Multiscale Simulation of Polymeric Fluids using Sparse Grids

Dissertation

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Alexander Rüttgers

aus

Düren

Bonn 2016

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Gutachter: Prof. Dr. Michael Griebel

2. Gutachter: Prof. Dr. Marc Alexander Schweitzer

Tag der Promotion: 25. Oktober 2016

Erscheinungsjahr: 2016

Für Katharina

Summary

The numerical simulation of *non-Newtonian* fluids is of high practical relevance since most complex fluids developed in the chemical industry are not correctly modeled by classical fluid mechanics. Non-Newtonian effects are used in practice, for instance, to improve the oil flow in pipelines, to model solute transport in groundwater flows or to optimize the drop generation in inkjet printing. Furthermore, many fluids in nature like blood, egg white or even lava are non-Newtonian.

Classical approaches for the modeling of non-Newtonian polymeric fluids base on an additional stress tensor in the Navier-Stokes fluid equations. The stress tensor evolves according to a differential or integral constitutive equation. These so-called *macroscopic models* are comparatively easy to approximate but feature two serious drawbacks that limit their usability: they involve high modeling errors and they are prone to numerical instabilities. The numerical instabilities are referred to as the *high Weissenberg number problem*.

Both drawbacks can be avoided in a *multiscale approach*. This approach directly models the kinetic equations of the microscopic polymeric structure and leads to high-dimensional stochastic or Fokker-Planck equations. In this thesis, we implement a multiscale multi-bead-spring chain model into the three-dimensional Navier-Stokes solver *NaSt3DGPF* developed at the Institute for Numerical Simulation, University of Bonn. It is the first implementation of such a high-dimensional polymer model into a three-dimensional flow solver. Using the high-dimensional model, we present novel numerical simulations for a square-square contraction flow problem. We then compare the results of our 3D simulations with that of experimental measurements from the literature and obtain a very good agreement.

Up to now, high-dimensional multiscale approaches are hardly used in practical applications as grid-based methods suffer from the so-called *curse of dimensionality*. This term describes the exponential increase of the grid point number with the problem dimension. Consequently, the CPU running time of such a coupled system is dominated by the high-dimensional polymer model and hardly affected by the complexity of the macroscopic flow.

We combine two approaches to weaken or even break the curse of dimensionality for the considered problem. First, we use a domain decomposition with MPI to allow for massively parallel computations. Second, we employ a dimension-adaptive sparse grid variant, the *combination technique*, to reduce the computational complexity of the multiscale model. The main idea of the combination technique is to combine a sequence of coarse and anisotropic full grid solutions to obtain an approximation to a fine sparse grid solution. The small size full grid problems can be solved in parallel. The dimension-adaptive refinement process balances the computational effort for the spatial grid, the temporal grid, the stochastic resolution and the modeling accuracy. This is a novel approach in the context of multiscale polymer physics. Interestingly, the balancing of the different global error terms can be used for a wide range of similar problems. This is important, for instance, in applications that are related to *Uncertainty Quantification* where a large number of simulation runs with modified parameters is evaluated statistically.

Both approaches, the parallelization of the multiscale algorithm and the combination technique, can be combined. A combined approach has a perfect parallel scaling behavior on large clusters and can be ideally used in exascale computing.

Contents

Introduction 1

I. Multiscale simulation of polymeric fluids 9

1. Mathematical modeling of polymeric fluids 11

1.1. Macroscopic models for non-Newtonian fluids . . . 11

1.1.1. Differential stress tensor models . . . 11

1.1.2. The log-conformation method . . . 13

1.2. Multiscale polymeric fluid models . . . 14

1.2.1. Dynamics of multi-bead-spring chains . . . 14

1.2.2. Derivation of the Fokker-Planck equation . . . 17

1.2.3. Multiscale Navier-Stokes-BCF system . . . 23

1.2.4. Existence of global weak solutions for multiscale model . . . 31

2. Numerical modeling of polymeric fluids 35

2.1. Monte Carlo quadrature for diffusion problems . . . 35

2.1.1. General concepts . . . 35

2.1.2. Random number generation . . . 38

2.1.3. Variance reduction schemes . . . 42

2.1.4. Higher-order Quasi Monte Carlo methods . . . 44

2.2. Discretization of the multiscale Navier-Stokes-BCF system . . . 48

2.2.1. Spatial discretization . . . 48

2.2.2. Temporal discretization . . . 51

2.3. Complexity and parallelization . . . 55

2.3.1. Complexity of full grid approach . . . 55

2.3.2. Parallelization . . . 56

3. Numerical results on full grids 59

3.1. Homogeneous flows . . . 59

3.1.1. Dumbbell models . . . 60

3.1.2. Spring-chain models . . . 62

3.1.3. Reconstruction of the probability density function . . . 66

3.2. Complex multiscale flows . . . 69

3.2.1. Planar contraction flow . . . 69

3.2.2. 4 : 1 square-square contraction flow . . . 76

II. Sparse grids for polymeric fluids 93

4. Sparse grids 95
4.1. General concepts 95
4.1.1. Hierarchical increment space 96
4.1.2. Sparse grids 97
4.1.3. Assumptions on the regularity of the solution 99
4.2. Sparse grid combination technique 101
4.2.1. Error bounds for combination technique 103
4.2.2. Analysis of computational complexity 104
4.3. Relation multilevel Monte Carlo - sparse grids 107
4.3.1. Multilevel Monte Carlo method 107
4.3.2. Interpretation of MLMC as sparse grid approximation 109
4.3.3. Dimension-adaptive sparse grids 111

5. Discretization of multiscale polymer model on sparse grids 113
5.1. Dimension-adaptive combination technique 113
5.1.1. Generalized index sets 114
5.1.2. Error estimation in adaptive algorithm 116
5.2. Dimension-adaptivity for multiscale polymeric fluids 119
5.2.1. Effect of coupled system on adaptive refinement 121
5.2.2. Sparse dimension-adaptive algorithm 127

6. Numerical results on sparse grids 129
6.1. Couette flows 129
6.1.1. Homogeneous Couette flows as a stochastic ODE problem 131
6.1.2. General Couette flows as a stochastic PDE problem 135
6.1.3. Effect of modified flow conditions on index sets 143
6.2. Extensional flows 146

Conclusion 157

Bibliography 161

Introduction

Motivation

Non-Newtonian fluids: "panta rhei"

The famous phrase "panta rhei" which means "everything flows" by the Greek philosopher Heraclitus of Ephesus expresses that everything changes with time and that fluid motion is all around us. The well-known Navier-Stokes equations describe a class of liquids and gases that is denoted as *Newtonian*. A Newtonian fluid fulfills Newton's law of viscosity which states that the shear stress is proportional to the shear rate with the viscosity as constant of proportionality. This fluid class contains water and most gases.

In the 20th century various complex fluids have been developed in the chemical industry which are not correctly described by classical fluid mechanics. Paint, engine oils with polymeric additives, toothpaste and shampoo are examples for these kinds of fluids. Furthermore, many fluids in nature like blood, egg white or even lava violate Newton's law of viscosity and have a stress-dependent viscosity. In general, all fluids with a microstructure larger in size than the atomic scale show a *non-Newtonian* behavior. In the 1920's, Eugene Bingham, an American chemistry professor, coined the term *rheology* for the study of non-Newtonian fluids - inspired by the aphorism "panta rhei" as made famous by Heraclitus.

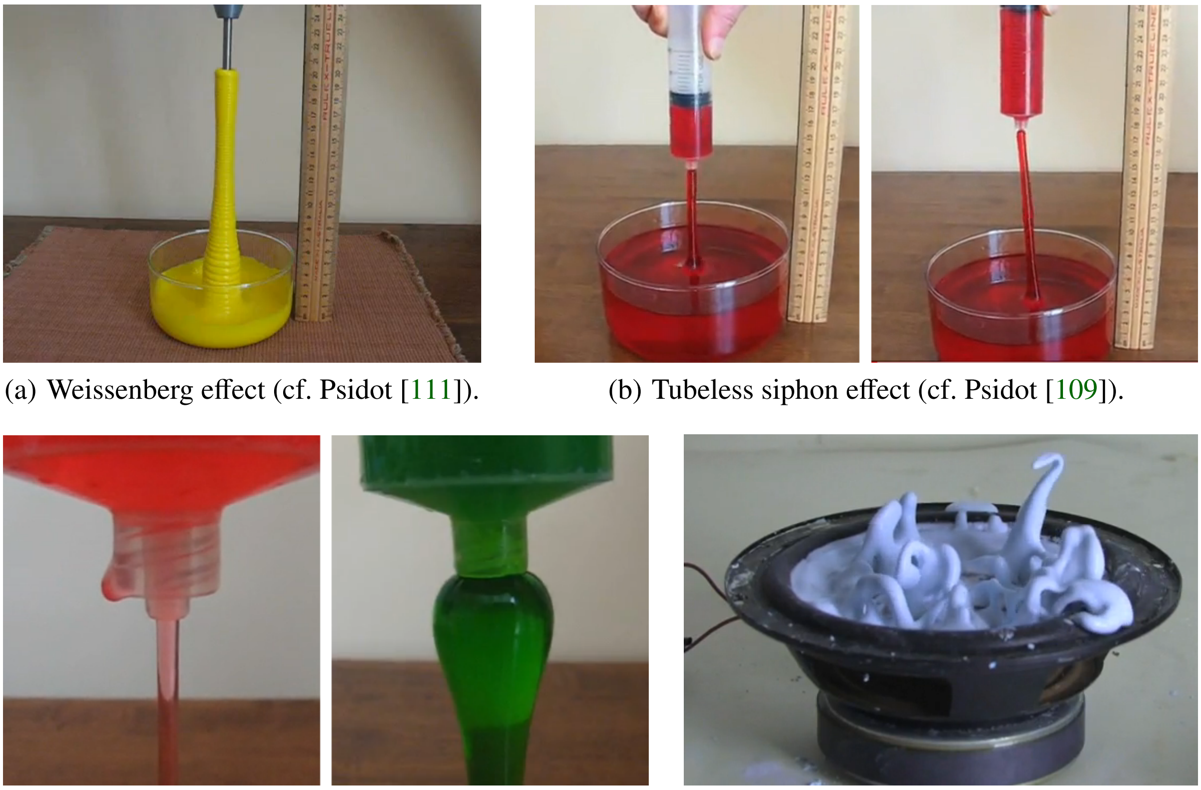
This thesis focuses on dilute *polymeric fluids* which are a subclass of non-Newtonian fluids. Polymeric fluids consist of long-chain molecules immersed in a Newtonian fluid. The fluid's motion leads to deformations and new orientations of these molecules. The molecules attempt to resume their initial configuration which causes an elastic force on the macroscopic fluid in opposite direction. Since polymeric fluids are not only affected by viscous forces but also by elastic forces, they are often named *viscoelastic fluids*.

Phenomena in experiments and industrial use

We illustrate the different behavior of Newtonian and non-Newtonian fluids by discussing several non-Newtonian phenomena observed in experiments.

A well known non-Newtonian phenomenon is the *Weissenberg effect*; cf. Fig. 0.1 (a). Here, a rotating rod leads to an upward movement of the fluid against gravity. In contrast to this, a Newtonian fluid such as water would simply be pushed away due to centrifugal forces. The non-Newtonian movement results from non-zero normal stresses that cause a tension in flow direction. We also know this effect from baking in the kitchen when dough moves upwards along the beaters of a handheld electric mixer.

The *tubeless syphon effect* occurs when certain non-Newtonian fluids become extended. Then, large elastic forces lead to the behavior that is illustrated in Fig. 0.1 (b). A syringe is put into the fluid and filled. The filling process can be continued even if the syringe has been removed from the fluid. In that



(a) Weissenberg effect (cf. Psidot [111]).

(b) Tubeless siphon effect (cf. Psidot [109]).

(c) Barus effect (cf. Psidot [110, 112]).

(d) Shear thickening effect on loudspeaker (cf. Bend [10]).

Figure 0.1.: Non-Newtonian phenomena in fluid experiments.

case, a water column connects the syringe with the ground reservoir. This connection would directly break for a Newtonian fluid such as water.

A further effect that is caused by normal stresses is the *Barus effect*. In Fig. 0.1 (c) we compare the effect for a Newtonian fluid colored in red (left) and a non-Newtonian fluid colored in green (right). In contrast to the Newtonian fluid, the jet that forms outside a pipe is much increased in diameter in the non-Newtonian case.

As mentioned before, a non-Newtonian fluid has a complex relation between shear stress and shear deformation. A lot of viscoelastic fluids are either *shear-thinning* or *shear-thickening*. In that case, the viscosity decreases or increases with the shearing, respectively. Fig. 0.1 (d) illustrates the effect of shear-thickening. A mixture of corn-starch and water is put on a loudspeaker that vibrates with a frequency of about 30 *Hz*. The fluid is disturbed by the oscillations which leads to an increase in viscosity and forms weird structures.

Non-Newtonian fluids are often used in industrial applications to optimize the production process. One example is the use of *drag reducing agents* in the oil industry. Usually, the flow rate of oil in a pipeline is limited due to turbulence effects. Drag reducing agents (DRA) are added in small concentrations to the oil and decrease turbulence effects. As a result, the oil can be transported with a higher flow rate for the same pressure or, alternatively, with the same flow rate as before but for a reduced pressure.

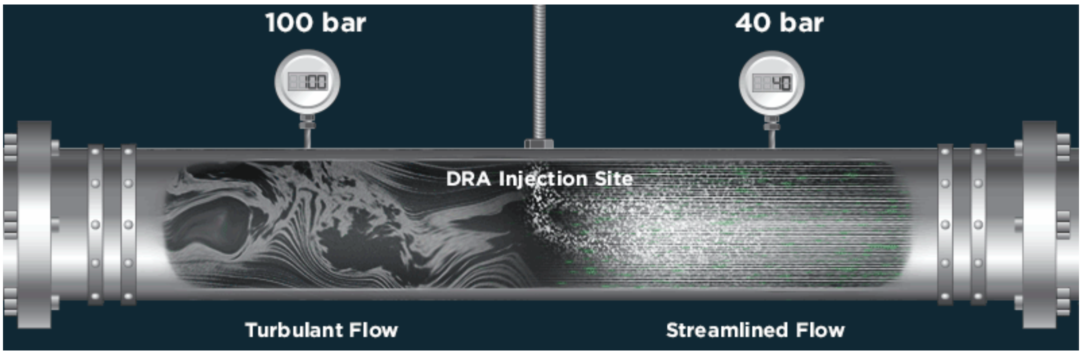


Figure 0.2.: Illustration of turbulent drag reduction for oil transport in a pipeline. The drag reducing agent (DRA) is added at the channel's center and changes the turbulent oil flow into a laminar flow. The figure is taken from an advertisement for the TURBOFLO DRA by Flowchem [62].

This allows oil companies to save energy and money. We illustrate this principle in Fig. 0.2 which is part of an advertisement from the company *Flowchem* [62]. Interestingly, there are considerations to also use DRAs in medicine. For instance, DRAs might be used to improve the blood flow for patients with arterial stenoses which is a narrowing in a blood vessel.

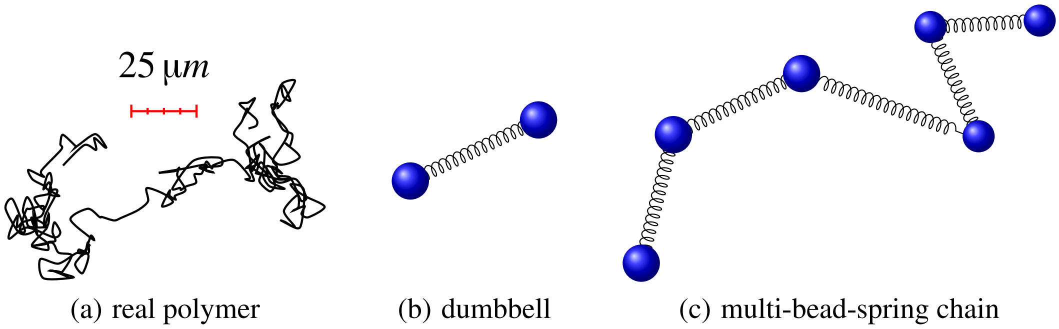
Further applications in which non-Newtonian effects have to be taken into account are the modeling of solute transport in groundwater flows, the optimization of the oil or gas fracking process or the drop generation in inkjet printing. The first application is of major importance in situations in which toxic chemical solutions might pollute drinking water reservoirs. Next, fracking is a technique in which a shear-thinning fluid is injected with high-pressure into a borehole to increase the output of gas or oil. Finally, viscoelasticity is required to allow for faster inkjet printers since it can reduce the creation of unwanted satellite drops while printing.

Almost all numerical simulations in the literature employ coarse approximation models that are not able to fully describe the complex fluids in these applications. This strongly motivates further numerical studies with more advanced multiscale models.

About this thesis

In this dissertation, we couple an existing three-dimensional flow solver with a high-dimensional multiscale model for polymeric fluids. Due to the enormous complexity of multiscale polymer models, this is the first time that a high-dimensional polymer model is used for the simulation of three-dimensional flows. In order to cope with the high-dimensional problem, we employ a dimension-adaptive sparse grid approach. This method strongly reduces the complexity of the problem and allows us to perform simulations that otherwise would require months of computing time on a parallel cluster. Note also that this is the first application of dimension-adaptive sparse grids to non-Newtonian fluids.

In the following, we briefly discuss different modeling approaches for viscoelastic fluids. Then, we consider the curse of dimensionality in the context of our specific multiscale model.



(a) real polymer (b) dumbbell (c) multi-bead-spring chain

Figure 0.3.: A real polymeric molecule in comparison with two different model approximations.

Modeling of viscoelastic fluids

Mathematical modeling of viscoelastic fluids usually involves an additional stress tensor in the Navier-Stokes equations. There are different approaches in literature to determine the corresponding stress tensor components. We distinguish two major modeling approaches which are *macroscopic methods* on the one hand and *multiscale methods* on the other hand. Macroscopic methods model the stress tensor evolution according to an additional differential or integral constitutive equation. Some often used macroscopic models in literature are, for instance, the Oldroyd-B [98], the FENE-P [102] and the PTT model [106]. A comprehensive description of macroscopic models is given in the book by Owens and Phillips [100]. The major advantage of a macroscopic approach is its computational simplicity. Therefore, these models have a computational complexity that is comparable to classical Newtonian models. On the downside, macroscopic methods have two important drawbacks that limit their usability. These drawbacks are

- the additional *modeling error* due to the use of *closure approximations* and
- their sensitivity for the so-called *high Weissenberg number problem* (HWNP).

Most macroscopic constitutive equations are derived from kinetic equations of the microscopic polymeric structure. In general, the constitutive equations are not closed so that an additional modeling assumption, a closure approximation, is necessary. This closure leads to an additional modeling error. Furthermore, the *high Weissenberg number problem* [68] denotes the breakdown of numerical schemes beyond a critical Deborah or Weissenberg number. These dimensionless numbers characterize elastic forces in the fluid.

Both drawbacks can be avoided in a multiscale approach which this thesis concentrates on. These approaches directly solve the kinetic equations of the microscopic polymeric structure. The macroscopic stress tensor then results from the internal configurations of the molecular structure. A molecular structure as in nature is shown in Fig. 0.3 (a). A multiscale polymer model is a simplified representation of the complex polymer structure as, for instance, a dumbbell or a more complex multi-bead-spring chain; cf. Fig. 0.3 (b)–(c).

Two different approaches exist for the temporal evolution of the simplified polymer model on the microscale; see the survey by Keunings [70]. These are a stochastic approach [99] on the one hand and a Fokker-Planck-based approach [89] on the other hand. The Fokker-Planck approach evolves a probability density function, which describes the position and configuration of a polymer ensemble, over time. In contrast, the stochastic approach evolves the corresponding time-dependent random field according to a stochastic partial different equation. This thesis uses the multiscale approach in its stochastic formulation. However, both approaches lead to high-dimensional modeling spaces. Therefore, in contrast to macroscopic approaches, multiscale methods are often restricted to simple flows due to their computational complexity. As a result, they are hardly used in the literature even though they allow for much better modeling capabilities.

Breaking the curse of dimensionality

This thesis presents first-time multiscale simulation results for complex three-dimensional flows. Furthermore, we state a dimension-adaptive algorithm which strongly reduces the computational complexity of multiscale simulations.

