	> 999	—	> 999	—	288	13.12	> 999	—

Table 6.12: *Number of iterations and CPU times for 2:1 planar contraction geometry on coarse grid and coarse time step. BiCGstab iterative method used.*

GMRES(100)	TRANSFORMED						UNTRANSFORMED			
	BGS		**BD**		**BLT**		**UBGS**		**UBIT**	
$\tau = 10^{-2}$	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU
$BRILU_A(0.8)$ $RILU_\Lambda(0.65)$	198	7.78	889	29.88	899	33.62	> 999	—	983	37.62
$BRILU_A(0.8)$ $Jaccob_\Lambda(8)$	75	3.36	200	8.99	200	9.74	756	33.99	379	35.69
$BILU_A$ $RILU_\Lambda(0.65)$	298	12.6	> 999	—	> 999	—	> 999	—	195	8.8
$BRILU_A(0.8)$ ILU_Λ	270	10.38	799	27.49	799	30.34	> 999	—	498	22.28
$BILU_A$ $Jaccob_\Lambda(8)$	74	3.45	179	7.81	179	8.44	399	19.23	95	9.2

Table 6.13: *Number of iterations and CPU times for 2:1 planar contraction geometry on coarse grid and coarse time step. GMRES(100) iterative method used.*

BiCGstab	TRANSFORMED						UNTRANSFORMED			
	BGS		**BD**		**BLT**		**UBGS**		**UBIT**	
$\tau = 10^{-5}$	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU
$BRILU_A(0.8)$ $RILU_\Lambda(0.65)$	44	1.31	100	1.86	101	2.63	214	4.3	177	6.08
$BRILU_A(0.8)$ $Jaccob_\Lambda(8)$	33	1.66	154	5.96	146	6.8	222	9.27	85	11.71
$BILU_A$ $RILU_\Lambda(0.65)$	75	2.55	155	3.14	158	4.43	306	7.62	76	3.16
$BRILU_A(0.8)$ ILU_Λ	47	1.44	76	1.49	76	2.02	211	4.56	109	4.33
$BILU_A$ $Jaccob_\Lambda(8)$	36	1.99	170	7.05	139	6.74	206	9.3	56	8.17

Table 6.14: *Number of iterations and CPU times for 2:1 planar contraction geometry on coarse grid and fine time step. BiCGstab iterative method used.*

GMRES(100)	TRANSFORMED						UNTRANSFORMED			
	BGS		**BD**		**BLT**		**UBGS**		**UBIT**	
$\tau = 10^{-5}$	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU
$BRILU_A(0.8)$ $RILU_\Lambda(0.65)$	79	2.72	193	6.42	193	7.28	$>$ 999	—	160	6.19
$BRILU_A(0.8)$ $Jaccob_\Lambda(8)$	49	1.87	142	5.71	142	6.3	293	13.22	148	13.48
$BILU_A$ $RILU_\Lambda(0.65)$	119	4.76	298	10.47	298	11.52	$>$ 999	—	100	4.8
$BRILU_A(0.8)$ ILU_Λ	78	2.71	129	4.04	129	4.53	$>$ 999	—	97	4.4
$BILU_A$ $Jaccob_\Lambda(8)$	59	2.4	100	4.62	100	4.97	282	13.19	80	7.67

Table 6.15: *Number of iterations and CPU times for 2:1 planar contraction geometry on coarse grid and fine time step. GMRES(100) iterative method used.*

	BGS preconditioner			
	BiCGstab		GMRES(100)	
	$\tau = 10^{-2}$	$\tau = 10^{-5}$	$\tau = 10^{-2}$	$\tau = 10^{-5}$
$BRILU_A(0.8)$ $RILU_\Lambda(0.65)$	$>$ 999 iter. —	208 iter. 63.9 sec.	$>$ 999 iter. —	$>$ 999 iter. —
$BRILU_A(0.8)$ $Jaccob_\Lambda(8)$	495 iter. 278.37 sec.	150 iter. 84.57 sec.	260 iter. 132.85 sec.	283 iter. 146.45 sec.
$BILU_A$ $RILU_\Lambda(0.65)$	$>$ 999 iter. —	$>$ 999 iter. —	$>$ 999 iter. —	$>$ 999 iter. —
$BRILU_A(0.8)$ ILU_Λ	$>$ 999 iter. —	182 iter 59.14 sec.	$>$ 999 iter. —	$>$ 999 iter. —
$BILU_A$ $Jaccob_\Lambda(8)$	$>$ 999 iter. —	489 iter. 301.61 sec.	390 iter. 217.83 sec.	330 iter. 180.21 sec.