As a motivation, we discuss the complexity of both polymer models in Fig. 0.3. The most simple polymer model that accounts for the fluid's elasticity is the dumbbell model. Even for a simple dumbbell, the resulting model equation is six-dimensional. More precisely, three dimensions are required for the dumbbell's position in the flow space and three dimensions are used to describe the dumbbell's orientation and extension in the so-called configuration space. Independent of the polymer model, the physical flow space is three-dimensional or less. But, for more complex polymer models, the configuration space becomes high-dimensional. Fig. 0.3 (c) shows a multi-bead-spring chain with $N = 5$ spring segments. The corresponding configuration space of the system is of dimension $3N = 15$. Therefore, a time-dependent multiscale simulation with a 5-segment spring-chain model in a three-dimensional flow space is a 19-dimensional problem (1D in time, 3D in flow space, 15D in configuration space). We show in this thesis that it is actually possible to perform such complex simulations. For this purpose, we consider two complementary approaches. These approaches are

- massively parallel computations using a domain decomposition approach with MPI and
- dimension-adaptive sparse grids which reduce the computational complexity of the mathematical model.

We perform parallel simulations on the parallel cluster *Atacama* that is operated by the Institute for Numerical Simulation and the Sonderforschungsbereich 1060 at the University of Bonn. For this purpose, the high-dimensional stochastic equations on the microscale have been implemented and parallelized in the existing Newtonian flow solver *NaSt3DGPF* [48, 30]. We note that the running time of *NaSt3DGPF* is then dominated by the high-dimensional microscopic equations and not by the macroscopic fluid equations any more. Since *NaSt3DGPF* is a full grid flow solver, classical sparse grid discretizations [18] cannot be directly applied. For this reason, we use the *combination technique* [53] as an alternative sparse grid representation. This approach allows us to reuse existing full grid solvers such as *NaSt3DGPF* and approximates a sparse grid solution by combining a sequence of coarse full grid solutions. Since an optimal balancing of the different problem dimensions is not known in advance, we employ a dimension-adaptive refinement algorithm. In the end, the dimension-adaptive combination technique allows us to reduce the computational complexity of the multiscale polymer model by several orders of magnitude.

Both approaches, a parallelization of the algorithm on CPU clusters and the dimension-adaptive combination technique, can be used simultaneously. This is due to the fact that the combination technique is intrinsically parallel as each subproblem can be computed independently. Then, the individual subproblems can also be solved in parallel by using the multiscale model with *NaSt3DGPF*. This adds a second level of parallelization. For this reason, our algorithm is perfectly suited for exascale computing on systems with thousands of processing units as considered, for instance, in the "Software for Exascale Computing" (SPPEXA) German priority programme (SPP) 1648. The parallel scaling behavior of our implementation was tested on the *JUROPA* cluster (about 18000 CPU cores) from the Jülich Supercomputing Centre (JSC) at Jülich Research Centre.

Contributions of this thesis

The main contributions of this thesis are as follows:

- We present the first implementation and parallelization of a high-dimensional multi-bead polymer model into a three-dimensional flow solver. Using this implementation, we obtain novel simulation results for three-dimensional square-square contraction flows. These findings extend further results published in Griebel and Rüttgers [51] for a simple dumbbell model. Furthermore, we compare the simulation outcome of the dumbbell model with more complex 3-segment and 5-segment spring-chains for three-dimensional flows. Even for high flow rates, our implementation was not affected by the *high Weissenberg number problem* (HWNP). In the literature, three-dimensional contraction flows are up to now only considered with macroscopic approaches such as the PTT model which suffer from stability issues for the flow rates of interest (cf. Chapter 5.2 in Sousa et al. [123]: "In the numerical calculations we are only able to probe a range of Deborah numbers much smaller than in the experiments, as a consequence of the HWNP and the limitations of the PTT model.").

- Due to the high complexity of multiscale polymer simulations, we employ a dimension-adaptive sparse grid approach to allow for simulations with a reduced complexity. For this purpose, our dimension-adaptive algorithm does not only refine the numerical meshes but also takes the modeling accuracy into account. This novel concept for polymeric fluids allows for an optimal balancing of the different error contributions such as the stochastic variance, the spatial error, the temporal error or the modeling accuracy. Our approach can be considered as a generalization of the Multilevel Monte Carlo (MLMC) method for stochastic ODEs by Giles [47] which balances the stochastic variance and the temporal error only.

Further contributions of this thesis:

- We compare different multi-bead-spring chain models that differ in the number of spring segments. For specific applications and certain spring gauges, the elastic stress tensor predictions of the models are close to each other. Then, a 4-segment spring chain model might deliver a good approximation to a 5-segment spring chain but for a lower computational effort. We employ this novel concept for polymer physics in our dimension-adaptive algorithm.

- We state an algorithm to generate equilibrium samples for a general N-segment multi-bead-spring chain with nonlinear spring forces. There are no existing random number generators for this specific application. Our algorithm bases on the rejection sampling by von Neumann [131].

- We compare our simulation outcome for a two-dimensional contraction flow with the literature results by Wapperom, Keunings and Legat [133]. The literature results analyze the differences between the multiscale FENE model and the macroscopic FENE-P model. Our numerical approach allows us to use much higher grid resolutions as in Wapperom, Keunings and Legat [133]. We therefore extend the literature findings with new grid-independent results.

- Our dimension-adaptive refinement algorithm efficiently determines coarse full grid solutions used in the sparse grid combination technique. At a first view, this refinement algorithm leads to a computational overhead. However, we show that the outcome of the refinement algorithm can be reused for a wide range of similar flow problems. This effect is of high practical relevance for applications in the context of *Uncertainty Quantification* where hundreds or even thousands of similar problems, with slightly modified flow conditions, are considered. In these specific applications, the cost for the adaptive algorithm is negligible.

Outline

The remainder of this thesis is organized as follows:

Chapter 1 provides the basic theory for the simulation of macroscopic and multiscale dilute polymeric fluids. We analyze the drawbacks of macroscopic polymer models and motivate the usage of multiscale approaches. Furthermore, we discuss the existence of global weak solutions for the multiscale model.

Chapter 2 introduces numerical methods for the discretization of partial and stochastic differential equations that occur in the multiscale polymeric model. Moreover, we describe the implementation and parallelization of our approach in the full grid flow solver *NaSt3DGPF* and analyze the resulting parallel scaling behavior on a CPU cluster.

Chapter 3 contains our numerical results on full grids. First, we discuss homogeneous flow problems which do not depend on the physical flow space but already feature high-dimensional configuration spaces. Then, we present novel multiscale simulation results in complex two- and three-dimensional flow spaces.

Chapter 4 introduces the concepts of sparse grids and the combination technique. Furthermore, the relation between Multilevel Monte Carlo and dimension-adaptive sparse grids is discussed.

Chapter 5 describes the application of sparse grids to polymer physics. As a main result, we present a dimension-adaptive sparse grid algorithm to cope with the high-dimensional polymer model equation.

Chapter 6 deals with our numerical results on sparse grids for Couette and extensional flow problems. The high-dimensional problems are tackled with a dimension-adaptive refinement that balances the spatial, the temporal, the stochastic and the modeling accuracy. Furthermore, we show that the outcome of the refinement process can be used efficiently for a large class of similar problems.

Danksagung

An dieser Stelle möchte ich die Gelegenheit nutzen, mich bei allen zu bedanken, die mich bei der Anfertigung dieser Dissertation unterstützt und motiviert haben.

Zunächst gilt mein Dank Prof. Dr. Michael Griebel für die Überlassung des Themas, die Bereitstellung exzellenter Arbeitsbedingungen und für die vielen Anregungen und Diskussionen. Prof. Dr. Marc Alexander Schweitzer danke ich für die Übernahme des Zweitgutachtens und für seine Fähigkeit zur Motivation. Weiterhin danke ich Prof. Dr. Herbert Koch und Prof. Dr. Clemens Simmer für Ihre Bereitschaft, in meiner Promotionskommission mitzuwirken.

Viele Institutskollegen und ehemaligen Mitarbeiter haben Teile dieser Arbeit Korrektur gelesen. Ich bedanke mich für zahlreichen Anmerkungen und Hinweise bei Dr. Margrit Klitz, Dr. Jutta Neuen, Markus Burkow, Dr. Markus Siebenmorgen, Dr. Peter Zaspel, Dr. Bram Metsch und Dr. Alexander Hullmann. Die freundschaftliche Atmosphäre am Institut für Numerische Simulation (INS) habe ich stets genossen.

I would also like to thank Dr. Patricia Sousa from the University of Porto, Portugal, for interesting discussions on viscoelastic contraction flows and for providing experimental results that have been used in Section 3.2.2 of this thesis.

Darüber hinaus bedanke ich mich bei der Universität Bonn, der DFG und dem Sonderforschungsbereich Transregio 32, die meine Arbeit am INS finanziert haben.

Besonderer Dank gilt auch meiner Frau Katharina, meinen Eltern und meiner Schwester für ihre Unterstützung und ihr Vertrauen.

Bonn, im Juni 2016 Alexander Rüttgers

Part I.

Multiscale simulation of polymeric fluids

1. Mathematical modeling of polymeric fluids

1.1. Macroscopic models for non-Newtonian fluids

In the first section, we focus on purely macroscopic models for non-Newtonian fluids. The term *macroscopic* in this context indicates that the equations can be derived from continuum mechanics. Historically, macroscopic approaches have been the methods of choice due to their computational simplicity compared to more advanced multiscale approaches described in Section 1.2.

A comprehensive description of macroscopic models is given in the books by Renardy [114], Macosko [90], Joseph [65], Owens and Phillips [100] and Böhme [13]; further details are also given in Claus [25]. We only discuss differential stress tensor models. An alternative macroscopic formulation is given by integral models such as the Rivlin-Sawyers equation or the K-BKZ equation.

1.1.1. Differential stress tensor models

We consider fluid flow in a bounded domain $\mathcal{O} \subset \mathbb{R}^3$ and refer to $\mathcal{O}$ as physical space. For any position $\boldsymbol{x} \in \mathcal{O}$ and any time $t \in \mathcal{T} = [0, t_{\max}] \subset \mathbb{R}$ the current state of a purely Newtonian system is described by the fluid velocity field $\boldsymbol{u} : (\boldsymbol{x}, t) \in \mathcal{O} \times \mathcal{T} \mapsto \boldsymbol{u}(\boldsymbol{x}, t) \in \mathbb{R}^3$ and the hydrodynamic pressure field $\boldsymbol{p} : (\boldsymbol{x}, t) \in \mathcal{O} \times \mathcal{T} \mapsto p(\boldsymbol{x}, t) \in \mathbb{R}$ with corresponding initial and boundary conditions. For an incompressible and isothermal viscoelastic flow the conservation of mass and momentum is given by the coupled system

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u}\right) = -\nabla p + \eta_s \Delta \boldsymbol{u} + \nabla \cdot \boldsymbol{\tau}_p \tag{1.1}$$

$$\nabla \cdot \boldsymbol{u} = 0. \tag{1.2}$$

In (1.1) and (1.2) the coefficient $\rho \in \mathbb{R}^+$ denotes the fluid's density, $\eta_s \in \mathbb{R}^+$ represents the fluid's viscosity and $\boldsymbol{\tau}_p$ is the second-order tensor for the polymeric stress contribution. These equations are coupled with initial conditions

$$\begin{aligned} \boldsymbol{u}(\boldsymbol{x}, 0) &= \boldsymbol{u}_0(\boldsymbol{x}), \\ p(\boldsymbol{x}, 0) &= p_0(\boldsymbol{x}), \\ \boldsymbol{\tau}_p(\boldsymbol{x}, 0) &= \boldsymbol{\tau}_0(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \mathcal{O}. \end{aligned}$$

Furthermore, one of the conditions

$$\boldsymbol{u}|_{\Gamma_1} = \boldsymbol{u}_0 \qquad \text{on the inflow boundary } \Gamma_1, \tag{1.3a}$$

$$\boldsymbol{u}|_{\Gamma_2} = 0 \qquad \text{on the no-slip boundary } \Gamma_2, \tag{1.3b}$$

$$\partial_{\boldsymbol{n}}(\boldsymbol{u} \cdot \boldsymbol{n})|_{\Gamma_3} = 0, \ \partial_{\boldsymbol{n}}(\boldsymbol{u} \cdot \boldsymbol{t})|_{\Gamma_3} = 0 \qquad \text{on the outflow boundary } \Gamma_3 \tag{1.3c}$$

holds for the velocity field on the boundary $\partial O = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3$. In this case, $\boldsymbol{n}$ denotes the outward pointing unit normal and $\boldsymbol{t}$ denotes the tangential vector on ∂O.

Most macroscopic methods only differ in their approach of modeling the polymeric stress tensor $\boldsymbol{\tau}_p$. A general macroscopic model that covers various types of viscoelastic fluids is given by Phan-Thien and Tanner [105, 106]. The so called Phan-Thien Tanner (PTT) model is defined as

$$\lambda\left(\frac{\partial \boldsymbol{\tau}_p}{\partial t} - \mathcal{L}\boldsymbol{\tau}_p - \boldsymbol{\tau}_p\mathcal{L}^T\right) + \mathrm{f}(\mathrm{tr}\,\boldsymbol{\tau}_p) = \eta_p \boldsymbol{D} \tag{1.4}$$

$$\mathrm{f}(\mathrm{tr}\,\boldsymbol{\tau}_p) = \begin{cases} 1 + \frac{\lambda\epsilon}{\eta_p}\,\mathrm{tr}\,\boldsymbol{\tau}_p & \text{(linear PTT)} \\ \exp\left(\frac{\lambda\epsilon}{\eta_p}\right) & \text{(exponential PTT)} \end{cases} \tag{1.5}$$

with the rate of deformation tensor $\boldsymbol{D} = \nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T$. Here, $\lambda \in \mathbb{R}^+$ denotes the relaxation time of the mesoscopic polymer system, $\eta_p \in \mathbb{R}^+$ the zero shear rate polymer viscosity and $\mathcal{L} = (\nabla\boldsymbol{u} - \xi\boldsymbol{D})$ is the effective velocity gradient tensor. Furthermore, the equations (1.4) and (1.5) contain two dimensionless real numbers ϵ (elongation parameter) and ξ (slip parameter) that are used to adapt the model system to a real fluid's behavior. Depending on the choice of $\mathrm{f}(\mathrm{tr}\,\boldsymbol{\tau}_p)$ in (1.5) the model system is either called linear or exponential PTT model. Several simplified non-Newtonian models can be derived from (1.4) and (1.5). We derive

- by $\epsilon = 0$, the Johnson-Segalman model,
- by $\xi = 0$, the simplified Phan-Thien Tanner model and
- by $\epsilon = \xi = 0$, the Oldroyd-B model.

Moreover, we recover the classical Newtonian fluid equations for the case $\lambda = 0$ which leads to $\boldsymbol{\tau}_p = 0$.

The coupled system of equations (1.1)–(1.5) can be written in a dimensionless form by scaling the equations with the characteristic units L_c (characteristic length in macroscopic flow), U_c (characteristic fluid velocity), ρ_c (fluid density, scaling pressure term with $1/(\rho {U_c}^2)$), and we normalize the polymeric stress tensor with $L_c/(U_c(\eta_s + \eta_p))$. For reasons of simplicity, we use the same notation as before to indicate the dimensionless pressure field p and the dimensionless velocity field $\boldsymbol{u}$. As a result, this leads to the following dimensionless system of equations

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = -\nabla p + \frac{1}{Re}\beta\Delta\boldsymbol{u} + \frac{1}{Re}\nabla\cdot\boldsymbol{\tau}_p \tag{1.6}$$

$$\nabla\cdot\boldsymbol{u} = 0 \tag{1.7}$$

$$\mathrm{f}(\mathrm{tr}\,\boldsymbol{\tau}_p) + De\,\overset{\triangledown}{\boldsymbol{\tau}}_p = 2(1-\beta)\boldsymbol{D} - De\,\xi\left(\boldsymbol{\tau}_p\boldsymbol{D} + \boldsymbol{D}\boldsymbol{\tau}_p\right) \tag{1.8}$$

$$\mathrm{f}(\mathrm{tr}\,\boldsymbol{\tau}_p) = \begin{cases} 1 + \frac{De\,\epsilon}{1-\beta}\,\mathrm{tr}\,\boldsymbol{\tau}_p & \text{(linear PTT)} \\ \exp\left(\frac{De\,\epsilon}{1-\beta}\right) & \text{(exponential PTT).} \end{cases} \tag{1.9}$$

The system of equations contains the three dimensionless parameter groups De (Deborah number), Re (Reynolds number) and β (viscosity ratio). They are defined as

$$Re = \frac{\rho_c U_c L_c}{\eta_s + \eta_p}, \qquad De = \frac{\lambda U_c}{L_c}, \qquad \beta = \frac{\eta_s}{\eta_s + \eta_p}. \tag{1.10}$$

We also note that the Deborah number is also known as Weissenberg number in the literature. Furthermore, equation (1.8) contains the upper convected derivative or Oldroyd derivative $\overset{\triangledown}{\boldsymbol{\tau}}_p$ which is defined as

$$\overset{\triangledown}{\boldsymbol{A}} \equiv \frac{\partial \boldsymbol{A}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{A} - \nabla\boldsymbol{u} \cdot \boldsymbol{A} - \boldsymbol{A} \cdot (\nabla\boldsymbol{u})^T \tag{1.11}$$

for an arbitrary second-order tensor $\boldsymbol{A}$.

The major advantage of a macroscopic approach as in (1.8) and (1.9) is its computational simplicity since the unknown $\boldsymbol{\tau}_p$ is of comparable complexity to p and $\boldsymbol{u}$. On the other hand, macroscopic methods have two important drawbacks that limit their usability. These drawbacks are

- the additional *modeling error* due to the use of *closure approximations* and
- their sensitivity for the so called *high Weissenberg number problem* (HWNP).

The additional *modeling error* results from the fact that most constitutive equations are derived from kinetic equations of the underlying polymeric molecules. In general, the constitutive equations are not closed so that an additional modeling assumption, the closure approximation, is necessary. However, in some simple cases there exists a formal equivalence between the kinetic equation and the resulting constitutive equation. One example for this is given by the *Oldroyd-B* model. This model predicts an unlimited shear stress in extensional flows and is therefore only applicable to flows with a dominant shear component.

The *high Weissenberg number problem* describes the failure of numerical simulation beyond a critical Deborah / Weissenberg number. It is, however, still not clear whether the HWNP is only a numerical problem or also a breakdown of the underlying constitutive law. A comprehensive description on this problem is given by Keunings [68].

1.1.2. The log-conformation method

In the following, we discuss a recently proposed macroscopic approach, the *log-conformation method*, which is able to reduce the *high Weissenberg number problem* to some extent. The HWNP is known since about 1970. One reason for the numerical breakdown is caused by large gradients of $\boldsymbol{\tau}_p$ that occur for high Deborah number flows. In regions with a high deformation rate, for instance at stagnation points, $\boldsymbol{\tau}_p$ can have an exponential profile. Classical discretization techniques employ a polynomial-based approximation to this exponential profile. The basic idea to overcome this numerical problem was given by Fattal and Kupferman [36, 37]. Here, a new variable that scales logarithmically with the stress tensor is introduced. An obvious choice is to consider the logarithm of the stress tensor $\boldsymbol{\tau}_p$. However, the logarithm only exists for the case that $\boldsymbol{\tau}_p$ is strictly positive definite. This cannot be guaranteed by the numerical scheme. The solution of Fattal and Kupferman is to evolve $\boldsymbol{\psi} = \log(\boldsymbol{c})$ instead of $\boldsymbol{\tau}_p$ in time. The variable $\boldsymbol{c}$ is called *conformation tensor* and is symmetric positive definite. On the downside, the numerical scheme has to ensure that the positive definiteness of $\boldsymbol{c}$ is preserved over time.

For the PTT model given in (1.8) and (1.9), the conformation tensor $\boldsymbol{c}$ is related to the stress tensor $\boldsymbol{\tau}_p$ via

$$\boldsymbol{\tau}_p = \frac{(1-\beta)}{De}(\boldsymbol{c} - \boldsymbol{Id}). \tag{1.12}$$

The method is called *log-conformation method* since it evolves the logarithm ψ of the conformation tensor. The transformation from $\boldsymbol{\psi}$ to $\boldsymbol{c}$ guarantees that $\boldsymbol{c}$ is positive definite by construction.

The log-conformation method removes the instability caused by an exponential stress tensor profile. According to Fattal and Kupferman, spurious instabilities that can also occur in viscometric flows for high Deborah numbers are not prohibited with this approach. Furthermore, there is a degree of freedom in the constitutive equation for $\boldsymbol{\psi}$ that is derived from (1.8) and (1.9). As a result, the development of the log-conformation method is still an active area of research and different variants of the method have been proposed in the literature; see e.g. Coronado et al. [27] and Knechtges, Behr and Elgeti [72].

We note that the multiscale approach described in Section 1.2 is less prone to the *high Weissenberg number problem* than macroscopic methods. More precisely, Mangoubi, Hulsen and Kupferman [91] state that a multiscale approach seems to be immune to the type of instability caused by the HWNP. Moreover, the authors derive a log-conformation equivalent to the multiscale approach but have to conclude that this approach gives no advantage over the classical multiscale formalism.