Table 6.16: *Number of iterations and CPU times for 2:1 planar contraction geometry on fine grid and fine time step. BiCGstab against GMRES(100) iterative method used.*

BGS preconditioner	GMRES(10)			
	Contraction - expansion		2:1 planar contraction	
	Iter.	CPU	Iter.	CPU
$BRILU_A(0.8)$ $RILU_\Lambda(0.65)$	541	10.4	721	13.12
$BRILU_A(0.8)$ $Jaccob_\Lambda(8)$	410	12.85	269	8.06
$BILU_A$ $RILU_\Lambda(0.65)$	816	17.64	$>$ 999	—
$BRILU_A(0.8)$ ILU_Λ	660	12.95	978	18.13
$BILU_A$ $Jaccob_\Lambda(8)$	310	10.41	320	10.07

Table 6.17: *Number of iterations and CPU times for 2:1 planar contraction and contraction expansion geometries on coarse grid and fine time step. GMRES(10) iterative method used.*

Chapter 7

Concluding remarks.

This thesis deals with modeling aspects of generalized Newtonian and of non-Newtonian fluids, as well as with development and validation of algorithms used in simulation of such fluids.

The main contribution in the modeling part is the introduction and analysis of a new model for the generalized Newtonian fluids, where constitutive equation is of an algebraic form. Distinction between shear and extensional viscosities leads to anisotropic viscosity model. It can be considered as a natural extension of the well known Carreau model (isotropic viscosity), that deals only with shear viscosity properties of the fluid, by taking additionally into account extensional viscosity properties. Numerical results, presented in section 6.2, show that the anisotropic viscosity model gives much better agreement with experimental observations compared to the isotropic one. Choosing extensional–thickening and shear–thinning fluid result, according to the experiments, in growth of the vortices in contraction flows, contrary to the shear–thinning fluid only (described by Carreau model). Moreover, as in experiments, such effect is associated with the extensional stress, e.g. only fluids that exhibit extensional stress growth with higher extensional–rates has shown it.

The second contribution of the thesis consist of development and analysis of robust and reliable algorithm for simulation of generalized Newtonian fluids. For such fluids the momentum equations are strongly coupled through mixed derivatives appearing in the viscous term (unlike the case of Newtonian fluids). It is shown in this thesis, that a careful treatment of those derivatives is essential in deriving robust algorithms. Modification of a standard SIMPLE-like algorithm is given, where all the viscous terms from the momentum equations are discretized in an implicit manner. Moreover, it is shown that a block diagonal preconditioner to the viscous operator is good enough to be used in simulations. Furthermore, different solution techniques, namely projection type methods (consist of solving momentum equations and pressure correction equation) and fully coupled methods (momentum and continuity equations are solved together), are compared. It is shown, that explicit discretization of the mixed derivatives lead to stability problems. Finally, analytical estimates for three different preconditioners, applied to the transformed system arising after discretization and linearization of the momentum and continuity equations, are provided. A new preconditioner, block Gauss-Seidel applied to the transformed system, is proposed. From the analysis one can see, that this preconditioner is able to cluster eigenvalues around the unity number independent of the transformation step, which is not the case for other preconditioners applied to the transformed system and discussed here. It has also shown the best behavior (among all preconditioners discussed in the thesis) in numerical experiments.

The third contribution consist of comparison and validation of numerical algorithms applied in simulations of non-Newtonian fluids modeled by time integral constitutive equations. Numerical results from simulations of dilute polymer solutions, described by the integral Oldroyd B model, have shown very good quantitative agreement with the results obtained by differential Oldroyd B counterpart in 4:1 planar contraction domain at low Weissenberg numbers. In this case, the Weissenberg number is changed by changing the relaxation time. However, contrary to the differential Oldroyd B model, the integral one allows to perform stable simulations also in the range of high Weissenberg numbers. Moreover, we have obtained very good agreement with experimental observations of Boger B64 fluid (section 6.3.1), where, according to experiments, growth of the vortices with increasing Weissenberg number, by increasing inflow velocity, is observed. Next, comparison of two methods used for approximation of the time integral constitutive equation, namely Deformation Field Method (DFM) and Backward Lagrangian Particle Method (BLPM), is performed. In BLPM the particle paths are recalculated at every time step of simulations, what has never been tried before. The results have shown, that in considered geometries both methods give similar results. Finally, simulations of concentrated polymer solutions (polystyrene and polybutadiene solutions), described by the integral Doi Edwards model, supplemented by chain length fluctuations, have shown very good qualitative agreement with the results obtained by its differential approximation in 4:1:4 constriction domain. Again, much higher Weissenberg numbers can be achieved when integral model is used. Moreover, very good quantitative results with experimental data of polystyrene solution for the first normal stress difference ($N_1 = T_{xx} - T_{yy}$, where T_{xx} and T_{yy} are diagonal components of the stress tensor) and shear viscosity defined as the quotient of a shear stress and a shear rate, e.g. $\eta(\dot{\gamma}_{xy}) = \frac{T_{xy}}{\dot{\gamma}_{xy}}$, are obtained.

Summarizing, in this thesis two classes of fluids described by: an algebraic and a integral constitutive equations are considered. We have introduced new anisotropic viscosity model, describing the generalized Newtonian fluids, which gives ability to predict growth of the vortices even if shear–thinning fluid is considered. Moreover, it is shown, that integral constitutive equations, describing non–Newtonian (viscoelastic) fluids, allow to perform stable simulations for much higher Weissenberg numbers compared with their differential counterpart (Oldroyd B) or approximations (Doi Edwards). Finally, systematic analysis of solution techniques for generalized Newtonian fluids has been performed.

List of symbols

$\mathbf{v} = (v_1, v_2, v_3)$ - velocity,
$\mathbf{x} = (x_1, x_2, x_3)$ - space position,
p - pressure,
$\mathbf{T}$ - polymeric stress,
$\bar{\mathbf{T}}$ - approximation to $\mathbf{T}$,
$\boldsymbol{\sigma}$ - viscous stress,
$\boldsymbol{\tau}$ - total stress,
$\boldsymbol{\gamma}$ - rate of deformation tensor,
ρ - fluid density,
η_0 - zero shear-rate viscosity (dynamic viscosity),
ν - kinematic viscosity
$\mu(t, t')$ - memory function,
$\mathbf{Q}(t, t')$ - Doi Edwards partial orientation tensor,
$\hat{\mathbf{Q}}(t, t')$ - approximation to $\mathbf{Q}$,
G_e - elastic constant,
τ_{relax} - general relaxation time,
τ_d - relaxation time (Doi Edwards model),
τ_R - the Rouse time,
$\mathbf{E}(t, t')$ - deformation tensor,
$\mathbf{u}$ - unit vector tangent to the primitive chain,
L - average chain length,
L_0 - average chain length in equilibrium,
$\lambda = \frac{L}{L_0}$ - the average chain stretch ratio,
η_p - polymeric constant,
τ_{Oldr} - relaxation time (Oldroyd B model),
$\mathbf{B}(t, t')$ - Finger strain tensor,
δ - identity matrix,
$\dot{\gamma}$ - shear rate,
$\dot{\epsilon}$ - elongational rate,
$\eta(\dot{\gamma})$ - Carreau shear viscosity,
$\eta_e(\dot{\epsilon})$ - extensional viscosity,
D_v - discrete diffusion operator,
C_v - discrete convection operator,
B^T - discrete gradient operator,
B - discrete divergence operator,

$\tau = t^{n+1} - t^n$ - time step,
Cu, n - parameters to fit shear properties of the fluid,
Cu_e, n_e - parameters to fit elongational properties of the fluid,
T - temperature,
a_1, a_2 - parameters to fit zero shear-rate viscosity,
α - parameters to fit pressure dependence on viscosity,
$\delta x, \delta y$ - space discretization sizes,
δV - volume of CV,
A_w, A_e, A_n, A_s - areas of west, east, north and south faces of CV,
ξ - eigenvalue,
$[x, y]$ - eigenvector,
$\mathcal{D}_{GS}$ - block Gauss–Seidel preconditioner to untransformed system,
$\mathcal{D}_{IT}$ - block indefinite triangular preconditioner to untransformed system,
$\bar{\mathcal{D}}_D$ - block diagonal preconditioner to transformed system,
$\bar{\mathcal{D}}_{LT}$ - block lower triangular preconditioner to transformed system,
$\bar{\mathcal{D}}_{GS}$ - block Gauss–Seidel preconditioner to transformed system,
$Re = \frac{\rho \bar{U} L}{\eta_0}$ - Reynolds number,
$We = \frac{\tau_{relax} \bar{U}}{L}$ - Weissenberg number,
L - width of upstream channel in 2D planar contraction geometry,
$\bar{U}$ - average upstream channel velocity.