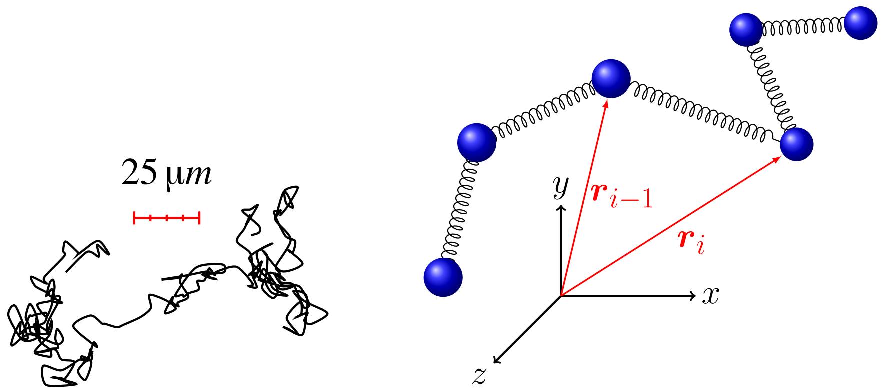
1.2. Multiscale polymeric fluid models

In this thesis, we use a multiscale micro-macro approach to describe viscoelastic fluids. This section focuses on the kinetic equations for the microscopic polymer structure. The polymer is modeled as a multi-bead-spring chain. In the following, we employ the notation that is given in the article by Barrett and Süli [2] and the book by Öttinger [99].

1.2.1. Dynamics of multi-bead-spring chains

In Fig. 1.1 we a show a realistic representation of a polymer on the left and its mathematical approximation with a multi-bead-spring chain on the right. A polymeric molecule is modeled as an arrangement of $N+1$ beads with mass m that are connected with N elastic springs without mass. As illustrated in Fig. 1.1 (b), the state of the chain is fully determined by the position of the $N+1$ beads denoted by $\boldsymbol{r}_i \in \mathcal{O} \subset \mathbb{R}^3$, with $i = 1, \dots, N+1$. The subset $\mathcal{O}$ has to be a bounded open Lipschitz domain and is denoted as the *flow space*. In general, we consider not only one single chain but a full ensemble of molecules. In this case, we define $\boldsymbol{R}_i$ as a multivariate random variable $\boldsymbol{R}_i : \omega \in \Omega_i \mapsto \boldsymbol{R}_i(\omega) = \boldsymbol{r}_i \in \mathcal{O} \subset \mathbb{R}^3$ with $i = 1, \dots, N+1$ where the outcome for a given event ω in the sample spaces Ω_i is the position of bead i. In the following, we use lowercase letters such as $\boldsymbol{r}$ and $\boldsymbol{q}$ to describe a single bead or spring realization in configuration space and furthermore uppercase letters such as $\boldsymbol{R}$ and $\boldsymbol{Q}$ to denote the random variables that induce a distribution according to the underlying probability measure.

A force balance equation for bead-spring chains Next, the dynamics of the polymer chain model is considered. For this purpose, we consider a time interval $\mathcal{T}$ which is either the closed interval $[0, t_{\max}]$ or the non-negative axis $[0, \infty]$. Then, the sequence of $N+1$ random variables $\boldsymbol{R}_i$ becomes a sequence of stochastic processes $\boldsymbol{R}_i(t)$, $t \in \mathcal{T}$. Thereby, for each fixed $t \in \mathcal{T}$, the function $\boldsymbol{R}_i(t, \cdot) : \omega \in \Omega_i \mapsto \boldsymbol{R}_i(t, \omega) \in \mathcal{O} \subset \mathbb{R}^3$ is a multivariate random variable. Then, for each $\omega \in \Omega_i$ the function $\boldsymbol{R}_i(\cdot, \omega) : t \in \mathcal{T} \mapsto \boldsymbol{R}_i(t, \omega) \in \mathcal{O} \subset \mathbb{R}^3$ is a real function defined on $\mathcal{T}$ that we could also write in a more simple notation as $\boldsymbol{r}_i : \mathcal{T} \to \mathcal{O}$. For this reason, we simplify our notation for the stochastic process by writing $\boldsymbol{R}_i(t)$ instead of $\boldsymbol{R}_i(t, \omega)$, i.e. we omit the underlying sample spaces.



(a) real polymer (b) multi-bead-spring chain

Figure 1.1.: Approximation of a polymeric molecule with a multi-bead-spring chain.

According to Newton's second law, the force balance equation for a single bead $\boldsymbol{r}_i(t)$ with mass m is

$$
\begin{aligned}
m\frac{\mathrm{d}^2\boldsymbol{r}_1(t)}{\mathrm{d}t^2} &= \boldsymbol{F}_1^{\text{drag}} + \boldsymbol{F}_1(\boldsymbol{r}_2(t) - \boldsymbol{r}_1(t)) && + \boldsymbol{B}_1(t) && (1.13a)\\
m\frac{\mathrm{d}^2\boldsymbol{r}_i(t)}{\mathrm{d}t^2} &= \boldsymbol{F}_i^{\text{drag}} + \boldsymbol{F}_{i+1}(\boldsymbol{r}_{i+1}(t) - \boldsymbol{r}_i(t)) - \boldsymbol{F}_i(\boldsymbol{r}_i(t) - \boldsymbol{r}_{i-1}(t)) && + \boldsymbol{B}_i(t) \quad \text{for } i = 2, \dots, N && (1.13b)\\
m\frac{\mathrm{d}^2\boldsymbol{r}_{N+1}(t)}{\mathrm{d}t^2} &= \boldsymbol{F}_{N+1}^{\text{drag}} - \boldsymbol{F}_{N+1}(\boldsymbol{r}_{N+1}(t) - \boldsymbol{r}_N(t)) && + \boldsymbol{B}_{N+1}(t). && (1.13c)
\end{aligned}
$$

Normally, we neglect inertial effect by assuming $m\frac{\mathrm{d}^2\boldsymbol{r}_i(t)}{\mathrm{d}t^2} \approx 0$ for $i = 1, \dots, N+1$ since the mass m of the beads is comparatively small; see Schieber and Öttinger [119] for a justification of this simplification. The right-hand side of (1.13) consists of three force terms:

- $\boldsymbol{F}_i^{\text{drag}} : \mathcal{O} \times \mathcal{T} \to \mathbb{R}^3$ is the *drag force* on bead i. The force opposes the relative motion of the bead through the fluid, $\boldsymbol{u}(\boldsymbol{r}_i(t), t) - \frac{\mathrm{d}(\boldsymbol{r}_i(t))}{\mathrm{d}t}$, and represents a kind of fluid resistance. Again, $\boldsymbol{u} : \mathcal{O} \times \mathcal{T} \to \mathbb{R}^3$ denotes the fluid's velocity field. The drag force is defined as

$$\boldsymbol{F}_i^{\text{drag}} = \boldsymbol{\zeta}\left(\boldsymbol{u}(\boldsymbol{r}_i(t), t) - \frac{\mathrm{d}(\boldsymbol{r}_i(t))}{\mathrm{d}t}\right) \tag{1.14}$$

  with a second-order tensor $\boldsymbol{\zeta}$. In the literature, different definitions are used for $\boldsymbol{\zeta}$. Here, we employ the simple ansatz $\boldsymbol{\zeta} = \operatorname{diag}(\zeta, \zeta, \zeta)$ that is derived from Stoke's law with $\zeta \in \mathbb{R}^+$ as constant friction coefficient ($[\zeta] = \frac{kg}{s}$); see Bird et al. [11, 12].

- $\boldsymbol{F}_i : D_i \subset \mathbb{R}^3 \to \mathbb{R}^3$ denotes an elastic *spring force* which results from a spring segment that connects bead i with bead $i+1$. The subdomains $D_i \subset \mathbb{R}^3$ are called *configuration spaces*. Depending on the spring force terms $\boldsymbol{F}_i$, the configuration spaces differ; see Section 1.2.3. In all cases, $\boldsymbol{F}_i$ contains a coefficient $H \in \mathbb{R}^+$ that is called the Hookean spring constant ($[H] = kg/s^2$).

- $\boldsymbol{B}_i : \mathcal{T} \to \mathbb{R}^3$ is a Brownian force that is caused by random collisions between bead i and adjacent molecules. We model the Brownian force on bead i as a three-component Wiener process according to

$$\boldsymbol{B}_i(t)\,\mathrm{d}t \;=\; \sqrt{2k_BT\zeta}\,\mathrm{d}\boldsymbol{W}_i(t). \tag{1.15}$$

Since we assume an isotropic noise in space, the Brownian force does not explicitly depend on its position in the flow space O. A Wiener process is Gaussian and can be charaterized by its first two moments. For each component $W_{i,j}(t)$ with $j = 1,2,3$ the expectation is

$$\langle W_{i,j}(t)\rangle \;=\; 0 \qquad \text{for } j = 1,2,3 \tag{1.16}$$

and the variance for $t_1, t_2 \in \mathcal{T}$ is

$$\langle W_{i,j}(t_1)W_{i,j}(t_2)\rangle \;=\; \int_0^{t_1}\int_0^{t_2} \delta\,(t' - t'')\;\mathrm{d}t''\,\mathrm{d}t' \;=\; \min(t_1, t_2). \tag{1.17}$$

The coefficient $\sqrt{2k_BT\zeta}$ in (1.15) results from the equipartition of energy theorem. In this case, k_B ($[k_B] = J/K$) is the Boltzmann constant and T denotes the thermodynamic temperature ($[T] = K$).

Dimensionless formulation The equations for the bead-spring chains (1.13a)–(1.13c) fully describe the dynamics of the polymer system. However, since we aim for coupling these equations with the dimensionless Navier-Stokes equations (1.6) and (1.7), we require this model system also in a dimensionless formulation. For this purpose, we employ l_c, λ, L_c, and U_c as intrinsic parameters of the system. We note that L_c and U_c are the same as in Section 1.1 for the non-dimensionalization of the Navier-Stokes equations. All in all, we choose

- $l_c = \sqrt{k_BT/H}$ as characteristic length-scale of a dumbbell ([l_c]=m),
- $\lambda = \frac{\zeta}{4H}$ as characteristic relaxation time of a dumbbell ([λ]=s),
- L_c as characteristic length of the macroscopic flow ([L_c]=m),
- and U_c as characteristic velocity of the macroscopic flow ([U_c]=m/s)

with H as Hookean spring constant as coefficient in $\boldsymbol{F}_i$. Using these dimensionless units, we define the dimensionless quantities

$$\boldsymbol{r}_i^* \;:=\; \boldsymbol{r}_i/L_c, \tag{1.18a}$$
$$\boldsymbol{u}^* \;:=\; \boldsymbol{u}/U_c, \tag{1.18b}$$
$$t^* \;:=\; U_c/L_c t \tag{1.18c}$$

for $i = 1, \ldots, N+1$. Furthermore, we use $\boldsymbol{F}_i = HL_c\boldsymbol{F}_i^*(\boldsymbol{r}_{i+1}^* - \boldsymbol{r}_i^*)$ and $\boldsymbol{W}_i(t) = (\frac{L_c^3}{U_c l_c^2})^{1/2}\,\boldsymbol{W}_i^*(t)$ to obtain the dimensionless spring forces and Wiener processes $\boldsymbol{F}_i^*$ and $\boldsymbol{W}_i^*$ and we transform the differential operators according to the chain rule. As a result, we obtain the dimensionless stochastic ODE system

$$\begin{bmatrix} \mathrm{d}\boldsymbol{R}_1(t) \\ \mathrm{d}\boldsymbol{R}_2(t) \\ \vdots \\ \mathrm{d}\boldsymbol{R}_{N+1}(t) \end{bmatrix} = \left(\begin{bmatrix} \boldsymbol{u}(\boldsymbol{R}_1(t), t) \\ \boldsymbol{u}(\boldsymbol{R}_2(t), t) \\ \vdots \\ \boldsymbol{u}(\boldsymbol{R}_{N+1}(t), t) \end{bmatrix} + \frac{1}{4De}\boldsymbol{B} \begin{bmatrix} \boldsymbol{F}_1\,(\boldsymbol{R}_2(t) - \boldsymbol{R}_1(t)) \\ \boldsymbol{F}_2\,(\boldsymbol{R}_3(t) - \boldsymbol{R}_2(t)) \\ \vdots \\ \boldsymbol{F}_N\,(\boldsymbol{R}_{N+1}(t) - \boldsymbol{R}_N(t)) \end{bmatrix} \right) \mathrm{d}t + \sqrt{\frac{1}{2De}} \begin{bmatrix} \mathrm{d}\boldsymbol{W}_1(t) \\ \mathrm{d}\boldsymbol{W}_2(t) \\ \vdots \\ \mathrm{d}\boldsymbol{W}_{N+1}(t) \end{bmatrix} \tag{1.19}$$

with given initial conditions for $\boldsymbol{R}_1(t), \dots, \boldsymbol{R}_{N+1}(t)$. Note that the asterisk notation for dimensionless units has been dropped for simplicity in (1.19) and in the following, analogously to Section 1.1. Moreover, equation (1.19) contains the Deborah number De defined in (1.10) as a non-dimensional constant which relates the polymer's relaxation time to the macroscopic observation time. Finally, the coupling between the different spring segments is given by the matrix

$$\boldsymbol{B} := \begin{pmatrix} 1 & 0 & & & \\ -1 & 1 & \ddots & & \\ 0 & \ddots & \ddots & \ddots & \\ & \ddots & \ddots & \ddots & 0 \\ & & \ddots & \ddots & 1 \\ & & & 0 & -1 \end{pmatrix} \in \mathbb{R}^{(N+1)\times N}. \tag{1.20}$$

We now rewrite equation (1.19) in a generic form. Therefore, we define

$$\boldsymbol{R}(t) := \begin{bmatrix} \boldsymbol{R}_1(t) \\ \boldsymbol{R}_2(t) \\ \vdots \\ \boldsymbol{R}_N(t) \end{bmatrix} \in \underbrace{O \times \cdots \times O}_{N+1 \text{ times}} \subset \mathbb{R}^{3(N+1)}, \tag{1.21}$$

and set

$$\boldsymbol{W}(t) := \begin{bmatrix} \boldsymbol{W}_1(t) \\ \boldsymbol{W}_2(t) \\ \vdots \\ \boldsymbol{W}_{N+1}(t) \end{bmatrix}, \qquad \mu(\boldsymbol{R}(t)) := \begin{bmatrix} \boldsymbol{u}(\boldsymbol{R}_1(t), t) \\ \boldsymbol{u}(\boldsymbol{R}_2(t), t) \\ \vdots \\ \boldsymbol{u}(\boldsymbol{R}_{N+1}(t), t) \end{bmatrix} + \frac{1}{4De}\boldsymbol{B} \begin{bmatrix} \boldsymbol{F}(\boldsymbol{R}_2(t) - \boldsymbol{R}_1(t)) \\ \boldsymbol{F}(\boldsymbol{R}_3(t) - \boldsymbol{R}_2(t)) \\ \vdots \\ \boldsymbol{F}(\boldsymbol{R}_{N+1}(t) - \boldsymbol{R}_N(t)) \end{bmatrix}$$

to obtain a generic stochastic differential equation

$$\mathrm{d}\boldsymbol{R}(t) = \mu(\boldsymbol{R}(t))\,\mathrm{d}t + \sigma \mathrm{d}\boldsymbol{W}(t) \tag{1.22}$$

with given initial conditions and $\sigma = \frac{1}{\sqrt{2De}}\,\mathbf{Id}$. We note that equation (1.22) is actually a short hand notation for an integral equation that consists of an ordinary Lebesgue integral (first sum on the RHS of (1.22)) and an Itō integral (second sum on the RHS of (1.22)).

1.2.2. Derivation of the Fokker-Planck equation

In this chapter we state a result from stochastic analysis that connects the stochastic differential equation (1.19) with a parabolic partial differential equation for the probability density function associated with the process $\boldsymbol{R}(t)$.

Theorem 1.1

Let $\boldsymbol{R}(t)$ be a $3(N+1)$-component stochastic process for which a probability density function $\psi : (\boldsymbol{r}, t) \in \mathbb{R}^{3(N+1)} \times \mathcal{T} \mapsto \psi(\boldsymbol{r}, t) \in \mathbb{R}^+$ exists in the Hölder space $C^{2,1}(\mathbb{R}^{3(N+1)} \times \mathcal{T})$. Furthermore, let $\boldsymbol{R}(0) = \boldsymbol{R}$ be *a square-integrable random variable with probability density function* $\psi_0 \in C^2(\mathbb{R}^{3(N+1)})$*. If* $\boldsymbol{R}(t)$ *evolves according to* (1.22) *for a globally Lipschitz continuous* μ *and* $\boldsymbol{c} = \sigma\sigma^T$*, then* ψ *evolves according to*

$$\frac{\partial\psi}{\partial t} + \sum_{j=1}^{3(N+1)} \frac{\partial}{\partial r_j}(\mu_j\psi) = \frac{1}{2}\sum_{i,j=1}^{3(N+1)} \frac{\partial^2(c_{ij}\psi)}{\partial r_i\,\partial r_j} \tag{1.23}$$

with $\psi(\boldsymbol{r}, 0) = \psi_0(\boldsymbol{r})$.

Proof: The proof is given as Corollary 5.2.10 in the book by Lapeyre, Pardoux and Sentis [81]. Note that in our application equation (1.23) simplifies to

$$\frac{\partial\psi}{\partial t} + \sum_{j=1}^{3(N+1)} \frac{\partial}{\partial r_j}(\mu_j\psi) = \frac{1}{4De}\Delta_{\boldsymbol{r}}\psi \tag{1.24}$$

since $\boldsymbol{c} = \sigma\sigma^T = \mathrm{diag}(\frac{1}{2De}, \dots, \frac{1}{2De})$.

Transformation of stochastic processes on spring-oriented variables The stochastic differential equation (1.22) or the corresponding Fokker-Planck equation (1.24) fully determine the mesoscopic system. Nevertheless, we are interested in a different formulation that better allows for a coupling with the macroscopic Navier-Stokes equations; see (1.6) and (1.7) in Section 1.1. For this purpose, we perform a change of variables to obtain an equivalent description of the multi-bead-spring chain by

$$\begin{bmatrix} \boldsymbol{r}_1(t) \\ \boldsymbol{r}_2(t) \\ \vdots \\ \vdots \\ \boldsymbol{r}_{N+1}(t) \end{bmatrix} \hookrightarrow \begin{bmatrix} \boldsymbol{r}_2(t) - \boldsymbol{r}_1(t) \\ \boldsymbol{r}_3(t) - \boldsymbol{r}_2(t) \\ \vdots \\ \boldsymbol{r}_{N+1}(t) - \boldsymbol{r}_N(t) \\ \frac{1}{N+1}\sum_{j=1}^{N+1} \boldsymbol{r}_j(t) \end{bmatrix} =: \begin{bmatrix} \boldsymbol{q}_1(t) \\ \boldsymbol{q}_2(t) \\ \vdots \\ \boldsymbol{q}_N(t) \\ \boldsymbol{x}(t) \end{bmatrix}, \tag{1.25}$$

which is then applied to the sequence of $N+1$ stochastic processes. We denote these stochastic processes as $\hat{\boldsymbol{Q}}_i(t) \in D_i \subset \mathbb{R}^3$ and $\boldsymbol{X}(t) \in O$. Again, the underlying probability space Ω is not explicitly mentioned. The stochastic processes $\hat{\boldsymbol{Q}}_i(t)$ describe an ensemble of connecting vectors $\boldsymbol{q}_i(t) \in D_i \subset \mathbb{R}^3$ between two adjacent beads i and $i+1$ with a distribution of $\hat{\boldsymbol{Q}}_i(t)$ according to the underlying probability measure.

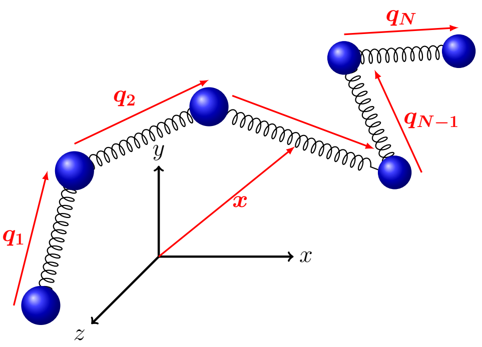


Figure 1.2.: Description with spring-oriented variables $\boldsymbol{q}_1, \dots, \boldsymbol{q}_N$.

The subdomains D_i are the configuration spaces that were previously defined in Section 1.2.1 for the spring forces terms $\boldsymbol{F}_i$. The random variable $\boldsymbol{X}(t) \in \mathcal{O}$ represents the bead system's center of mass. We illustrate the new system of variables in Fig. 1.2.