List of Figures

2.1 *Shear-thinning fluid. Viscosity versus shear-rate.* 20

2.2 *Shear-thickening fluid. Viscosity versus shear-rate.* 21

2.3 *Schematic picture of highly entangled polymer solutions.* 25

2.4 *Schematic picture of the tube model.* 26

2.5 *Schematic picture of dilute polymer solutions.* 28

2.6 *The dumbbell model.* 28

3.1 *A representative CV.* 37

4.1 *A representative backward particle tracking.* 49

4.2 *Figure presents which deformation fields are annihilated at appropriate time iteration. Black squares denote deformations used in approximation of integral constitutive equation, red one denote shifted deformation fields after time step and green circles denote annihilated deformations, which later are used to store the youngest one.* 53

5.1 *Eigenvalue distribution for block matrix A preconditioned with a block diagonal preconditioner.* 61

5.2 *Eigenvalue distribution for optimal choice of block diagonal preconditioner* $\bar{\mathcal{D}}_D$ *to transformed system* $\bar{\mathcal{L}}$. 66

5.3 *Eigenvalue distribution for optimal choice of block lower triangular preconditioner* $\bar{\mathcal{D}}_{LT}$ *to transformed system* $\bar{\mathcal{L}}$. 67

5.4 *Eigenvalue distribution for optimal choice of block Gauss–Seidel preconditioner* $\bar{\mathcal{D}}_{GS}$ *to transformed system* $\bar{\mathcal{L}}$. 69

6.1 *2D planar contraction domain.* 72

6.2 *The effect of varying Cu number on the calculated centerline velocity for n=0.2, Carreau model.* 72

6.3 *Carreau viscosity plots for* $n = 0.2$, $Cu = 0.01$ *(left figure),* $Cu = 100$ *(right figure).* 72

6.4 *The effect of varying n number on the calculated centerline velocity for* $Cu = 100$, *Carreau model.* 73

6.5 *The effect of decreasing n with* $Cu = 100$ *(left figure) and increasing* Cu *with* $n = 0.2$ *(right figure) on axial velocity profile at the contraction* $x = x_c$. 73

6.6 *Streamline plots for Newtonian fluid (upper left) and shear–thinning fluids of the same zero–shear rate viscosity for* $n = 0.2$ *and* $Cu = 0.01$ *(upper right),* $Cu = 1$ *(lower left),*$Cu = 100$ *(lower right).* 74

6.7 *Streamline plots for Newtonian fluid (upper left) and shear–thinning extensional–thickening fluids of the same zero–shear rate viscosity* $\eta_0 = 10[Pas]$ *for* $Cu = 1, n = 0.8, Cu_e = 1, n_e = 2$*. Inflow velocity is increased for upper right, lower left and lower right figure respectively.* 75

6.8 *3D planar contraction (left figure) and 3D square–to–square contraction (right figure) geometries.* 76

6.9 *3D planar contraction geometry. Streamline plots, Newtonian fluid (left figure) and extensional–thickening fluid (right figure).* 76

6.10 *3D planar contraction geometry. Extensional viscosity (left figure) and shear viscosity (right figure).* 77

6.11 *3D square–to–square contraction geometry. Streamline plots, Newtonian fluid (left figure) and extensional–thickening fluid (right figure).* 77

6.12 *3D square–to–square contraction geometry. Extensional viscosity (left figure) and shear viscosity (right figure).* 78

6.13 *Reattachement lengths against* We *numbers (left figure). Reattachement lengths against* We *numbers for different meshes (right figure).* 78

6.14 *Reattachement lengths against high* We *numbers.* 79

6.15 *We=0 (left figure), We=0.5 (center figure), We=1 (right figure).* . 79

6.16 *We=1.5 (left figure), We=2 (center figure), We=2.5 (right figure).* . . 80

6.17 DFM simulations.$We = 0$ (upper left figure), $We = 0.05$, $Re = 0.0125$ (upper right figure), $We = 0.1$, $Re = 0.025$ (lower left figure), $We = 0.2$, $Re = 0.05$ (lower right figure). 81

6.18 *Pressure drops of the Newtonian (green color) and Boger (black color) fluids against* We *numbers. The Newtonian pressure drops were sampled at the appropriate mean outflow velocities* $\bar{U} = \frac{We \cdot L}{\tau_{Oldr}}$ 81

6.19 *Deformation Field Method (left figure), Backward Lagrangian Particle Method (right figure) at* $We = 1$. 82

6.20 *BLPM simulations of Boger B64 fluid.* $We = 0.05$, $Re = 0.0125$ *(left figure),* $We = 0.1$, $Re = 0.025$ *(center figure),* $We = 0.2$, $Re = 0.05$ *(right figure).* 82

6.21 *Streamlines contours in 3D planar 4:1 contraction domains. Newtonian flow (left figure) and Boger B64 flow (right figure).* 83

6.22 *Streamlines contours in 3D square–to–square 4:1 contraction domains. Newtonian flow (left figure) and Boger B64 flow (right figure).* . . . 84

6.23 *Rheological results for polystyrene solution. Left figure presents comparison of* $N_1 = T_{xx} - T_{yy}$ *between the experimental data (black triangles) given in [6] and calculated results (blue circle - original Doi Edwards model, green star - Doi Edwards with chain stretch) against shear–rate* $\dot{\gamma}$. *Central figure presents comparison of the shear viscosity, defined as* $\eta(\dot{\gamma_{xy}}) = \frac{T_{xy}}{\dot{\gamma_{xy}}}$, *against shear–rate* $\dot{\gamma_{xy}}$ *between experimental measurements and calculated results. Right figure shows calculated data of* N_1/G_e *and* T_{xy}/G_e *against* $\tau_d\dot{\gamma}$. 85
6.24 *Streamlines plots for polystyrene solution at* $We = 0$ *(left figure),* $We = 100$ *(center figure) and* $We = 200$ *(right figure).* 85
6.25 *Polystyrene solution, the chain stretch at* $We = 50$ *(left figure) with* $max(\lambda) = 5.096$ *and at* $We = 200$ *(right figure) with* $max(\lambda) = 12.096$ 86
6.26 *2D planar contraction–expansion domain.* 86
6.27 *Streamlines plots for polystyrene solution in 4:1:4 constriction at* $We = 0$ *(left figure),* $We = 250$ *(center figure) and* $We = 500$ *(right figure).* 86
6.28 *Polystyrene solution, the chain stretch at* $We = 250$ *(left figure) with* $max(\lambda) = 8.494$ *and at* $We = 500$ *(right figure) with* $max(\lambda) = 11.436$ 87
6.29 *Streamline plots for polybutadiene solution at* $We = 0$ *(upper left),* $We = 1$ *(upper right),* $We = 5$ *(lower left) and* $We = 10$ *(lower right).* 88
6.30 *Polybutadiene solution, the chain stretch at* $We = 1$ *(left figure) with* $max(\lambda) = 1.487$ *and at* $We = 10$ *(right figure) with* $max(\lambda) = 2.363$ 88
6.31 *Streamlines plots for polybutadiene solution in 4:1:4 constriction at* $We = 0$ *(upper left),* $We = 10$ *(upper right),* $We = 25$ *(lower left) and* $We = 100$ *(lower right).* 89
6.32 *Chain stretch* λ *for polystyrene solution in 4:1:4 constriction at* $We = 10$ *with* $max(\lambda) = 1.76$ *(left figure),* $We = 25$ *with* $max(\lambda) = 2.226$ *(center figure) and* $We = 100$ *with* $max(\lambda) = 2.42$ *(right figure).* . . 90
6.33 *The variable viscosities in Venturi domain (left plot) and 2:1 planar contraction domain (right plot).* 93

List of Tables

6.1 *Time convergence for integral Oldroyd B model. Reattachement length L_v against time step τ at $We = 1.5$.* 80

6.2 *Stability performance of three different solution techniques for shear–thinning fluid, modeled by the Carreau constitutive equation (2.22), with the fluid parameters $Cu_s = 1.0$, $n_s = 0.4$.* 91

6.3 *Stability performance of three different solution techniques for shear–thinning and extensional–thickening fluid, modeled by the extension of the Carreau constitutive equation (2.26), with the fluid parameters $Cu_s = 1.0$, $n_s = 0.4$, $Cu_e = 1.0$, $n_e = 1.1$.* 92