The change of variables in (1.25) leads to a modified force balance equation. Therefore, equation (1.19) has to be rewritten as

$$\begin{aligned}\begin{bmatrix} \mathrm{d}\hat{\boldsymbol{Q}}_1(t) \\ \mathrm{d}\hat{\boldsymbol{Q}}_2(t) \\ \vdots \\ \mathrm{d}\hat{\boldsymbol{Q}}_N(t) \\ \mathrm{d}\boldsymbol{X}(t) \end{bmatrix} &= \left(\begin{bmatrix} \boldsymbol{u}(\boldsymbol{R}_2(t),t) - \boldsymbol{u}(\boldsymbol{R}_1(t),t) \\ \boldsymbol{u}(\boldsymbol{R}_3(t),t) - \boldsymbol{u}(\boldsymbol{R}_2(t),t) \\ \vdots \\ \boldsymbol{u}(\boldsymbol{R}_{N+1}(t),t) - \boldsymbol{u}(\boldsymbol{R}_N(t),t) \\ \frac{1}{N+1}\sum_{j=1}^{N+1} \boldsymbol{u}(\boldsymbol{R}_j(t),t) \end{bmatrix} - \frac{1}{4De}\hat{\boldsymbol{A}} \begin{bmatrix} \boldsymbol{F}_1\left(\hat{\boldsymbol{Q}}_1(t)\right) \\ \boldsymbol{F}_2\left(\hat{\boldsymbol{Q}}_2(t)\right) \\ \vdots \\ \boldsymbol{F}_N\left(\hat{\boldsymbol{Q}}_N(t)\right) \end{bmatrix} \right) \mathrm{d}t \\ &\quad + \sqrt{\frac{1}{2De}} \begin{bmatrix} \mathrm{d}\boldsymbol{W}_2(t) - \mathrm{d}\boldsymbol{W}_1(t) \\ \mathrm{d}\boldsymbol{W}_3(t) - \mathrm{d}\boldsymbol{W}_2(t) \\ \vdots \\ \mathrm{d}\boldsymbol{W}_{N+1}(t) - \mathrm{d}\boldsymbol{W}_N(t) \\ \frac{1}{N}\sum_{j=1}^{N+1} \mathrm{d}\boldsymbol{W}_j(t) \end{bmatrix} \end{aligned} \tag{1.26}$$

with

$$\hat{\boldsymbol{A}} := \begin{pmatrix} 2 & -1 & & & 0 \\ -1 & 2 & \ddots & & \\ 0 & \ddots & \ddots & \ddots & \\ & \ddots & \ddots & \ddots & -1 \\ & & \ddots & -1 & 2 \\ 0 & & & 0 & 0 \end{pmatrix} \in \mathbb{R}^{(N+1)\times N}. \tag{1.27}$$

The matrix $\hat{\boldsymbol{A}}$ has only zero entries in its last row as the spring force terms cancel out in the sum $\sum_{j=1}^{N+1} d\boldsymbol{r}_j(t)$. In practice, only a submatrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ of $\hat{\boldsymbol{A}}$ is considered. This matrix can be obtained by omitting the last row of $\hat{\boldsymbol{A}}$ in (1.27). The resulting matrix

$$\boldsymbol{A} := \begin{pmatrix} 2 & -1 & & & 0 \\ -1 & 2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ 0 & & & -1 & 2 \end{pmatrix} \in \mathbb{R}^{N\times N} \tag{1.28}$$

is called *Rouse matrix* and fulfills $\boldsymbol{A} = \boldsymbol{B}^T\boldsymbol{B}$ with the matrix $\boldsymbol{B}$ as defined in (1.20).

For a better numerical treatment, (1.26) is further simplified by the following assumptions (cf. LeBris and Lelièvre [83]):

- the length of a polymer is much smaller than the spatial variation of the velocity field $\boldsymbol{u}$, i.e. the Taylor expansion of $\boldsymbol{u}$ at $\boldsymbol{x}$

  $$\boldsymbol{u}(\boldsymbol{r}_i(t), t) \approx \boldsymbol{u}(\boldsymbol{x}(t), t) + \nabla\boldsymbol{u}(\boldsymbol{x}(t), t)(\boldsymbol{r}_i(t) - \boldsymbol{x}(t)) + O\Big((\boldsymbol{r}_i(t) - \boldsymbol{x}(t))^2\Big)$$

  leads, with second order accuracy in $(\boldsymbol{r}_i(t) - \boldsymbol{x}(t))$, to the simplifications

  $$\boldsymbol{u}(\boldsymbol{r}_{i+1}(t), t) - \boldsymbol{u}(\boldsymbol{r}_i(t), t) \approx \nabla\boldsymbol{u}(\boldsymbol{x}(t), t)\, \boldsymbol{q}_i(t) \tag{1.29}$$

  $$\frac{1}{N+1}\sum_{j=1}^{N+1} \boldsymbol{u}(\boldsymbol{r}_j(t), t) \approx \boldsymbol{u}(\boldsymbol{x}(t), t) \tag{1.30}$$

  in equation (1.26).

- The term $\frac{1}{N+1}\sum_{j=1}^{N+1} \boldsymbol{u}(\boldsymbol{R}_j(t), t)\, dt$ in the last row of (1.26) is of macroscopic size and therefore much larger than the microscopic variation $\frac{1}{N}\sum_{j=1}^{N+1} d\boldsymbol{W}_j(t)$ due to Brownian forces on the $N + 1$ beads. It can be shown that the Brownian motion scales quadratically in the ratio of microscopic length scale l_c to macroscopic length scale L_c when the equations are nondimensionalized. We therefore approximate in the last row

  $$\frac{1}{N}\sum_{j=1}^{N+1} \boldsymbol{W}_j(t) \approx 0. \tag{1.31}$$

As a result (1.26) simplifies to a sequence of two stochastic differential equations

$$\begin{bmatrix} \mathrm{d}\hat{\boldsymbol{Q}}_1(t) \\ \mathrm{d}\hat{\boldsymbol{Q}}_2(t) \\ \vdots \\ \mathrm{d}\hat{\boldsymbol{Q}}_N(t) \end{bmatrix} = \left(\begin{bmatrix} \nabla\boldsymbol{u}(\boldsymbol{X}(t), t)\, \hat{\boldsymbol{Q}}_1(t) \\ \nabla\boldsymbol{u}(\boldsymbol{X}(t), t)\, \hat{\boldsymbol{Q}}_2(t) \\ \vdots \\ \nabla\boldsymbol{u}(\boldsymbol{X}(t), t)\, \hat{\boldsymbol{Q}}_N(t) \end{bmatrix} - \frac{1}{4De}\boldsymbol{A} \begin{bmatrix} \boldsymbol{F}_1\left(\hat{\boldsymbol{Q}}_1(t)\right) \\ \boldsymbol{F}_2\left(\hat{\boldsymbol{Q}}_2(t)\right) \\ \vdots \\ \boldsymbol{F}_N\left(\hat{\boldsymbol{Q}}_N(t)\right) \end{bmatrix} \right) \mathrm{d}t + \sqrt{\frac{1}{2De}} \begin{bmatrix} \mathrm{d}\boldsymbol{W}_2(t) - \mathrm{d}\boldsymbol{W}_1(t) \\ \mathrm{d}\boldsymbol{W}_3(t) - \mathrm{d}\boldsymbol{W}_2(t) \\ \vdots \\ \mathrm{d}\boldsymbol{W}_{N+1}(t) - \mathrm{d}\boldsymbol{W}_N(t) \end{bmatrix} \tag{1.32}$$

$$\mathrm{d}\boldsymbol{X}(t) = \boldsymbol{u}(\boldsymbol{X}(t), t)\, \mathrm{d}t \tag{1.33}$$

for the given initial conditions $\hat{\boldsymbol{Q}}(0)$ and $\boldsymbol{X}(0)$. Equation (1.33) appears as a deterministic equation for the trajectory of the dumbbell particles. We note, however, that $\boldsymbol{u}(\boldsymbol{X}, t)$ is also a random field since the initial condition $\boldsymbol{X}(0)$ is a random field.

Equations (1.32) and (1.33) in combination with the Navier-Stokes equations are used for the CONNFFESSIT approach; see Laso and Öttinger [82]. Here, individual realizations of $(\hat{\boldsymbol{Q}}(0), \boldsymbol{X}(0))$ evolve according to (1.32) and (1.33). This can be seen as a Lagrangian or particle based description of the flow field.

Fokker-Planck equation in chain specific variables The Fokker-Planck equation (1.24) can also be written in a new coordinate system $(\boldsymbol{q_1}, \dots, \boldsymbol{q_N}, \boldsymbol{x})$ with the change of variables (1.25). For this purpose, we write the simplified system of stochastic ODEs (1.32) and (1.33) in the combined form

$$\begin{bmatrix} \mathrm{d}\hat{\boldsymbol{Q}}_1(t) \\ \mathrm{d}\hat{\boldsymbol{Q}}_2(t) \\ \vdots \\ \mathrm{d}\hat{\boldsymbol{Q}}_N(t) \\ \mathrm{d}\boldsymbol{X}(t) \end{bmatrix} = \left(\begin{bmatrix} \nabla\boldsymbol{u}(\boldsymbol{X}(t),t)\,\hat{\boldsymbol{Q}}_1(t) \\ \nabla\boldsymbol{u}(\boldsymbol{X}(t),t)\,\hat{\boldsymbol{Q}}_2(t) \\ \vdots \\ \nabla\boldsymbol{u}(\boldsymbol{X}(t),t)\,\hat{\boldsymbol{Q}}_N(t) \\ \boldsymbol{u}(\boldsymbol{X}(t),t) \end{bmatrix} - \frac{1}{4De}\boldsymbol{A} \begin{bmatrix} \boldsymbol{F}_1\left(\hat{\boldsymbol{Q}}_1(t)\right) \\ \vdots \\ \vdots \\ \boldsymbol{F}_N\left(\hat{\boldsymbol{Q}}_N(t)\right) \\ 0 \end{bmatrix} \right) \mathrm{d}t + \sigma \begin{bmatrix} \mathrm{d}\boldsymbol{W}_1(t) \\ \mathrm{d}\boldsymbol{W}_2(t) \\ \vdots \\ \vdots \\ \mathrm{d}\boldsymbol{W}_{N+1}(t) \end{bmatrix} \tag{1.34}$$

with

$$\sigma = \frac{1}{\sqrt{2De}} \begin{pmatrix} -1 & 1 & 0 & & \\ 0 & -1 & 1 & \ddots & \\ 0 & \ddots & \ddots & \ddots & 0 \\ & \ddots & \ddots & -1 & 1 \\ 0 & 0 & \dots & & 0 \end{pmatrix} \in \mathbb{R}^{(N+1)\times(N+1)} \tag{1.35}$$

such that Theorem 1 is applicable.

We then apply Theorem 1 to the ODE system (1.34). The resulting Fokker-Planck equation equivalent to (1.24) is then

$$\begin{aligned} &\frac{\partial\psi}{\partial t} + \nabla_{\boldsymbol{x}} \cdot (\boldsymbol{u}\psi) + \sum_{i=1}^{N} \nabla_{\boldsymbol{q}_i} \cdot \left((\nabla_{\boldsymbol{x}}\boldsymbol{u})^T \boldsymbol{q}_i \psi - \frac{1}{4\,De} \sum_{j=1}^{N} A_{ij}\boldsymbol{F}(\boldsymbol{q}_i) \right) \\ &= \frac{1}{4\,De} \sum_{i=1}^{N}\sum_{j=1}^{N} A_{ij} \nabla_{\boldsymbol{q}_i} \cdot \nabla_{\boldsymbol{q}_j}\psi \end{aligned} \tag{1.36}$$

with $\psi : D_1\times\cdots\times D_N\times O\times[0,T] \to \mathbb{R}^+$, $(\boldsymbol{q}_1, \dots, \boldsymbol{q}_N, \boldsymbol{x}, t) \mapsto \psi(\boldsymbol{q}_1, \dots, \boldsymbol{q}_N, \boldsymbol{x}, t)$ and $\psi(\boldsymbol{q}_1, \dots, \boldsymbol{q}_N, \boldsymbol{x}, 0) = \psi_0(\boldsymbol{q}_1, \dots, \boldsymbol{q}_N)$. Note that this equation does not contain the center of mass diffusion $\Delta_{\boldsymbol{x}}$ due to the simplification in (1.31). The center of mass diffusion is typically in the order of $O(10^{-8})$ since it scales quadratically with the ratio of the polymer length to the macroscopic length. This ratio is typically in the order of 10^{-4}. Barrett and Süli [5] however show that it is necessary to keep $\Delta_{\boldsymbol{x}}$ to be able to show the existence of global weak solutions for some dilute polymer models.

Brownian Configuration Fields

The random variable $\boldsymbol{X}(t)$ in (1.33) appears as a parameter in the stochastic ODE (1.32). It is therefore possible to rewrite the stochastic ODE for the stochastic process $\hat{\boldsymbol{Q}}(t)$ as a stochastic PDE for a random field $\boldsymbol{Q}(\boldsymbol{x}, t)$, $\boldsymbol{x} \in O \subset \mathbb{R}^3$. The description in form of a stochastic ODE is called *Lagrangian* perspective. An alternative representation as a random field is denoted as *Eulerian* perspective. The correspondence principle between these two equivalent descriptions is given in Table 1.1. Both descriptions can be

Table 1.1.: Comparison of Lagrangian and Eulerian description of the microscopic bead-spring-chain system

	Lagrangian	Eulerian
chain description	$N+1$ $3D$-valued stochastic processes $\hat{\boldsymbol{Q}}_1(t), \ldots, \hat{\boldsymbol{Q}}_N(t), \boldsymbol{X}(t)$	N time-dependent $3D$-valued random fields $\boldsymbol{Q}_1(\boldsymbol{x}, t), \ldots, \boldsymbol{Q}_N(\boldsymbol{x}, t)$
dynamics	N+1 $3D$-valued stochastic ODEs	N $3D$-valued stochastic PDEs
discrete approach	CONNFFESSIT	Brownian Configuration fields

converted into each other. For this purpose, we

- identify $\boldsymbol{Q}(\boldsymbol{X}(t), t) \leftrightarrow \hat{\boldsymbol{Q}}(t)$ with $\boldsymbol{X}(t)$ as solution of (1.33) and
- use the material derivative / Lagrangian derivative $\frac{\mathrm{d}}{\mathrm{d}t} := \frac{\partial}{\partial t} + \boldsymbol{u} \cdot \nabla$ to transform the Lagrangian equation of motion into the Eulerian description and vice versa. The material derivative allows to rewrite the evolution of $\hat{\boldsymbol{Q}}(t)$ as

$$\frac{\mathrm{d}\hat{\boldsymbol{Q}}(t)}{\mathrm{d}t} = \frac{\partial \boldsymbol{Q}(\boldsymbol{x}, t)}{\partial t} + (\boldsymbol{u}(\boldsymbol{x}, t) \cdot \nabla)\, \boldsymbol{Q}(\boldsymbol{x}, t). \tag{1.37}$$

The Eulerian reformulation of (1.32) and (1.33) leads to the stochastic partial differential equation

$$\begin{bmatrix} \mathrm{d}\boldsymbol{Q}_1(\boldsymbol{x},t) \\ \mathrm{d}\boldsymbol{Q}_2(\boldsymbol{x},t) \\ \vdots \\ \mathrm{d}\boldsymbol{Q}_N(\boldsymbol{x},t) \end{bmatrix} = \left(- \begin{bmatrix} (\boldsymbol{u}(\boldsymbol{x},t)\cdot\nabla)\boldsymbol{Q}_1(\boldsymbol{x},t) \\ (\boldsymbol{u}(\boldsymbol{x},t)\cdot\nabla)\boldsymbol{Q}_2(\boldsymbol{x},t) \\ \vdots \\ (\boldsymbol{u}(\boldsymbol{x},t)\cdot\nabla)\boldsymbol{Q}_N(\boldsymbol{x},t) \end{bmatrix} + \begin{bmatrix} \nabla\boldsymbol{u}(\boldsymbol{x},t)\cdot\boldsymbol{Q}_1(\boldsymbol{x},t) \\ \nabla\boldsymbol{u}(\boldsymbol{x},t)\cdot\boldsymbol{Q}_2(\boldsymbol{x},t) \\ \vdots \\ \nabla\boldsymbol{u}(\boldsymbol{x},t)\cdot\boldsymbol{Q}_N(\boldsymbol{x},t) \end{bmatrix} \right. \tag{1.38}$$

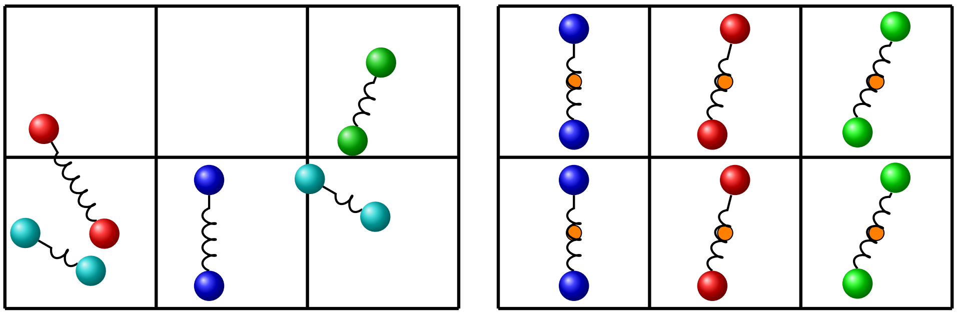
$$\left. - \frac{1}{4De}\boldsymbol{A} \begin{bmatrix} \boldsymbol{F}(\boldsymbol{Q}_1(\boldsymbol{x},t)) \\ \boldsymbol{F}(\boldsymbol{Q}_2(\boldsymbol{x},t)) \\ \vdots \\ \boldsymbol{F}(\boldsymbol{Q}_N(\boldsymbol{x},t)) \end{bmatrix} \right) \mathrm{d}t + \sqrt{\frac{1}{2De}} \begin{bmatrix} \mathrm{d}\boldsymbol{W}_2(t) - \mathrm{d}\boldsymbol{W}_1(t) \\ \mathrm{d}\boldsymbol{W}_3(t) - \mathrm{d}\boldsymbol{W}_2(t) \\ \vdots \\ \mathrm{d}\boldsymbol{W}_{N+1}(t) - \mathrm{d}\boldsymbol{W}_N(t) \end{bmatrix}$$

or in compact notation

$$\mathrm{d}\boldsymbol{Q}(\boldsymbol{x}, t) = \Big(- \boldsymbol{u}\nabla\boldsymbol{Q}(\boldsymbol{x}, t) + (\nabla\boldsymbol{u})^T\boldsymbol{Q}(\boldsymbol{x}, t) - \frac{1}{4\,De}\boldsymbol{A}\cdot\boldsymbol{F}(\boldsymbol{Q}(\boldsymbol{x}, t))\Big)\,\mathrm{d}t + \sigma\mathrm{d}\boldsymbol{W}(t) \tag{1.39}$$

for the random field $\boldsymbol{Q}(\boldsymbol{x}, t) = (\boldsymbol{Q}_1(\boldsymbol{x}, t), \ldots, \boldsymbol{Q}_N(\boldsymbol{x}, t))$. Again, $\boldsymbol{A}$ in (1.39) denotes the *Rouse matrix* (1.28) and $\boldsymbol{W}(t) = (\boldsymbol{W}_1(t), \boldsymbol{W}_2(t), \ldots, \boldsymbol{W}_{N+1}(t))$. Furthermore, $\sigma \in \mathbb{R}^{N\times(N+1)}$ is obtained from (1.35) when the last row is omitted and, for simplicity, we use the same notation as before.

One discrete approach for the solution of (1.39) is the sampling of the random field $\boldsymbol{Q}(\boldsymbol{x}, t)$ and the evolution of each particle according to (1.39). In the context of non-Newtonian fluid mechanics the



(a) CONNFFESSIT (b) BCF

Figure 1.3.: Schematic difference of the sampling method in a discrete approximation.

sample particles are called *Brownian Configuration fields* (BCF); see Hulsen, Van Heel and Van Den Brule [61]. The same sampling can be applied to the ODE system (1.32) and (1.33). As mentioned before, the approach is then denoted as CONNFFESSIT; see Laso and Öttinger [82]. We illustrate both approaches in Fig. 1.3.