6.4 *Stability performance of three different solution techniques for shear–thickening and extensional–thinning fluid, modeled by the extension of the Carreau constitutive equation (2.26), with the fluid parameters $Cu_s = 1.0$, $n_s = 1.1$, $Cu_e = 1.0$, $n_e = 0.4$.* 92

6.5 *Preconditioners to the fully coupled method.* 94

6.6 *The set up for the performed tests in order to check different preconditioning techniques applied to the fully coupled method* 94

6.7 *Number of iterations and CPU times for contraction expansion geometry on coarse grid and coarse time step. BiCGstab iterative method used.* 95

6.8 *Number of iterations and CPU times for contraction expansion geometry on coarse grid and coarse time step. GMRES(100) iterative method used.* 96

6.9 *Number of iterations and CPU times for contraction expansion geometry on coarse grid and fine time step. BiCGstab iterative method used.* 97

6.10 *Number of iterations and CPU times for contraction expansion geometry on coarse grid and fine time step. GMRES(100) iterative method used.* 98

6.11 *Number of iterations and CPU times for contraction expansion geometry on fine grid and fine time step. BiCGstab against GMRES(100) iterative method used.* 99

6.12 *Number of iterations and CPU times for 2:1 planar contraction geometry on coarse grid and coarse time step. BiCGstab iterative method used.* 100

6.13 *Number of iterations and CPU times for 2:1 planar contraction geometry on coarse grid and coarse time step. GMRES(100) iterative method used.* 100
6.14 *Number of iterations and CPU times for 2:1 planar contraction geometry on coarse grid and fine time step. BiCGstab iterative method used.* 101
6.15 *Number of iterations and CPU times for 2:1 planar contraction geometry on coarse grid and fine time step. GMRES(100) iterative method used.* 101
6.16 *Number of iterations and CPU times for 2:1 planar contraction geometry on fine grid and fine time step. BiCGstab against GMRES(100) iterative method used.* 102
6.17 *Number of iterations and CPU times for 2:1 planar contraction and contraction expansion geometries on coarse grid and fine time step. GMRES(10) iterative method used.* 102

Bibliography

[1] M.Aboubacar, H.Matallah, M.F.Webster, *Highly elastic solutions for Oldroyd-B and Phan-Thien/Tanner fluids with a finite volume/element method: planar contraction flows*, *J.Non-Newtonian Fluid Mech.* **103**, pp 65-103 (2002).

[2] M.Aboubacar, T.N.Phillips, H.R.Tamaddon-Jahromi, M.F.Webster, A.J.Williams, *Numerical simulation of contraction flows for Boger fluids using finite volume methods.*

[3] M.A.Alves, P.J,Oliviera, F.T.Pinho, *Benchmark solutions for the flow of Oldroyd-B and PTT fluids in planar contractions*, *J.Non-Newtonian Fluid Mech.* **110**, pp 45-75 (2003).

[4] O.Axelsson, M.Neytcheva, *Preconditioning methods for linear systems arising in constrained optimization problems*, *Numerical Linear Algebra with Applications***10**, pp 1-26 (2003).

[5] R.Barrett, M.Berry, T.F.Chan, J.Demmel, J.Donato, J.Dongarra, V.Eijkhout, R.Pozo, C.Romine, and H.Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, Philadelphia, PA, 1994.

[6] P.K. Bhattacharjee, D.A. Nguyen, G.H.McKinley, T.Sridhar, *Extensional stress growth and stress relaxation in entangled polymer solutions*, *J. Rheol.*, **47** , 269-290 (2003).

[7] D.V.Boger, *Viscoelastic flows through contractions*, *Ann.Rev.Fluid Mech.* **19**, pp 157-182 (1987).

[8] D.V.Boger, D.U.Hur, R.J.Binnington, *Further observations of elastic effects in tubular entry flows*, *J.Non-Newtonian Fluid Mech.* **20**, pp 31-49 (1986).

[9] M.Doi, S.F.Edwards, The Theory of Polymer Dynamics, *Oxford University Press, New York*, (1986).

[10] H.Elman, D.J.Silvester, A.J.Wathen *Performance and analysis of saddle point preconditioners for the discrete steady-state Navier-Stokes equations*, *Numer. Math.*, **90**, 665-688 (2002).

[11] R.E.Evans, K.Walters, *Flow characteristics associated with abrupt changes in flow geometry in the case of highly elastic liquids*, *J.Non-Newtonian Fluid Mech.* **20**, pp 11-29 (1986).