1.2.3. Multiscale Navier-Stokes-BCF system

Up to now, the fluid's velocity field $\boldsymbol{u}(\boldsymbol{x}, t)$ in the stochastic PDE (1.38) and in the previous equations in Section 1.2 was assumed to be known. We now want to couple the equations for the spring-chain on the microscale with the macroscopic Navier-Stokes equations for the velocity field $\boldsymbol{u}(\boldsymbol{x}, t)$ and the pressure field $p(\boldsymbol{x}, t)$ as in Section 1.1. This requires a coupling between both length scales. The coupling from the macroscale equations to the microscale equations is given by the velocity field $\boldsymbol{u}(\boldsymbol{x}, t)$. The reverse coupling is given, up to a normalization constant, by the expectation of the Brownian Configuration field's tensor product. This expectation is the stress tensor field $\boldsymbol{\tau_p}(\boldsymbol{x}, t)$ that is already known from the purely macroscopic approach in Section 1.1. The calculation of the expectation is denoted as *Kramers' expression* in the context of computational rheology. We illustrate this coupling in Fig. 1.4.

The stress tensor

In the multiscale approach the macroscopic stress tensor $\boldsymbol{\tau_p} : (\boldsymbol{x}, t) \in \mathcal{O} \times \mathcal{T} \mapsto \boldsymbol{\tau_p}(\boldsymbol{x}, t) \in \mathbb{R}^{3\times 3}$ is also a random field $\boldsymbol{\tau_p} : (\omega, \boldsymbol{x}, t) \in \Omega \times \mathcal{O} \times \mathcal{T} \mapsto \boldsymbol{\tau_p}(\omega, \boldsymbol{x}, t) \in \mathbb{R}^{3\times 3}$ since it depends on the random field $\boldsymbol{Q}(\boldsymbol{x}, t)$. The stress tensor for a general spring chain is defined as

$$\boldsymbol{\tau_p}(\boldsymbol{x}, t) = \frac{3(1-\beta)\,\alpha_{b,d}}{De(N)\cdot((N+1)^2-1)} \sum_{i=1}^{N} \left(\int_D \boldsymbol{q}_i(\boldsymbol{x}, t) \otimes \boldsymbol{F}_i(\boldsymbol{q}_i(\boldsymbol{x}, t))\, \psi(\boldsymbol{q}, \boldsymbol{x}, t)\, \mathrm{d}\boldsymbol{q} - \mathbf{Id} \right) \tag{1.40}$$

$$\hat{=} \frac{3(1-\beta)\,\alpha_{b,d}}{De(N)\cdot((N+1)^2-1)} \sum_{i=1}^{N} \left(\mathbb{E}[\boldsymbol{Q}_i(\boldsymbol{x}, t) \otimes \boldsymbol{F}_i(\boldsymbol{Q}_i(\boldsymbol{x}, t))] - \mathbf{Id} \right) \tag{1.41}$$

where i again denotes the spring element of the spring chain, $\boldsymbol{F}_i : D_i \subset \mathbb{R}^3 \to \mathbb{R}^3$ is the spring force of Section 1.2.3, $\alpha_{b,d} \in \mathbb{R}$ is a given spring force dependent constant, $\otimes : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$ is the tensor product of two vector fields in $\mathbb{R}^3$ and $\mathbb{E}[\cdot] = \int_D \cdot\, \psi(\boldsymbol{q}, \boldsymbol{x}, t)\, \mathrm{d}\boldsymbol{q}$ is the expectation defined on the N-fold

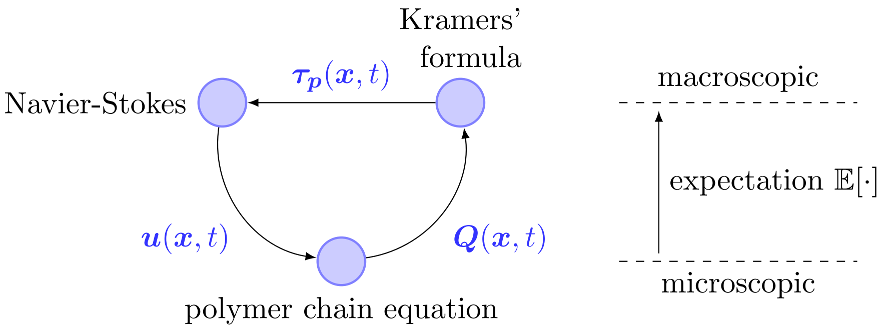


Figure 1.4.: Coupling of microscopic and macroscopic equations.

Cartesian product $D = D_1 \times \cdots \times D_N$ of balanced open sets D_i, $i = 1, \ldots, N$. In this application, *balanced* means that $\boldsymbol{q}_i \in D_i$ only if $-\boldsymbol{q}_i \in D_i$; see Barrett and Süli [4]. Moreover, $\psi(\boldsymbol{q}, \boldsymbol{x}, t) \in C^{2,1}(\mathbb{R}^{3(N+1)} \times \mathcal{T})$ is the probability density function from Theorem 1.1.

We note that the representation in (1.40) is used for the deterministic Fokker-Planck based approach that is derived from Theorem 1.1. In contrast to this, equation (1.41) is used in the case of the spring chain being modeled as a stochastic process as in our application. The relation between (1.40) and (1.41) is therefore that we can identify a spring configuration vector with realizations of a stochastic process.

Furthermore, (1.40) and (1.41) involve two dimensionless parameters $De(N)$ (N-spring chain Deborah number) and β (viscosity ratio) known from the macroscopic approach in (1.10). Here, we define the Deborah number as

$$De(N) = \frac{\lambda(N)\, U_c}{L_c} \tag{1.42}$$

since the relaxation time λ depends on the number of spring segments N. We will discuss this dependency in Section 1.2.3. However, we note that $De(1)$ is equivalent to De as defined in (1.10) and therefore only use the notation $De(N)$ if $N > 1$. Analogously to Section 1.1, the reference units L_c (characteristic length in macroscopic flow) and U_c (characteristic fluid velocity) are used for the nondimensionalization of (1.41).

Connector spring forces

In equations (1.39) and (1.41) the explicit choice of the spring force $\boldsymbol{F}_i : D_i \subset \mathbb{R}^3 \to \mathbb{R}^3$ has not been specified. There are four popular choices in the literature for the characterization of intermolecular forces $\boldsymbol{F}_i$, namely, the *Hookean spring force*, the *FENE spring force* (cf. Warner [134]) the *FENE-P spring force* (cf. Keunings [69]) and *Cohen's Padé approximation* (cf. Cohen [26]). They are defined as

$$\boldsymbol{F}_i(\boldsymbol{q}_i) = \boldsymbol{q}_i, \qquad \boldsymbol{q}_i \in \mathbb{R}^3 \qquad \text{(Hooke)}, \tag{1.43a}$$

$$\boldsymbol{F}_i(\boldsymbol{q}_i) = \frac{\boldsymbol{q}_i}{1 - \|\boldsymbol{q}_i\|^2 / b(N)}, \qquad \|\boldsymbol{q}_i\|^2 \leq b(N) \qquad \text{(FENE)}, \tag{1.43b}$$

$$\boldsymbol{F}_i(\boldsymbol{q}_i) = \frac{\boldsymbol{q}_i}{1 - \langle \boldsymbol{q}_i^2 \rangle / b(N)}, \qquad \langle \boldsymbol{q}_i^2 \rangle \leq b(N) \qquad \text{(FENE-P)}, \tag{1.43c}$$

$$\boldsymbol{F}_i(\boldsymbol{q}_i) = \frac{1 - \|\boldsymbol{q}_i\|^2/(3b(N))}{1 - \|\boldsymbol{q}_i\|^2/b(N)}\boldsymbol{q}_i, \qquad \|\boldsymbol{q}_i\|^2 \leq b(N) \qquad \text{(Cohen)} \tag{1.43d}$$

for a parameter $b(N) \in \mathbb{R}^+$ that depends on the number of spring segments. This parameter can be considered as a normalized maximum extension of each spring segment. Furthermore, the configuration spaces D_i depend on the choice of the spring force model. These spaces are

$$D_i = \begin{cases} \mathbb{R}^3 & \text{for the Hookean and the FENE-P model and} \\ B(0, b(N)^{1/2}) & \text{for the FENE model and the Cohen's Padé approximant.} \end{cases}$$

Here, $B(0, b(N)^{1/2})$ denotes an open ball with radius $b(N)^{1/2} \in \mathbb{R}^+$ centered at the origin.

The Hookean (1.43a), FENE (1.43b) and Cohen (1.43d) spring forces can alternatively be characterized as the gradient of a scalar spring potential U with

$$U(\boldsymbol{q}_i) = \frac{1}{2}\|\boldsymbol{q}_i\|^2 \qquad \text{(Hooke),} \tag{1.44a}$$

$$U(\boldsymbol{q}_i) = -\frac{b(N)}{2}\log\left(1 - \frac{\|\boldsymbol{q}_i\|^2}{b(N)}\right) \qquad \text{(FENE),} \tag{1.44b}$$

$$U(\boldsymbol{q}_i) = \frac{\|\boldsymbol{q}_i\|^2}{6} - \frac{b(N)}{3}\log\left(1 - \frac{\|\boldsymbol{q}_i\|^2}{b(N)}\right) \qquad \text{(Cohen).} \tag{1.44c}$$

This is not possible for the FENE-P spring force (1.43c) which is derived by averaging the denominator of the FENE spring force (1.43b), i.e. the FENE-P spring is a closure approximation.

Equation (1.41) contains a further constant $\alpha_{b,d}$ that also depends on the spring force. Depending on the spring force, the constant is

$$\alpha_{b,d} \equiv \begin{cases} 1, & \text{for a Hookean spring } (b(N) \to \infty), \\ \frac{b(N)+5}{b(N)}, & \text{for a 3-dimensional FENE spring,} \\ \frac{b(N)+3}{b(N)}, & \text{for a 3-dimensional FENE-P spring.} \end{cases} \tag{1.45}$$

The coefficient $\alpha_{b,d}$ is more complicated for Cohen's Padé approximant as it leads to hypergeometric functions of $b(N)$. We note that the spring force laws (1.43b)–(1.43d) are more or less accurate approximations of the *inverse Langevin force law*. This force law is defined by

$$\boldsymbol{F}_i(\boldsymbol{q}_i) = \frac{b(N)}{3}\,\mathrm{L}^{-1}\left(\frac{\|\boldsymbol{q}_i\|}{\sqrt{b(N)}}\frac{\boldsymbol{q}_i}{\|\boldsymbol{q}_i\|}\right) \tag{1.46}$$

with the Langevin function $\mathrm{L}(x) = \coth(x) - 1/x$, $x \in [0, \infty)$; see Barrett and Süli [2]. The Langevin function tends to 1 when its argument goes to infinity. For that reason, the inverse Langevin function L^{-1} with the argument as in (1.46) goes to infinity when $\boldsymbol{q}_i \to \sqrt{b(N)}$. This is the case for the FENE and for Cohen's spring force laws (1.43b) and (1.43d) but not for the FENE-P model (1.43c). This emphasizes the limitations of the FENE-P closure approximation.

Spring forces as in (1.43a)–(1.43d) can be subdivided into two different groups. For some simple spring force models there exists a closed form solution. Then, these models can be rewritten as a macroscopic model as considered in Section 1.1. This is the case for the Hookean spring force in (1.43a)

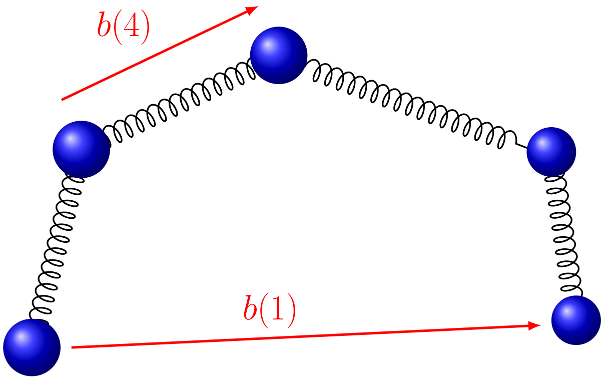


Figure 1.5.: Comparison of maximum single FENE spring extension in a four segment chain with the corresponding extension of a FENE dumbbell.

and for the FENE-P spring force in (1.43c). Since a macroscopic constitutive equation is much easier to solve than its multiscale equivalent, these two spring forces are only considered for comparative reasons. In fact, only those spring forces without a macroscopic equivalent are of practical importance. This is the case for the FENE spring force in (1.43b) and for Cohen's Padé approximation in (1.43d). In most applications in the literature, the FENE spring is the connector force of choice if a purely multiscale model is investigated.

Relaxation time and spring extension gauge

FENE models The stress tensor equation (1.41), (1.42) and furthermore the springe force equations (1.43b)–(1.43d) contain two dimensionless parameters $\lambda(N)$ (relaxation time of spring segment) and $b(N)$ (maximum spring extension of a single segment) depending on the number N of spring segments. In physical experiments, the material parameters from a real fluid can be estimated. This delivers information on the relaxation time of the full spring chain $\lambda \in \mathbb{R}^+$ and its maximum normalized extension $b \in \mathbb{R}^+$. There has to be, however, a gauge from the full chain to a single segment, i.e. we have to relate b with $b(N)$ and λ with $\lambda(N)$. A detailed explanation of the dumbbell's gauging is, for instance, given by Jin and Collins [64].

First, we consider the dumbbell case $N = 1$ for which we set

$$b(1) = b, \qquad \lambda(1) = \lambda \tag{1.47}$$

since the single dumbbell represents the full chain; cf. Fig. 1.5.

This becomes more complicated for the general case with $N > 1$ segments. For the maximum spring extension we follow an approach by de Gennes [33] and set

$$b(N) = \frac{b}{N}. \tag{1.48}$$

Thereby, the sum of all maximum elongated segments is the full spring length; cf. Fig. 1.5. This approach is used, for instance, by Koppol, Sureshkumar and Khomami [76] and by Jin and Collins [64].

There exist several gauges for the relaxation time λ and consequently for the Deborah number De.

An adequate gauge for λ is problem-dependent. Popular choices are

$$\lambda(N) = \lambda, \tag{1.49}$$

$$\begin{aligned}\lambda(N) &= \lambda/d_1 \text{ with } d_1 = \frac{b(N)}{b(N)+5}\cdot\frac{\sqrt{(b+5)(b+7)}}{b(N)} \qquad (1.50)\\ &\cdot\Big[\frac{(2(N+1)^2+7)\cdot((N+1)^2-1)}{45} - \frac{12((N+1)^4-1)}{45(N+1)(b(N)+7)}\Big]^{1/2},\end{aligned}$$

$$\begin{aligned}\lambda(N) &= \lambda/d_2 \text{ with } d_2 = \frac{b+7}{15b}\cdot\frac{b(N)}{b(N)+5} \qquad (1.51)\\ &\cdot\left[(2(N+1)^2+7) - \frac{12[(N+1)^2+1]}{(N+1)(b(N)+7)}\right].\end{aligned}$$

The first gauge for $\lambda(N)$ equals the relaxation time of the dumbbell with the relaxation time of the bead-spring model. The gauge in (1.50) is used to equate the zero shear material function $\psi_{10}^{N\text{-bead}}$ and $\psi_{10}^{\text{dumbbell}}$ such that

$$\begin{aligned}\psi_{10}^{N\text{-bead}} =& nk_BT\lambda(N)^2\cdot\frac{2b(N)^2}{45(b(N)+5)^2}\Bigg[((N+1)^2-1)(2(N+1)^2+7)\\ &-\frac{12\left[(N+1)^4-1\right]}{(N+1)(b(N)+7)}\Bigg]\\ \overset{!}{=}& nk_BT\lambda^2\cdot\frac{2b^2}{(b+5)(b+7)} = \psi_{10}^{\text{dumbbell}}\end{aligned} \tag{1.52}$$

according to equation (55) in Wiest and Tanner [139]. Equation (1.52) is solved for $\lambda(N)$ to obtain the second gauge.

The third relation (1.51) follows from matching the zero shear characteristic relaxation time scale $\psi_{10}/(2\eta_{p0})$ of the spring-chain to a dumbbell system which leads to

$$\begin{aligned}\left(\frac{\psi_{10}}{2\eta_p}\right)_{N\text{-bead}} =& \frac{b(N)\,\lambda(N)}{15(b(N)+5)}\left[(2(N+1)^2+7) - \frac{12\left[(N+1)^2+1\right]}{(N+1)(b(N)+7)}\right]\\ \overset{!}{=}& \lambda\cdot\frac{b}{b+7} = \left(\frac{\psi_{10}}{2\eta_p}\right)_{\text{dumbbell}}.\end{aligned} \tag{1.53}$$

Equation (1.53) is solved for $\lambda(N)$ to obtain the third gauge.

In this thesis, we employ the third mapping (1.53) for $\lambda(N)$ for which Koppol et al. [76] report a high agreement with experimental results in shear and extensional flows. Basically, $\lambda(N)$ decreases with increasing segment number N in (1.53) , i.e. each spring segment becomes stiffer. Fig. 1.6 shows the decrease of the time scale ratio $\lambda(N)/\lambda$ according to (1.53) for different maximum spring extensibility parameters $b = 20, 60, 120, 180$. The values of b have been chosen such that they coincide with the experimental parameters in Chapter 3. Interestingly, the larger the extensibility parameter b the faster the decrease of $\lambda(N)/\lambda$, i.e. long spring-chains segments require a more stiff adjustment than short ones.

Finally, we note that all scalings of $\lambda(N)$ and $b(N)$ represent choices to balance the characteristics of spring-chain models that differ in the number of segments N. Normally, these models cannot be

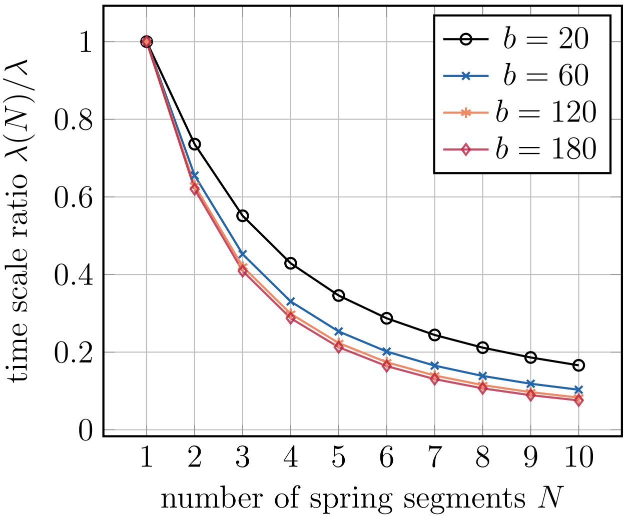


Figure 1.6.: Decrease of the spring segment time scale ratio according to the mapping in (1.53) for different spring extensibilities b.

mapped such that they predict an identical flow behavior. A more detailed analysis of this problem is given by Ghosh et al. [46].

Hookean springs - Rouse model In case the connector segments are represented by Hookean springs we obtain the Rouse model. The relation from the Hookean relaxation time λ to the more general Rouse spring chain $\lambda(N)$ depends on the gauge, i.e.

$$\lambda(N) = \lambda \tag{1.54a}$$

$$\lambda(N) = \lambda/d_1 \text{ with } d_1 = \frac{(2(N+1)^2+7)\cdot((N+1)^2-1)}{45} \tag{1.54b}$$

$$\lambda(N) = \lambda/d_2 \text{ with } d_2 = \frac{(2(N+1)^2+7)}{15}. \tag{1.54c}$$

The relations (1.54b) and (1.54c) can be obtained from (1.50) and (1.51) for the limit $b \to \infty$.

Initial condition for the probability density function

Depending on the chosen spring connector force (1.43a)–(1.43d) the initial condition $\psi_0(\boldsymbol{q}_1, \dots, \boldsymbol{q}_N)$ for the probability density function in (1.36) or, equivalently, the initial distribution of the random field $\boldsymbol{Q}(\boldsymbol{x}, 0) \sim \psi_0(\boldsymbol{q}_1, \dots, \boldsymbol{q}_N)$ in (1.39) is different.

We employ the equilibrium solution ψ_{eq} as initial condition for the Fokker-Planck or stochastic differential equation. ψ_{eq} is derived as the solution of the Fokker-Planck equation (1.36) for the velocity field $\boldsymbol{u} = 0$ in Ω. Then, for $\boldsymbol{q} = (\boldsymbol{q}_1, \dots, \boldsymbol{q}_N) \in \mathbb{R}^{3N}$ equation (1.36) simplifies to

$$-\nabla_{\boldsymbol{q}} \cdot \Big(\boldsymbol{F}(\boldsymbol{q})\psi_{eq}(\boldsymbol{q})\Big) = \Delta_{\boldsymbol{q}}\psi_{eq}(\boldsymbol{q}). \tag{1.55}$$

First, we state the equilibrium solutions for one spring segment $N = 1$. Following Bonvin [14] and the article by Herrchen and Öttinger [59], the equilibrium solutions are

$$\psi_{eq}(\boldsymbol{q}) = \frac{1}{(2\pi)^{3/2}} \exp\left(-\frac{\|\boldsymbol{q}\|^2}{2}\right) \qquad \text{(Hooke)}, \tag{1.56a}$$

$$\psi_{eq}(\boldsymbol{q}) = \frac{1}{2\pi b(N)^{3/2}\mathcal{B}(\frac{3}{2}, \frac{b(N)+2}{2})} \left(1 - \frac{\|\boldsymbol{q}\|^2}{b(N)}\right)^{b(N)/2} \qquad \text{(FENE)}, \tag{1.56b}$$

$$\psi_{eq}(\boldsymbol{q}) = \left(\frac{b(N)+3}{2\pi b(N)}\right)^{3/2} \exp\left(-\frac{b(N)+3}{2b(N)}\|\boldsymbol{q}\|^2\right) \qquad \text{(FENE-P)}, \tag{1.56c}$$

$$\psi_{eq}(\boldsymbol{q}) = C \exp\left(-\frac{\|\boldsymbol{q}\|^2}{6}\right)\left(1 - \frac{\|\boldsymbol{q}\|^2}{b(N)}\right)^{b(N)/3} \qquad \text{(Cohen)} \tag{1.56d}$$

with a normalization constant $C \in \mathbb{R}$ and the beta function $\mathcal{B}(.,.)$ that is defined by

$$\mathcal{B}(x, y) = \int_0^1 s^{x-1}(1-s)^{y-1}\,\mathrm{d}s \qquad \text{for } x, y > 0. \tag{1.57}$$

Except for the FENE-P model, the equilibrium solution is proportional to the negative exponential of the spring force potential U, i.e. $\psi_{eq} \propto \exp(-U)$ with U as in (1.44a)–(1.44c). Again, the FENE-P spring behaves differently since it is a closure approximation from the FENE spring to reduce the computational complexity.

Next, we consider the equilibrium solution for the general case with $N > 1$ spring segments. These are the generalizations of the single segment density functions. Since the equilibrium random fields for the single segments are independent from each other, the joint density is the product of the single densities. We then obtain the multi segment equilibrium solutions

$$\psi_{eq}(\boldsymbol{q}_1, \ldots, \boldsymbol{q}_N) = \frac{1}{(2\pi)^{(3N)/2}} \prod_{i=1}^{N} \exp\left(-\frac{\|\boldsymbol{q}_i\|^2}{2}\right) \qquad (N\text{-segment Rouse chain}), \tag{1.58a}$$

$$\psi_{eq}(\boldsymbol{q}_1, \ldots, \boldsymbol{q}_N) = \frac{1}{\left(2\pi b(N)^{3/2}\mathcal{B}(\frac{3}{2}, \frac{b(N)+2}{2})\right)^N} \prod_{i=1}^{N} \left(1 - \frac{\|\boldsymbol{q}_i\|^2}{b(N)}\right)^{b(N)/2} \qquad (N\text{-segment FENE chain}), \tag{1.58b}$$

$$\psi_{eq}(\boldsymbol{q}_1, \ldots, \boldsymbol{q}_N) = \left(\frac{b(N)+3}{2\pi b(N)}\right)^{(3N)/2} \prod_{i=1}^{N} \exp\left(-\frac{b(N)+3}{2b(N)}\|\boldsymbol{q}_i\|^2\right) \qquad (N\text{-segment FENE-P chain}), \tag{1.58c}$$

$$\psi_{eq}(\boldsymbol{q}_1, \ldots, \boldsymbol{q}_N) = C \prod_{i=1}^{N} \exp\left(-\frac{\|\boldsymbol{q}_i\|^2}{6}\right)\left(1 - \frac{\|\boldsymbol{q}_i\|^2}{b(N)}\right)^{b/3} \qquad (N\text{-segment Cohen chain}). \tag{1.58d}$$

We note that (1.58a) is a $3N$-dimensional normal distribution with zero mean vector and the identity as covariance matrix. Furthermore, (1.58c) is a $3N$-dimensional normal distribution with zero mean vector and covariance matrix $\frac{b(N)}{b(N)+3}\mathbf{Id}$.

Coupled multiscale flow system

The full multiscale Navier-Stokes-BCF system as illustrated in Fig. 1.4 consists of the equations for conservation of momentum and mass for an incompressible and isothermal viscoelastic one-phase flow (1.6) and (1.7) in addition to the BCF equation (1.39) for an N-bead-spring chain and Kramers' expression (1.41). We summarize the full problem for the FENE spring force (1.43b) since it will be primarily used for the applications in Chapter 3 and Chapter 6.

For given $\mathcal{T} = [0, t_{\max}]$ and given flow space $\mathcal{O} \subset \mathbb{R}^3$ find the solutions

- $\boldsymbol{u} : (\boldsymbol{x}, t) \in \mathcal{O} \times \mathcal{T} \mapsto \boldsymbol{u}(\boldsymbol{x}, t) \in \mathbb{R}^3$ (velocity field),
- $p : (\boldsymbol{x}, t) \in \mathcal{O} \times \mathcal{T} \mapsto p(\boldsymbol{x}, t) \in \mathbb{R}$ (pressure field)
- $\boldsymbol{\tau_p} : (\boldsymbol{x}, t) \in \mathcal{O} \times \mathcal{T} \mapsto \boldsymbol{\tau_p}(\boldsymbol{x}, t) \in \mathbb{R}^3 \times \mathbb{R}^3$ (stress tensor field)
- $\boldsymbol{Q} : (\boldsymbol{x}, t) \in \mathcal{O} \times \mathcal{T} \mapsto \boldsymbol{Q}(\boldsymbol{x}, t) = (\boldsymbol{Q}_1(\boldsymbol{x}, t), \dots, \boldsymbol{Q}_N(\boldsymbol{x}, t)) \in D_1 \times \cdots \times D_N \subset \mathbb{R}^{3N}$ (Brownian Configuration field)

such that

$$\frac{\partial \boldsymbol{u}(\boldsymbol{x},t)}{\partial t} + (\boldsymbol{u}(\boldsymbol{x},t)\cdot\nabla)\boldsymbol{u}(\boldsymbol{x},t) = -\nabla p(\boldsymbol{x},t) + \frac{1}{Re}\beta\Delta\boldsymbol{u}(\boldsymbol{x},t) + \frac{1}{Re}\nabla\cdot\boldsymbol{\tau}_p(\boldsymbol{x},t) \tag{1.59}$$

$$\nabla\cdot\boldsymbol{u}(\boldsymbol{x},t) = 0 \tag{1.60}$$

$$\begin{aligned} \mathrm{d}\boldsymbol{Q}(\boldsymbol{x},t) = \Big[&-\boldsymbol{u}(\boldsymbol{x},t)\nabla\boldsymbol{Q}(\boldsymbol{x},t) + (\nabla\boldsymbol{u}(\boldsymbol{x},t))^T\boldsymbol{Q}(\boldsymbol{x},t) \\ &- \frac{1}{4\,De(N)}\boldsymbol{A}\cdot\boldsymbol{F}(\boldsymbol{Q}(\boldsymbol{x},t))\Big]\mathrm{d}t + \sigma\mathrm{d}\boldsymbol{W}(t) \end{aligned} \tag{1.61}$$

$$\boldsymbol{\tau_p}(\boldsymbol{x},t) = \frac{3(1-\beta)\,(b(N)+5)}{b(N)\,De(N)\,((N+1)^2-1)}\sum_{i=1}^{N}\left(\mathbb{E}[\boldsymbol{Q}_i(\boldsymbol{x},t)\otimes\boldsymbol{F}_i(\boldsymbol{Q}_i(\boldsymbol{x},t))] - \mathbf{Id}\right) \tag{1.62}$$

$$\boldsymbol{F}_i(\boldsymbol{Q}_i) = \frac{\boldsymbol{Q}_i}{1-\|\boldsymbol{Q}_i\|^2/b(N)} \quad \text{for } i = 1,\dots,N \tag{1.63}$$

with the initial conditions

$$\boldsymbol{u}(\boldsymbol{x},0) = \boldsymbol{u}_0(\boldsymbol{x}), \tag{1.64}$$

$$p(\boldsymbol{x},0) = p_0(\boldsymbol{x}), \tag{1.65}$$

$$\boldsymbol{Q}_i(\boldsymbol{x},0) \sim \psi_{\mathrm{eq}}(\boldsymbol{q}) = \frac{1}{2\pi b(N)^{3/2}\mathcal{B}(\frac{3}{2},\frac{b(N)+2}{2})}\left(1-\frac{\|\boldsymbol{q}\|^2}{b(N)}\right)^{b(N)/2} \quad \text{for } i = 1,\dots,N \tag{1.66}$$

$$\boldsymbol{\tau}_p(\boldsymbol{x},0) = \boldsymbol{\tau}_0(\boldsymbol{x}) \quad \forall\boldsymbol{x}\in\mathcal{O}. \tag{1.67}$$

Furthermore, the Reynolds number Re and the viscosity ratio β are given in (1.10) and the maximum spring segment extension $b(N)$ as well as the modified Deborah number $De(N)$ are stated in (1.48) and (1.49)–(1.51).

The boundary conditions for $\boldsymbol{u}$ are the same as in the macroscopic formulation (1.3a)–(1.3c). As in the Newtonian case, homogeneous Neumann boundary conditions are assumed for the pressure field p in all applications. Furthermore, the restriction on $\boldsymbol{Q}$ is that all of its realizations are in the configuration space

$D_1 \times \cdots \times D_N$, i.e. $\|\boldsymbol{Q}_i(\boldsymbol{x}, t, \omega)\| \leq \sqrt{b} \quad \forall (\boldsymbol{x}, t, \omega) \in D_i \times \mathcal{T} \times \Omega$ and $i = 1, \ldots, N$. This corresponds to a zero Dirichlet boundary condition for the density function ψ that is associated with $\boldsymbol{Q}$ on the boundaries ∂D_i, $i = 1, \ldots, N$. Since the equilibrium configuration fields are independent from each other, we individually sample each spring segment i. We note, however, that the segments are coupled in (1.61) by the Rouse matrix $\boldsymbol{A}$ from (1.28) . Therefore, for the general case $t \neq 0$ the segments are not independent and the probability density functions ψ has no product form. Finally, we note that the random field $\boldsymbol{Q}$ also requires a boundary condition with respect to the position $\boldsymbol{x}$ in flow space O. Usually, these are Dirichlet boundary conditions on the inflow boundary and homogeneous Neumann boundary conditions otherwise. Their implementation into NaSt3DGPF will be discussed in Section 2.2.1.

1.2.4. Existence of global weak solutions for multiscale model

The existence and uniqueness of solutions for the three-dimensional Navier-Stokes equations (1.59) and (1.60) is still an area of active research; see Fefferman [38]. The existence and uniqueness of solutions for the two-dimensional Navier-Stokes equations in $\mathbb{R}^3$ has been shown by Ladyzhenskaya [80]. For the three-dimensional case only partial results have been shown:

- If the initial velocity $\boldsymbol{u}_0$ in (1.64) satisfies a smallness condition, smooth and globally defined solutions of the three-dimensional Navier-Stokes equations exist; see e.g. Koch and Tataru [74].

- For a general initial velocity such a solution exists only on a small interval $[0, T)$ in which T depends on $\boldsymbol{u}_0$. The maximum T in this context is called *blowup time*.

Furthermore, additional results on the existence of weak solutions of the Navier-Stokes equations have been shown. Leray [86] proved the existence of global weak solutions for a general initial velocity. It is however unknown if this solution is unique.

The mathematical analysis of the multiscale flow system (1.59)–(1.60) is even more complicated. Beside the nonlinear transport term $\boldsymbol{u} \cdot \nabla \boldsymbol{u}$ in (1.59), the coupling terms $\boldsymbol{u} \nabla \boldsymbol{Q}$ and $(\nabla \boldsymbol{u})^T \boldsymbol{Q}$ in (1.61) lead to additional difficulties in proving existence. A further nonlinearity comes from the FENE spring force (1.63) which also complicates finding existence results. For this reason, only the existence of globally weak solutions has been shown. In the following, we summarize the most important findings. These are primarily given for the the Fokker-Planck description of the system with the density function ψ so that (1.61) is replaced with (1.36) in the coupled multiscale system.

Entropy estimates for the multiscale system As indicated by Le Bris and Lelièvre [84] most results on the existence of weak solutions base on a priori estimate of the relative entropy. The relative entropy is also known as Kullback-Leibler divergence and gives a non-symmetric measure of the difference between two probability density functions. In our application, the relative entropy of $\psi(t)$ and $t \in \mathcal{T}$ with respect to a steady state solution ψ_{steady} is defined as

$$D_{KL}(\psi(t) \| \psi_{\text{steady}}) = \int_{D_1 \times \ldots D_N} \psi(t) \log \left(\frac{\psi(t)}{\psi_{\text{steady}}} \right) d\boldsymbol{q}. \tag{1.68}$$

The entropy $D_{KL}(\psi(t) \| \psi_{\text{steady}})$ is always non-negative. Furthermore, it is zero only if $\psi(t) = \psi_{\text{steady}}$ almost everywhere according to Gibb's inequality. Using $D_{KL}(\psi(t) \| \psi_{\text{steady}})$, we can estimate the L^1-

norm of $\psi(t) - \psi_{\text{steady}}$ as

$$\int_{D_1 \times \ldots D_N} |\psi(t) - \psi_{\text{steady}}| \, d\boldsymbol{q} \;\leq\; \sqrt{2 D_{KL}(\psi || \psi_{\text{steady}})} \tag{1.69}$$

according to the Csiszár-Kullback inequality [32]. Consequently, if we can find a steady state solution ψ_{steady} so that $D_{KL}(\psi(t)||\psi_{\text{steady}}) \to 0$ for $t \to \infty$ we obtain convergence in L^1 from (1.69). Crucial for the proof $D_{KL}(\psi(t)||\psi_{\text{steady}}) \to 0$ for $t \to \infty$ is the existence of a logarithmic Sobolev inequality. There is a logarithmic Sobolev inequality for ψ_{steady} if there exists a constant $C > 0$ for which

$$\int_{D_1 \times \ldots D_N} \phi \log\left(\frac{\phi}{\psi_{\text{steady}}}\right) d\boldsymbol{q} \leq \frac{1}{2C} \int_{D_1 \times \ldots D_N} |\nabla \log\left(\frac{\phi}{\psi_{\text{steady}}}\right)|^2 \, d\boldsymbol{q} \tag{1.70}$$

for all probability density functions ϕ defined on $D_1 \times \ldots D_N$. The existence of a logarithmic Sobolev inequality (1.70) and a steady state solution is difficult to show in practice. We summarize some of the results in the literature.

Existence results for Fokker-Planck equation in simple flows

Various results have been shown for the case in which the Fokker-Planck equation (1.36), or equivalently the stochastic PDE (1.61), is not coupled with the fluid equations (1.59) and (1.60) with (1.62); see Jourdain et al. [66]. In that case, the density function ψ does not depend on the spatial variable $\boldsymbol{x}$. We only consider the simplified situation in which the velocity gradient does not depend on space and on time, i.e. $\nabla \boldsymbol{u}(\boldsymbol{x}, t) = \kappa \in \mathbb{R}^{3\times 3}$. Depending on the form of κ, different steady state solutions ψ_{steady} are known:

- If $\kappa = 0$ as in (1.55) or if κ is skew-symmetric, i.e. $\kappa^T = -\kappa$, then the unique steady state solution is

  $$\psi_{\text{steady}} = C \, \exp(-U). \tag{1.71}$$

  Here, $C \in \mathbb{R}$ is a spring force dependent constant and U is the spring potential as in (1.44a)–(1.44c). Note that in applications this is the initial condition that we employ for the Fokker-Planck equation (1.36).

- If κ is symmetric, then the unique steady state solution is

  $$\psi_{\text{steady}} = C \, \exp\left(-U + De \, \boldsymbol{q}^T \kappa \boldsymbol{q}\right) \tag{1.72}$$

  for the FENE spring force (1.43b) and for Cohen's spring force (1.43d). For the Hookean spring force (1.43a), the stated solution exists only for the case that the eigenvalues of κ are strictly smaller than $\frac{1}{2De}$.

  The most important application in which κ is symmetric is an extensional flow. In an extensional flow κ is a diagonal matrix with trace$(\kappa) = 0$ due to the continuity equation (1.60). We here concentrate on the special case for an extensional flow in which $\kappa = \text{diag}(\dot{\epsilon}, -\dot{\epsilon}/2, \dot{\epsilon}/2)$ with extension rate $\dot{\epsilon} \in \mathbb{R}$. As reported, for instance, by Owens and Phillips [100] the Hookean dumbbell spring / Oldroyd-B model predicts unlimited normal stress for extensional flows $\dot{\epsilon} \geq \frac{1}{2De}$. Consequently, these models have further restrictions on the eigenvalues of κ.

In both cases, ψ convergences exponentially to ψ_{steady} since

$$D_{KL}(\psi(t)||\psi_{\text{steady}}) \leq D_{KL}(\psi(0)||\psi_{\text{steady}}) \exp(-\frac{C}{De}t) = D_{KL}(\psi_{eq}||\psi_{\text{steady}}) \exp(-\frac{C}{De}t) \tag{1.73}$$

for all $t \in \mathcal{T}$ and $C > 0$ as the constant in (1.70). This result follows from the logarithmic Sobolev inequality (1.70), Gronwall's Lemma and from the decrease of $D_{KL}(\psi(t)||\psi_{\text{steady}})$ over time where ψ_{steady} is given by (1.71) or (1.72).

Existence results for multiscale modesl

We consider the general case for the coupled problem (1.59), (1.60), (1.36) and (1.41). Additionally to the results stated for the Navier-Stokes equations in the beginning of Section 1.2.4, the momentum equation (1.59) now contains the polymeric stress tensor $\boldsymbol{\tau}_p$. The stress tensor is primarily evolved by the velocity gradient $\nabla \boldsymbol{u}$. Consequently, the regularity of $\boldsymbol{\tau}_p$ resembles the regularity of $\nabla \boldsymbol{u}$.

The mathematical analysis of the multiscale model started with a local existence and uniqueness theorem by Renardy [115]. Renardy's proof bases on an iteration scheme to construct the solution. The scheme alternates between the macroscopic flow equations and the Fokker-Planck equation. Moreover, Le Bris and Lelièvre [84] report that global existence results have been primarily obtained for modified/ regularized formulations of the original problem:

- Barrett and Süli show the existence of global weak solutions for FENE type models in a series of articles. First in [5], the authors proof the existence of global solutions for the FENE dumbbell model $N = 1$ with center of mass diffusion, i.e. with an additional term $\Delta_{\boldsymbol{x}}\psi$ in (1.36), and with a microscopic cut-off function that restricts ψ by a certain value $L \in \mathbb{R}^+$. Later on this restrictions are weakened. In [6] and in [3] the cut-off function is removed by the proof of the limit $L \to \infty$. Furthermore, [6] considers the more general case of FENE spring chains with $N > 1$ segments and [3] focuses on the Rouse model. Recently in [7], the proof of global weak solutions has been extended to the case of polymers with variable density and viscosity.

  All of these existence results contain a center of mass diffusion term $\Delta_{\boldsymbol{x}}\psi$ in (1.31) that we skipped in the derivation. This additional term is justified by the physics of the system. Masmoudi [93], however, notes that this additional center of mass diffusion term simplifies the mathematical problem. Consequently, the results are not fully transferable to our applications.

- A larger class of global existence proofs uses a corotational derivative instead of the upper convected derivative (1.11). More precisely, the velocity gradient $\boldsymbol{\nabla u}$ in (1.36) is replaced by $\frac{\boldsymbol{\nabla u} - \boldsymbol{\nabla u}^T}{2}$. For this derivative, global existence results of weak solutions are given by Lions and Masmoudi [87]. Furthermore, the existence of strong solutions in 2D is shown by Masmoudi [92].

- An existence result that is applicable to our formulation of the multiscale FENE model has been given by Masmoudi [93]. Here, the author also omits the center of mass diffusion term and employs the classical upper convected derivative. The author proves the existence of global weak solutions for the FENE dumbbell system ($N = 1$ segments). An extension of this result to general FENE chains has not yet been given.

As mentioned before, the annotated existence results are given for the multiscale model in the Fokker-Planck formulation from equation (1.36). Much less results exists for the stochastic formulation that we employ in (1.59)–(1.62). Up to now, the most advanced result in the stochastic formulation is given by Jourdain, Lelièvre and Le Bris [67]. The authors show the existence of a local in time solution for the FENE dumbbell system (N = 1 segments).

2. Numerical modeling of polymeric fluids

This chapter focuses on discretization methods for the system of equations (1.59) – (1.63) that describes viscoelastic polymers in a Newtonian solvent. Due to the complex coupling of several nonlinear equations, the numerical techniques that are used for the microscopic stochastic partial differential equation in (1.61) differ from the techniques for the macroscopic three-dimensional Navier-Stokes equations in (1.59) and (1.60).