[12] K.Feigl, F.X.Tanner, B.J.Edwards, J.R.Collier, *A numerical study of the measurements of elongational viscosity of polymeric fluids in a semihyperbilically converging die*, *J. Non-Newtonian Fluid Mech.*, **115** (2-3), 191-215 (2003).

[13] J.H.Ferziger, M.Peric, *Computational methods for fluid dynamics* (Springer, 1999).

[14] L.J.Fetters, D.J.Lohse, D.Richter, T.A.Witten, A.Zirkel, *Connection between polymer molecular weight, density, chain dimensions, and melt viscoelastic properties*, *Macromolecules*, **27**, No.17, pp 4639-4647 (1994).

[15] C.A.J.Fletcher, *Computational Techniques for Fluid Dynamics*, (Springer-Verlag, 1991).

[16] L. Fuchs, H.S. Zhao, *Solution of three-dimensional viscous compressible flows by a multigrid methods*, *Int. Journal for Numerical Methods in Fluids* **4**, pp 539 - 555 (1984).

[17] M.Gerritsma, *Time dependent Numerical Simulations of a Viscoelastic Fluid on a Staggered Grid*, (1996).

[18] E.Guyon, J-P.Hulin, L.Petit, C.D.Mitescu, Physical hydrodynamics, *Oxford University Press, New York*, (2001).

[19] G.Ianniruberto, G.Marrucci, *A multi–mode CCR model for entangled polymers with chain stretch*, *J.Non-Newtonian Fluid Mech.*, **102**, pp. 383-395 (2002).

[20] G.Ianniruberto, G.Marrucci, *A simple constitutive equation for entangled polymers with chain stretch*, *J.Rheol.*, **45**(6), pp. 1305-1318 (2001).

[21] T.C.BMcLeish, R.G.Larson, *Molecular constitutive equations for a class of branched polymers: The pom-pom polymer*, *J.Rheol.*, **42**(1), pp. 81-110 (1998).

[22] O. Iliev, J. Linn, M. Moog, D. Niedziela, V. Starikovicius *On the performance of certain iterative solvers for coupled systems arising in discretization of non-Newtonian flow equations*, *ECCOMAS*, (2004).

[23] O.Iliev, D.Stoyanov, *Multigrid - adaptive local refinement solver for incompressible flows*, *Berichte des Fraunhofer ITWM*, **54** (2003)

[24] R.Keunings *Finite element methods for integral viscoelastic fluids* , *British Society of Rheology*, 167-195 (2003)

[25] M.E. Kime-E, R.A. Brown, R.C. Armstrong, *The roles of inertia and shear-thinning in flow of an inelastic liquid through an axisymmetric sudden contraction*, *J. Non-Newtonian Fluid Mech.*, **13** , 341-363 (1983).

[26] L. Little, Y. Saad, *Block LU Preconditioners for Symmetric and Nonsymmetric Saddle Point Problems*, *SIAM J. Sci. Comput.*, **25** No. 2, 729-748 (2003).

[27] G.Marrucci, F.Greco, G.Ianniruberto, *Integral and differential constitutive equations for entangled polymers with simple versions of CCR and force balance on entanglements*, *Rheol.Acta* , **40**, pp. 98-103 (2001).

[28] H.Matallah, P.Townsend, M.F.Webster, *Recovery and stress-spliting schemes for viscoelastic flows*, *J.Non-Newtonian Fluid Mech.*, **75**, pp. 139-166 (1998).

[29] D.W.Mead, R.G.Larson, M.Doi, *A Molecular Theory for Fast Flows of Entangled Polymers*, *Macromolecules*, **31** , 7895-7914 (1998).

[30] H.Nguyen, D.V.Boger, *The kinematics and stability of die entry flows*, *J.Non-Newtonian Fluid Mech.* **5**, pp 353-368 (1979).

[31] S.Nigen, K.Walters, *Viscoelastic contraction flows: comparison of axisymmetric and planar configurations*, *J.Non-Newtonian Fluid Mech.* **102**, pp 343-359 (2002).

[32] R.G.Owens, T.N.Phillips, *Computational Rheology*, *Imperial College Press* (July 1, 2002).

[33] S.V.Patankar, *Numerical Heat Transfer and Fluid Flow*, (Hemisphere, 1980).

[34] D.S.Pearson, A.D.Kiss, L.J.Fetters, M. Doi, *Flow–induced Birefringence of Concentrated Poyisoprene Solutions*, *J. Rheol.*, **33**(3) , 517-535 (1989).