A classical method to calculate the first moment of a stochastic process is given by Monte Carlo (MC) quadrature. Our application in polymer physics requires high-dimensional integration in a configuration space $D \subset \mathbb{R}^{3N}$ with N as the number of polymer segments. All concepts that are related to the integration problem are discussed in Section 2.1.

The discretization of the macroscopic Navier-Stokes equations is considered in Section 2.2. We give a brief overview of the discretization in the physical flow space $O \subset \mathbb{R}^3$ but refer to Griebel and Rüttgers [51, 52] and Rüttgers [117] for a more detailed description. Furthermore, Section 2.2 explains the coupling between the macroscopic and the microscopic equations and gives detailed information on the temporal evolution of the coupled system.

Finally, Section 2.3 analyzes the computational complexity of our full grid approach and describes an efficient parallelization approach that bases on domain decomposition.

2.1. Monte Carlo quadrature for diffusion problems

This section is concerned with the high-dimensional integration problem which determines the elastic stress tensor. First, Section 2.1.1 gives general information on the accuracy of Monte Carlo quadrature. Then, in Section 2.1.2 we concentrate on the generation of random numbers for the random field that describes the initial polymer configurations. Moreover, we state an explicit algorithm which allows to generate high-dimensional initial samples. Next, in Section 2.1.3 we explain variance reduction schemes that will be used in the final multiscale algorithm. At last, in Section 2.1.4 we discuss higher-order Quasi Monte Carlo (QMC) methods for polymer physics. We also show that the accuracy of QMC reduces to the classical Monte Carlo accuracy if the dimensionality of the integration problem exceeds a certain limit.

2.1.1. General concepts

We first recall that the SPDE (1.61) for an N-segment bead-spring chain and Kramers' expression (1.62) can be written as a high-dimensional integration problem of the form

$$I_d(\boldsymbol{x}, t; f) = \int_D f(\boldsymbol{q})\, \psi(\boldsymbol{q}, \boldsymbol{x}, t)\, \mathrm{d}\boldsymbol{q} \tag{2.1}$$

with $D \subset \mathbb{R}^{3N}$ for the polymeric stress tensor $\boldsymbol{\tau}_p(\boldsymbol{x}, t)$; see equation (1.40) for the precise form of the integrand f. Even if the probability density function $\psi : D \times \mathcal{O} \times [0, T] \to \mathbb{R}^+$, $(\boldsymbol{q}, \boldsymbol{x}, t) \mapsto \psi(\boldsymbol{q}, \boldsymbol{x}, t)$ in (2.1) was exactly known, an approximation of (2.1) with *product rules* as

$$I_d(\boldsymbol{x}, t; f) \approx \sum_{i_1=1}^{m} \dots \sum_{i_{3N}=1}^{m} \omega_{i_1} \cdot \ldots \cdot \omega_{i_{3N}} f(q_{i_1}, \ldots, q_{i_{3N}})\, \psi(q_{i_1}, \ldots, q_{i_{3N}}, \boldsymbol{x}, t) \cdot \mathrm{Vol}(D) \tag{2.2}$$

with m grid points in each coordinate direction would suffer from the so called *curse of dimensionality*; see Caflisch [22] and Dick, Kuo and Sloan [35]. Equation (2.2) is the $3N$-fold tensor product of a one-dimensional quadrature rule with quadrature weights ω_i. The total number of points is $M_s = m^{3N}$ which grows exponentially in the number of spring segments N or in the problem dimension $d = 3N$, respectively.

Furthermore, the same problem occurs for grid based approaches of (2.2) with respect to the convergence behavior. Classical quadrature formulas of order k with M_s points in the d-dimensional hypercube have a $M_s^{-1/d}$ grid spacing. Consequently, the convergence rate of the d-fold product rule is of order $O(M_s^{-k/d})$.

The Monte Carlo (MC) approach which we will use in the following consists of an equal-weight cubature rule

$$I_d(\boldsymbol{x}_f, t_f; f) \approx Q_{M_s,d}(\boldsymbol{x}_f, t_f; f) = \frac{1}{M_s} \sum_{i=1}^{M_s} f(\boldsymbol{t}_i) \tag{2.3}$$

with M_s samples $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{M_s} \in D$ of dimension d that are all i.i.d. (independent and identically distributed) drawn from the probability density function $\psi(\boldsymbol{q}, \boldsymbol{x}_f, t_f) : D \to \mathbb{R}^+$ for fixed $\boldsymbol{x}_f \in \mathcal{O}$ and $t_f \in \mathcal{T} = [0, T]$. Alternatively, but worse from a numerical perspective, (2.3) can also be rewritten as

$$I_d(\boldsymbol{x}_f, t_f; f) \approx \tilde{Q}_{M_s,d}(\boldsymbol{x}_f, t_f; f) = \frac{1}{M_s} \sum_{i=1}^{M_s} f(\boldsymbol{t}_i)\, \psi(\boldsymbol{t}_i, \boldsymbol{x}_f, t_f) \cdot \mathrm{Vol}(D) \tag{2.4}$$

with M_s i.i.d uniformly drawn samples $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{M_s} \in D$. In (2.3), the random samples are generated according to the measure $\psi d\boldsymbol{q}$. On the other hand, the probability measure in (2.4) is just $d\boldsymbol{q}$. The second approach in (2.4) is worse than (2.3) since a lot of samples are spent in regions in which ψ is close to zero, i.e. in regions that do not have a large contribution to the integral. In the literature, ψ in (2.3) is often denoted as an *importance sampling function* and the approach is known as *importance sampling*; see Srinivasan [125].

The main advantages of a Monte Carlo approach is that

- it does not require the integrand to be smooth but only to be square-integrable and that
- the order of convergence does not depend on the dimension d. However, the constant of proportionality C is still dimension-dependent.

In the following, we investigate the accuracy and the convergence behavior of the Monte Carlo method. As a short-hand notation, we subsequently write $Q_{M_s,d}(f)$ instead of $Q_{M_s,d}(\boldsymbol{x}, t; f)$ for the sake of simplicity. The convergence of $Q_{M_s,d}(f)$ in (2.3) to the real integral $I_d(f)$ in probability according to

$$\mathrm{Prob}\left(\lim_{M_s \to \infty} Q_{M_s,d}(f) = I_d(f) \right) = 1$$

is guaranteed by the *strong law of large numbers*; see e.g. Feller [39] for a proof. Furthermore, the approximation of $I_d(f)$ by $Q_{M_s,d}(f)$ is unbiased and therefore

$$\mathbb{E}[Q_{M_s,d}(f)] = I_d(f)$$

for all $M_s \in \mathbb{N}$. In this case, the expectation $\mathbb{E}[Q_{M_s,d}(f)]$ is understood with respect to the random samples $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{M_s}$ such that $\mathbb{E}[Q_{M_s,d}(f)] = \int_D \cdots \int_D \frac{1}{M_s} \sum_i f(\boldsymbol{t}_i)\psi(\boldsymbol{t}_i) \mathrm{d}\boldsymbol{t}_1 \ldots \mathrm{d}\boldsymbol{t}_{M_s}$.

Next, we define the Monte Carlo integration error as

$$\epsilon_{M_s}(f) = |I_d(f) - Q_{M_s,d}(f)|.$$

Then, the root-mean-square error (RMSE) is $\mathbb{E}[\epsilon_{M_s}(f)^2]^{1/2}$. The RMSE can be estimated by the following theorem.

Theorem 2.1 [Monte Carlo root-mean-square error]
Let f be square-integrable, then the RMSE of $Q_{M_s,d}(f)$ is

$$\mathbb{E}[\epsilon_{M_s}(f)^2]^{1/2} = \frac{\sigma(f)}{M_s^{1/2}} \tag{2.5}$$

with

$$\sigma(f)^2 = I_d(f^2) - I_d(f)^2 \tag{2.6}$$

as the variance of f.

Proof: A proof can be found, for instance, in Dick, Kuo and Sloan [35].

As a result, the RMSE of Monte Carlo integration is of order $O(M_s^{-1/2})$ with the variance of the integrand f as a dimension-dependent constant. A comparison of grid-based methods of order $O(M_s^{-k/d})$ with the Monte Carlo RMSE of order $O(M_s^{-1/2})$ suggests that Monte Carlo approaches are the method of choice for problem dimensions d with $k/d \leq 1/2$.

Practical error estimation In practice, the RMSE (2.5) can only be estimated since the exact knowledge of $\sigma(f)$ requires the exact knowledge of the cubature problem $I_d(f)$. Instead of that, a relation between $\sigma(f)^2$, the variance of f, and $\mathrm{Var}[Q_{M_s,d}(f)]$, the variance of the MC approximation, is used. They are related via

$$\mathrm{Var}[Q_{M_s,d}(f)] = \mathbb{E}[|I_d(f) - Q_{M_s,d}(f)|^2] = \frac{\sigma(f)^2}{M_s}. \tag{2.7}$$

The MC approach provides an unbiased estimator for $\mathrm{Var}[Q_{M_s,d}(f)]$ according to

$$\mathrm{Var}[Q_{M_s,d}(f)] \approx \frac{1}{M_s - 1} \sum_{i=1}^{M_s} \left(f(\boldsymbol{t}_i) - Q_{M_s,d}(f)\right)^2. \tag{2.8}$$

In combination with (2.7) this results in an estimator for $\mathbb{E}[\epsilon_{M_s}(f)^2]^{1/2}$ in (2.5).

Central Limit Theorem The statistical distribution of the Monte Carlo integration error is described by the Central Limit Theorem. It states that the error is normally distributed regardless of the underlying sample point distribution.

Theorem 2.2 [Central Limit Theorem]
If $0 < \sigma(f) < \infty$ *then*

$$\lim_{M_s\to\infty} \text{Prob}\left(I_d(f) - Q_{M_s,d}(f) \leq z\frac{\sigma(f)}{\sqrt{M_s}}\right) = \frac{1}{2\pi}\int_{-z}^{z} \exp(-x^2/2)\,\mathrm{d}x \tag{2.9}$$

for any $z \in \mathbb{R}$.

Proof: see e.g. Theorem 30.13 in Bauer [8].

Required number of samplings The Central Limit Theorem allows to determine the required number of samples M_s in (2.3) to achieve a given accuracy $\epsilon_{M_s}(f)$ up to a given confidence interval c. The confidence interval is just the right-hand side of equation (2.9) according to

$$c = \frac{1}{2\pi}\int_{-s(c)}^{s(c)} \exp(-x^2/2)\,\mathrm{d}x \tag{2.10}$$

with a real-valued confidence function $s : [0,1] \to \mathbb{R}$ that has to be determined at c. The required number of samples M_s is then

$$M_s = \epsilon_{M_s}(f)^{-2}\sigma(f)^2 s(c) \tag{2.11}$$

with confidence level c.

2.1.2. Random number generation

In the following, we focus on Monte Carlo integration with a non-uniform measure $\psi d\boldsymbol{q}$ as in (2.3) for the case $t = 0$. For our multiscale model in polymer physics, we require random numbers $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{M_s} \sim \psi(0)$ where $\psi(0)$ is the initial condition of the Fokker-Planck equation (1.36). Depending on the specific elastic spring force, the initial condition is given in (1.56a)–(1.58d). The FENE spring force, in which we are primarily interested in the applications, leads to a complex probability density function from which random samples cannot be generated easily, for instance, from an existing library. In the following, we shortly discuss our method of choice, the rejection sampling, and then state an algorithm to generate samples for an N-segment FENE chain (1.58b).

For the purpose of sample generation, two different approaches are primarily used:

Inversion method Let X be a real random variable with continuous cumulative distribution function $F_X(x) = P(X \leq x) : \mathbb{R} \to [0,1]$. Furthermore, let U be a uniform random variable on $[0,1]$. Then, $F_X^{-1}(U)$ follows the distribution F_X.

von Neumann rejection sampling This is a Monte Carlo type approach that was invented by von Neumann [131]. The basic idea is that samples for a random variable can be generated if its density function is known and the area that lies under the graph of the density function is uniformly sampled.

In practice, it can be complicated to construct F_X^{-1} in the inversion method. Since we want to compute samples that are distributed according to the bead-spring chain equilibrium density functions $\psi(0)$ in (1.36), we employ von Neumann's rejection sampling. The rejection sampling only requires that the

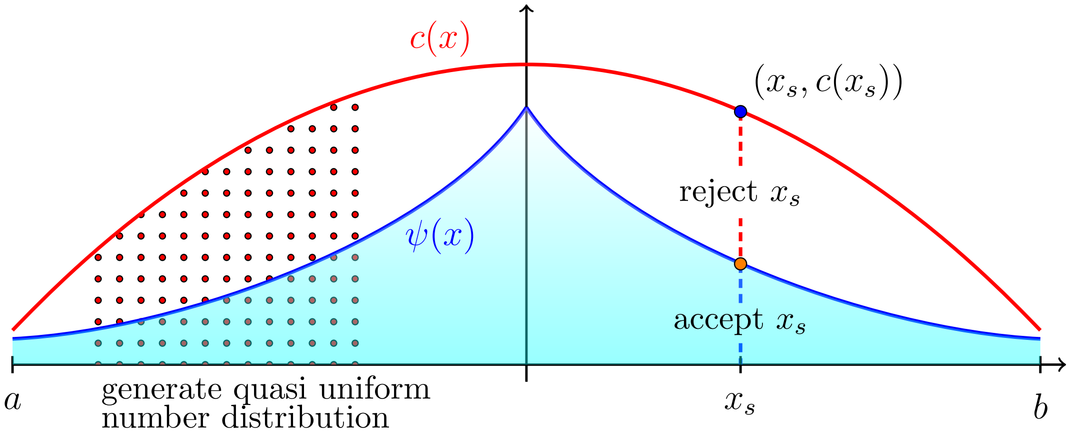


Figure 2.1.: The von Neumann rejection sampling generates random numbers distributed according to a known density $\psi(x)$ by using samples from a comparison function $c(x)$.

density function is known up to a constant. But, on the downside, this approach generates unwanted overhead samples.

The main idea of the rejection method bases on a geometric argument that we will discuss in one dimension for the simplified case $\psi : [a, b] \subset \mathbb{R} \to \mathbb{R}$. Fig. 2.1 shows the graph of the probability density function ψ in blue. If the area below the graph $(x, \psi(x))$ is sampled uniformly with a sequence of points $\{(x_1, \psi(x_1)), (x_2, \psi(x_2)), \ldots\}$, then the sequence of x-coordinates $\{x_s\}$, $s \in S \subsetneq \mathbb{N}$ is distributed according to ψ.

Von Neumann [131] proposed a method how to construct such an uniform sampling below the graph $(x, \psi(x))$. For this purpose, a comparison function $c : [a, b] \subset \mathbb{R} \to \mathbb{R}$ is required. The comparison function has to fulfill two requirements. These requirements are that

- $\psi(x) \leq c(x)$ for all $x \in [a, b]$ and
- a method is known which generates samples $x_1, x_2, \ldots$ distributed according to $c(x)$.

These samples $\{(x_s, y_s)\}_{s \in S \subsetneq \mathbb{N}}$ then fill the area below the graph $(x, c(x))$ uniformly if an imaginative y-coordinate is added; see Fig. 2.1. The rejection sampling method

- accepts the subset $\{(x_{\tilde{s}}, y_{\tilde{s}})\} \subset \{(x_s, y_s)\}_{s \in S}$ of points which also lie below the graph $(x, \psi(x))$ and
- rejects the remaining points.

As result, $\{x_{\tilde{s}}\}_{\tilde{s} \in S}$ is distributed according to $\psi(x)$. Algorithm 1 implements the von Neumann rejection sampling for an N-segment FENE chain (1.58b). The outcome of the algorithm cannot be easily visualized since the samples are $3N$-dimensional. For this reason, we show in Fig. 2.2 the sample points distributed according to a simplified $2D$ probability density function; cf. Rüttgers [117]. The density function is a special case of the FENE chain for a single segment, the dumbbell case, and a two-dimensional configuration space.

Algorithm 1 incorporates several implementation details. First, we use the fact that $\psi(\boldsymbol{q}_1, \ldots, \boldsymbol{q}_N, \boldsymbol{x})$ is a product density with respect to $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_N$. For the Monte Carlo approach this allows the simplified

Algorithm 1: The von Neumann rejection sampling algorithm applied to the initial density of an N-segment FENE chain with maximum spring segment extensibility b.

```
Data: $N$-segment FENE chain density function $\psi(\boldsymbol{q}_1, \dots, \boldsymbol{q}_N)$ as in (1.58b);
comparison function $c(\boldsymbol{q}_i) = 3/(4\pi b^{3/2})$ for $i = 1, \dots, N$;
uniform random number generator $U(0, 1)$;
Result: sequence of random numbers for $\psi(\boldsymbol{q}_1, \dots, \boldsymbol{q}_N)$      // $\psi = \psi_1(\boldsymbol{q}_1) \cdot \ldots \cdot \psi_N(\boldsymbol{q}_N)$
$L \leftarrow 2/(3 \cdot \mathcal{B}(\frac{3}{2}, \frac{b+2}{2}))$  // find a constant $L > 1$ with $\psi \leq L \cdot c$ pointwise, the beta
  function $\mathcal{B}(.,.)$ is defined in (1.57);
$s \leftarrow 1$ // sample number;
$i \leftarrow 1$ // spring segment number;
for $i \leftarrow 1$ to $N$ ;                                  // loop over all $N$ segments
do
    while $s \leq M_s$ do
        Sample random number $\boldsymbol{x}_i^{(s)} \sim c(\boldsymbol{q}_i)$  // restrict $\boldsymbol{x}_i^{(s)} \in U(-\sqrt{b}, \sqrt{b})^3$ on sphere;
        Generate random number $u \in U(0, 1)$;
        if $(uLc(\boldsymbol{x}_i^{(s)})) \leq \psi(\boldsymbol{x}_i^{(s)})$ then
            Accept $\boldsymbol{x}_i^{(s)}$ as an sample of $\psi(\boldsymbol{q}_i)$;
            $s \leftarrow s + 1$;
        else
            Reject the value $\boldsymbol{x}_i^{(s)}$;
        end
    end
end
Obtain $s = 1, \dots, M_s$ samples $\boldsymbol{t}^{(s)} = (\boldsymbol{x}_1^{(s)}, \dots, \boldsymbol{x}_N^{(s)}) \sim \psi(\boldsymbol{q}_1, \dots, \boldsymbol{q}_N)$.
```

computation of (2.1) for all $\boldsymbol{x} \in O$ as

$$
\begin{aligned}
&\int_D f(\boldsymbol{q}_i)\psi(\boldsymbol{q}_1, \dots, \boldsymbol{q}_N, \boldsymbol{x}) \,\mathrm{d}\boldsymbol{q} \\
&= \int_D f(\boldsymbol{q}_i)\psi(\boldsymbol{q}_1, \boldsymbol{x}) \cdot \ldots \cdot \psi(\boldsymbol{q}_N, \boldsymbol{x}) \,\mathrm{d}\boldsymbol{q} \\
&= \int_{D \setminus D_i} \left(\int_{D_i} f(\boldsymbol{q}_i)\psi(\boldsymbol{q}_i, \boldsymbol{x}) \,\mathrm{d}\boldsymbol{q}_i \right) \psi(\boldsymbol{q}_1, \boldsymbol{x}) \cdot \ldots \cdot \psi(\boldsymbol{q}_{i-1}, \boldsymbol{x}) \cdot \psi(\boldsymbol{q}_{i+1}, \boldsymbol{x}) \cdot \ldots \cdot \psi(\boldsymbol{q}_N, \boldsymbol{x}) \,\mathrm{d}\boldsymbol{q}_{1,\dots,i-1,i+1,\dots,N}
\end{aligned}
$$

for $i = 1, \dots, N$. Consequently, we obtain a sequence of N three-dimensional random fields which can be computed separately. For this reason, Algorithm 1 scales linearly in the number of spring segments N.