[35] M.Perić, R.Kessler and G.Scheuerer,*Comparison on finite-volume numerical methods with staggered and collocated grids*, *Comput. Fluids* **16** pp. 389-403 (1988).

[36] E.A.J.F.Peters, Generalization of the deformation field method to simulate advanced reptation models in complex flows, *J.Rheol.*, **44**, pp 811-829 (2000).

[37] E.A.J.F.Peters, M.A. Hulsen, B.H.A.A van den Brule, *Instationary Eulerian viscoelastic flow simulation using time separable Rilivin-Sawyers constitutive equations*, *J.Non-Newtonian Fluid Mech.* **89** (2000) 209-228.

[38] T.N.Phillips, A.J.Williams, *Viscoelastic flow through a planar contraction using a semi-Lagrangian finite volume method*, *J.Non-Newtonian Fluid Mech.* **87**, pp 215-246 (1999).

[39] Yousef Saad, *Iterative Methods for Sparse Linear Systems*, 2000.

[40] A.A.Samarskii, *Theory of difference schemes*, (Moskow, 1977, in Russian).

[41] A.Samarskii, P.Vabishchevich *Additive Schemes for Systems of Time–dependent Equations of Mathematical Physics*, *Lecture Notes In Computer Science* **2542**, pp 46-60 (2002).

[42] D. Silvester, H. Elman, *Efficient preconditioning of the linearized Navier-Stokes equations*, *J. of Computational and Applied Mathematics***128**, pp 261-279 (2001).

[43] S.Turek, *Efficient solvers for incompressible flow problems: An algorithmic approach in view of computational aspects*, (Springer, 1998/99).

[44] E.van Ruymbeke, R.Keunings, V.Stephenne, A.Hangenaars, C.Bailly, *Evaluation of reptation models for predicting the linear viscoelastic properties of entangled linear polymers*, *Macromolecules*, **35**, pp 2689-2699 (2002).

[45] H.K.Versteeg, W.Malalasekera, *An introduction to computational fluid dynamics: The finite volume method*, (Longmann Group Ltd, 1995).

[46] P. Wapperom, R. Keunings, *Impact of Decoupling Approximation Between Stretch and Orientation in Rheometrical and Complex Flow of Entangled Polymers*, *J. Non-Newtonian Fluid Mech.*, **122** , 33-43 (2004).

[47] P.Wapperom, R.Keunings, *Numerical simulations of branched polymer melts in transient complex flow using pom-pom models*, *J.Non-Newtonian Fluid Mech.* **97** pp. 267-281. (2001).

[48] P.Wapperom, R.Keunings, Simulation of linear polymer melts in transient complex flow, *J.Non-Newtonian Fluid Mech.* **95** pp. 67-83. (2000).

[49] P.Wapperom, R.Keunings, V.Legat, *The backward-tracking Lagrangian particle method for transient viscoelastic flows*, *J.Non-Newtonian Fluid Mech.*, **91**, pp. 273-295 (2000).

[50] P.Wesseling, Principles of computational fluid dynamics, *Springer-Verlag Berlin Heidelberg New York*, (2001).

[51] S.A.White, D.G.Baird, *The importance of extensional flow properties on planar entry flow patterns of polymer melts J.Non-Newtonian Fluid Mech.* **20**, pp 93-102 (1986).

[52] S.A.White,A.D.Gotsis, D.G.Baird, *Review of the entry flow problem: experimental and numerical. J.Non-Newtonian Fluid Mech.* **24**, pp 121-160 (1987).

[53] J.L.White, A.Kondo, *Flow patterns in polyethylene and polystyrene melts during extrusion through a die entry region: measurement and interpretation J.Non-Newtonian Fluid Mech.* **3**, pp 41-64 (1977).

CURRICULUM VITAE

13.06.1978	geboren in Zielona Gora
1985 – 1993	Besuch der Grundschule in Stary Kisielin
1993 – 1997	Besuch des Gymnasiums in Zielona Gora
Okt. 1997	Beginn des Studiums im Fach Mathematik an der Universität Zielona Gora
Apr. 2002 – Aug. 2002	Stipendium an der Universität Kaiserslautern
Aug. 2002	Diplom in Mathematik an der Universität Zielona Gora
seit Apr. 2003	Doktorand am Fraunhofer Institut für Techno- und Witschaftsmathematik