The previously stated requirement $\psi(x) \leq c(x)$ for all $x \in [a, b]$ cannot be fulfilled by any probability density function. More precisely, since both probability density functions necessarily satisfy $\int \psi \, dx = \int c \, dx = 1$ and if $\psi \neq c$ then there exists an $x_s \in [a, b]$ with $\psi(x_s) > c(x_s)$. As a solution, Algorithm 1 contains a real constant $L > 1$ with $\psi(x) < Lc(x)$ for all x. An optimal choice of L is essential for the efficiency of the algorithm. The larger the constant L in Algorithm 1 the more samples from $c(x)$ are

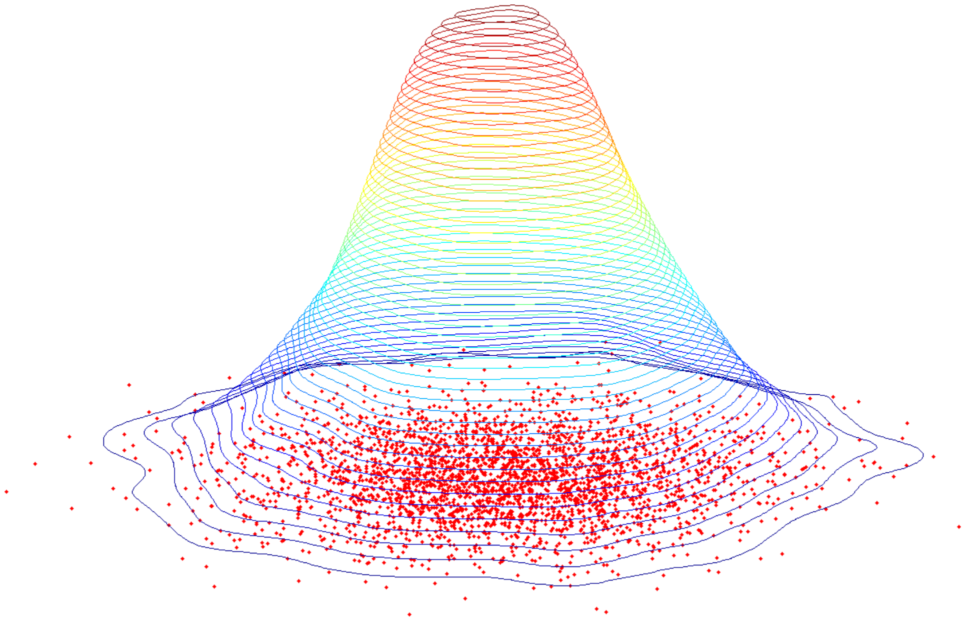


Figure 2.2.: A sequence of 3000 two-dimensional random numbers which is distributed according to the probability density function of a $2D$ FENE dumbbell model.

rejected as shown in Fig. 2.1. For this reason, the optimal choice L_{opt} for L is

$$L_{\text{opt}} = \max\left\{ x \in [a,b] \,\middle|\, \frac{\psi(x)}{c(x)} > 1 \right\}. \tag{2.12}$$

The number of rejected sampling points is minimized if L is chosen according to (2.12).

In a more precise mathematical notation, von Neumann's rejection sampling for generating samples of dimensionality D can be rewritten as integration problem of dimension $D+1$. This transforms the integration problem (2.1) into the integral

$$\int_D \int_0^1 f(\boldsymbol{q})\,\psi(\boldsymbol{q},\boldsymbol{x},t)\,\chi\!\left(y < \frac{1}{L_{\text{opt}} \cdot c(\boldsymbol{q})}\right) \mathrm{d}y\,\mathrm{d}\boldsymbol{q} \tag{2.13}$$

with characteristic function χ.

For the specific case of an N-segment FENE chain as in Algorithm 1 we employ a constant density $c(\boldsymbol{q}) = 3/(4\pi b^{3/2})$ as comparison function for the N segment equilibrium density $\psi_{eq}(\boldsymbol{q}_1,\ldots,\boldsymbol{q}_N)$. Then, the optimal choice L_{opt} is

$$L_{\text{opt}}^{\text{FENE}} = \frac{\psi_{eq}(0,\ldots,0)}{c(0)} = \frac{2}{3 \cdot \mathcal{B}(\frac{3}{2}, \frac{b+2}{2})}$$

with $\mathcal{B}(x,y)$ as the Eulerian Beta function that is defined as

$$\mathcal{B}(x,y) = \int_0^1 s^{x-1}(1-s)^{y-1}\mathrm{d}s \qquad \text{for } x,y > 0.$$

We note that a similar algorithm was proposed by Bonvin [14, 15] to generate samples for the FENE equilibrium density. In this case, the algorithm is restricted to the dumbbell case $N = 1$. The author explicitly uses spherical coordinates to take advantage of the radial symmetry of the solution. However, the algorithm by Bonvin still produces a large number of unwanted samples as it also bases on the rejection sampling approach.

Note that Section 2.1.1 and Section 2.1.2 considered Monte Carlo integration $I_d(\boldsymbol{x}_f, t_f; f)$ for the case $t_f = 0$ only. For $t_f > 0$, the probability density function ψ is not known in general. One solution is to evolve ψ according to the Fokker-Planck equation (1.36). Instead of that, as mentioned in Section 1.2.3, we evolve the samples $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{M_s} \in \mathcal{D}$ from (2.3) in time such that $\boldsymbol{t}_1(t), \ldots, \boldsymbol{t}_{M_s}(t) \sim \psi(t)$ for all $t \in \mathcal{T} = [0, t_{\max}]$. The algorithm for the temporal evolution of the stochastic realizations will be given in Section 2.2.2 where we directly consider the full coupled system with the Navier-Stokes equations.

2.1.3. Variance reduction schemes

Variance reduction schemes are used to decrease the statistical error of a Monte Carlo approach without increasing the computational effort, i.e. without increasing the number of samples M_s. In the literature, various methods for variance reduction have been proposed. As far as we know, only two of these approaches have been applied to the problem of polymer dynamics:

Importance sampling The basic idea of importance sampling is to determine those samples in (2.3) that have the largest influence on $I_d(\boldsymbol{x}_f, t_f; f)$. For instance, since the density functions of the bead-spring chains (1.58a)–(1.58d) are radial symmetric with their maximum at the origin, most random numbers are generated close to the origin. The integrand, however, is zero there and only has a minor effect on the result. Importance sampling gives larger importance to the samples with a more significant influence on the expectation. An application to polymer modes was first given by Melchior and Öttinger [94, 95]. We further note that the MC treatment according to (2.3) instead of (2.4) can already be considered as an importance sampling approach. In this case, we denote ψ in (2.3) as *importance function*.

Control variates Let $\boldsymbol{X}_t$ be a stochastic process for which $\mathbb{E}[\boldsymbol{X}_t]$ has to be determined. Instead of directly calculating $\mathbb{E}[\boldsymbol{X}_t]$, we decompose the expectation according to

$$\mathbb{E}[\boldsymbol{X}_t] = \mathbb{E}[\boldsymbol{X}_t - \boldsymbol{Y}_t] + \mathbb{E}[\boldsymbol{Y}_t] \tag{2.14}$$

for a further process $\boldsymbol{Y}_t$, the control variate. The basic idea of a control variate is to construct $\boldsymbol{Y}_t$ such that for all $t \in \mathcal{T} = [0, t_{\max}]$

- $\mathbb{E}[\boldsymbol{Y}_t]$ can be deterministically computed, i.e. $\mathrm{Var}[\boldsymbol{Y}_t] = 0$, and
- $\mathrm{Var}[\boldsymbol{X}_t - \boldsymbol{Y}_t] < \mathrm{Var}[\boldsymbol{X}_t]$.

A more detailed explanation of this approach can be found in Bonvin [14].

In the following, we focus on the method of control variates. This is due to the fact that for a specific choice of the control variate $\boldsymbol{Y}_t$ the method reduces the variance of the expectation without any significant computational overhead. For this purpose, we first give details on an optimal choice for $\boldsymbol{Y}_t$. Since $\mathrm{Var}[\boldsymbol{X}_t - \boldsymbol{Y}_t] = \mathrm{Var}[\boldsymbol{X}_t] + \mathrm{Var}[\boldsymbol{Y}_t] - 2\,\mathrm{Cov}(\boldsymbol{X}_t, \boldsymbol{Y}_t)$, the condition $\mathrm{Var}[\boldsymbol{X}_t] > \mathrm{Var}[\boldsymbol{X}_t - \boldsymbol{Y}_t]$ implies

$$\mathrm{Cov}(\boldsymbol{X}_t, \boldsymbol{Y}_t) > \frac{1}{2}\,\mathrm{Var}[\boldsymbol{Y}_t]. \tag{2.15}$$

As a result, the variance on the right-hand side of (2.14) is only reduced in those cases in which $\boldsymbol{X}_t$ and $\boldsymbol{Y}_t$ are strongly correlated.

The stochastic process in our case is the time-dependent random field $\boldsymbol{Q}(\boldsymbol{x}, t)$, cf. Vanmarcke [127], that evolves according to

$$\mathrm{d}\boldsymbol{Q}(\boldsymbol{x}, t) = \Big[-\boldsymbol{u}(\boldsymbol{x}, t)\nabla\boldsymbol{Q}(\boldsymbol{x}, t) + (\nabla\boldsymbol{u}(\boldsymbol{x}, t))^T\boldsymbol{Q}(\boldsymbol{x}, t) - \frac{1}{4\,De(1)}\boldsymbol{A}\cdot\boldsymbol{F}\left(\boldsymbol{Q}(\boldsymbol{x}, t)\right)\Big]\,\mathrm{d}t + \sigma\mathrm{d}\boldsymbol{W}(t) \tag{2.16}$$

as specified in equation (1.61). For simplicity, we first consider $\boldsymbol{Q}(\boldsymbol{x}, t)$ for a fixed position $\boldsymbol{x} = \boldsymbol{x}_f \in \mathcal{O}$ such that $\boldsymbol{X}_t = \boldsymbol{Q}(\boldsymbol{x}_f, t)$ is a stochastic process. We then search for a control variate $\boldsymbol{Y}_t$ that closely resembles $\boldsymbol{X}_t$. Two possible choices for $\boldsymbol{Y}_t$ are

- an equilibrium control variate and
- a macroscopic model control variate.

The key idea in both cases is that the same Wiener process $\boldsymbol{W}(t)$ as in (2.16) for $\boldsymbol{X}_t$ is used for the control variate. This will ensure a strong coupling between $\boldsymbol{X}_t$ and its control variate.

Equilibrium control variate An *equilibrium control variate* $\boldsymbol{Y}_t$ reuses the initial configuration $\boldsymbol{Y}_0 = \boldsymbol{X}_0$ and evolves $\boldsymbol{Y}_t$ according to a similar equation as (2.16) but with the simplifications

- $\boldsymbol{u}(t) = 0$ for all $t \in \mathcal{T}$ and
- $\boldsymbol{F}(\boldsymbol{Y}_t) = \boldsymbol{F}(\boldsymbol{X}_t)$ for all $t \in \mathcal{T}$, i.e. we reuse the original spring force.

The SPDE (2.16) then simplifies to

$$\mathrm{d}\boldsymbol{Y}_t = -\frac{1}{4\,De(1)}\boldsymbol{A}\cdot\boldsymbol{F}\left(\boldsymbol{Y}_t\right)\,\mathrm{d}t + \sigma\mathrm{d}\boldsymbol{W}(t). \tag{2.17}$$

Equation (2.17), however, is just the stochastic equivalent of the equilibrium equation (1.55) for which the solution is given in (1.58a)–(1.58d). Consequently, we determine $\mathbb{E}[\boldsymbol{Y}_t] = 0$ for all $t \in \mathcal{T}$ deterministically and approximate $\mathbb{E}[\boldsymbol{X}_t - \boldsymbol{Y}_t]$ numerically.

Fig. 2.3 gives an example for an equilibrium control variate that is used to determine the τ_{xx}-stress component of $\boldsymbol{\tau}_p$ for a homogeneous extensional flow. In this example, $\mathbb{E}[\boldsymbol{X}_t]$ is colored in blue, $\mathbb{E}[\boldsymbol{Y}_t]$ (numerical approximation of the analytical result zero) is colored in brown and the better approximation by the control variate $\mathbb{E}[\boldsymbol{X}_t - \boldsymbol{Y}_t]$ is colored in red. The steady state solution at $t \geq 3$ is $\tau_{xx} \approx 4.02$ for the considered flow field. We indicate the steady state solution with a dashed blue line and intentionally use only $M_s = 100$ samples or configurations fields to emphasize the improvement of the control variate (red solid line).

Macroscopic model control variate We have mentioned in Section 1.2.3 that some microscopic spring forces can be used to derive a noise free macroscopic equation for $\boldsymbol{\tau_p}$. Interestingly, the comparison between such a microscopic spring model with its macroscopic counterpart can be used as control variate for a purely microscopic model. This will be explained in the following.

Again, we use the initial configuration $\boldsymbol{Y}_0 = \boldsymbol{X}_0$ as initial configuration for $\boldsymbol{Y}_t$. Moreover, we evolve the control variate with

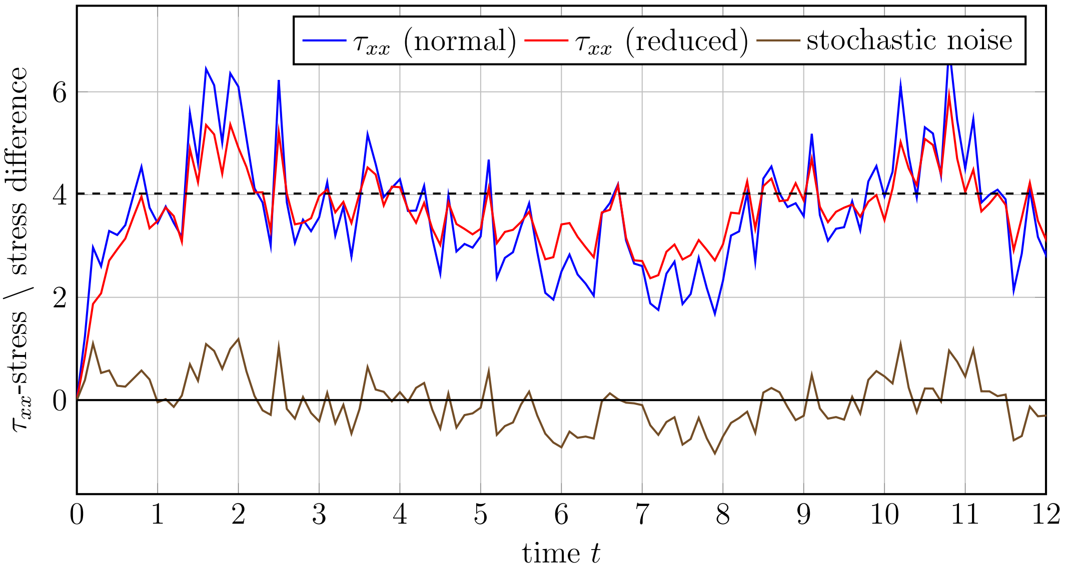


Figure 2.3.: Plot of the τ_{xx}-stress component over time for $M_s = 100$ samples. The normal simulation outcome (blue) is compared with the outcome of an equilibrium control variate (red). Moreover, the dashed line indicates the analytic steady state result.

- the same velocity field $\boldsymbol{u}(t)$ as for $\boldsymbol{X}_t$ in (2.16) and
- employ a spring force $\boldsymbol{F}$ which possesses a macroscopic equivalent.

For instance, if we want to use a Hookean control variate for the FENE spring-chain model (1.43b) we employ a Hookean spring force model (1.43a) for $\boldsymbol{Y}_t$. Then, $\mathbb{E}[\boldsymbol{X}_t - \boldsymbol{Y}_t]$ is calculated by Monte Carlo integration according to (2.3) and $\mathbb{E}[\boldsymbol{Y}_t]$ is the noise free solution from the macroscopic Oldroyd-B model in Section 1.1. This is due to the fact that the macroscopic Oldroyd-B model and the microscopic Hookean dumbbell system are formally equivalent. In situations in which the FENE and the Hookean model predict similar stress tensor values, the stochastic noise decreases strongly. On the other hand, the situation can become worse in situations in which both models differ, for instance, in extensional flows. Here, the Hookean spring model predicts infinite stress values which the FENE model does not.

As a conclusion, it strongly depends on the similarity of the purely microscopic model with the control variate if this approach improves the result. For this reason, a macroscopic model control variate has to be adapted to a specific flow problem.

2.1.4. Higher-order Quasi Monte Carlo methods

Quasi Monte Carlo (QMC) is an approach in which the random samples $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{M_s}$ for numerical integration in (2.3) and in (2.4) are replaced with a deterministic problem-adapted point set. A recent discussion of different QMC sequences can be found, for instance, in the article by Dick, Kuo and Sloan [35].

In the best case, the rate of convergence increases from $O(M_s^{-1/2})$ for MC points to $O(M_s^{-1})$ for QMC point sequences. Therefore, an obvious question is why do we still use a classical Monte Carlo

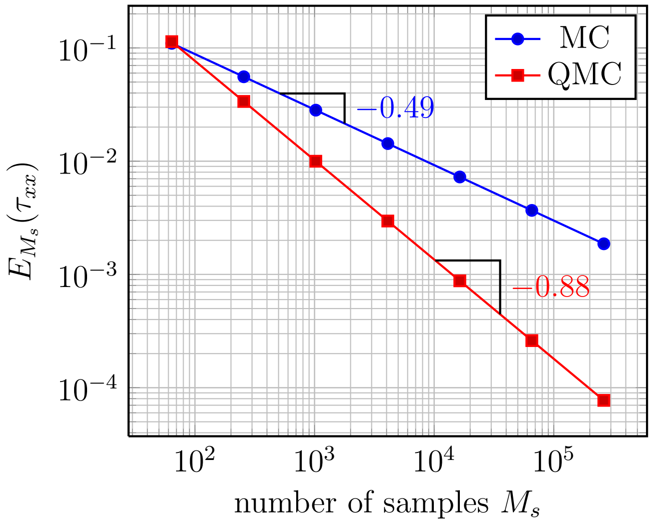


Figure 2.4.: Sampling of initial stress component τ_{xx} with MC and QMC sequences.

approach? In the following, we explain the difficulties of QMC point sets for diffusion problems as in our specific application which often reduces the rate of convergence to $O(M_s^{-1/2})$ as before.

Integration of the equilibrium density

The probability density function ψ in the high-dimensional integrand is exactly known only at the initial state at $t = 0.0$; see equilibrium solution ψ_{eq} in (1.56a)–(1.58d). In the following, we discuss a simple approach to generate a pseudo random sequence for ψ_{eq}. For this purpose, we modify Algorithm 1 by choosing $\boldsymbol{x}_i^{(s)} \in U(-\sqrt{b}, \sqrt{b})^3$ in line 7 from a QMC sequence. The rejection sampling algorithm then keeps the points $\boldsymbol{x}_i^{(s)} \sim \psi_{eq}$. It is a common approach to only select a subsequence out of a set of QMC points and skip the rest. For this reason, rejection sampling should not strongly decrease the efficiency of the QMC sequence as it can be considered as a skipping of points. On the other hand, rejection sampling can be considered as sampling of a non smooth integral as indicated in (2.13). Some articles in the literature, for instance Wang [132] or Nguyen and Ökten [97], observe that a smoothing of the characteristic function in (2.13) is required to obtain a rate of convergence as expected from QMC sequences.

In the following, the advantage of QMC points for a 2D FENE dumbbell model is considered since the corresponding sampling points can be visualized easily. In this case, the integral $I_d(\boldsymbol{x}, t = 0; \boldsymbol{q} \otimes \boldsymbol{F}(\boldsymbol{q}))$ is defined in a two-dimensional configuration space with a two-dimensional probability density function ψ_{eq}. We approximate the absolute error $E_{M_s}(\tau_{xx}) = |\tau_{xx,\text{numeric}}(t = 0)|$ of the first stress tensor component for M_s samples by averaging the outcome of ten independent simulations. The averaging of the stress tensor component is necessary since its convergence is only in probability. We also note that the analytic solution is $\tau_{xx,\text{analytic}}(t = 0) = 0$. All MC simulations are independent by construction. For the QMC simulations we first create a two-dimensional Sobol sequence $\boldsymbol{x}_i$ for $i = 1, \ldots, M_s$. Then, we add a random offset U according to

$$\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i + \boldsymbol{U} \pmod 1$$

with $U \sim U([0, 1]^2)$ to $\boldsymbol{x}_i$ and obtain independent point sets $\tilde{\boldsymbol{x}}_i$. Fig. 2.4 shows $E_{M_s}(\tau_{xx})$ over the number of samples for a 2D FENE dumbbell model with spring extensibility $b = 10$. As expected, a sampling with Monte Carlo points converges with a rate of about $1/2$. The rate of convergence increases with Quasi Monte Carlo points. In this case, we obtain an order of about 0.88 which is close to the theoretical order of 1.0. One explanation for the difference might be given by the unsmoothed characteristic function in (2.13). Nguyen and Ökten [97] give detailed information on smoothed characteristic functions that might be able to restore the theoretical rate of convergence of 1.0.

Integration of a time-dependent integral

The convergence study in Fig. 2.4 shows the improvement of QMC sequences over MC points in a situation when the probability density function ψ is known. As an illustrative example, the first row in Fig. 2.5 visualizes a typical distribution of 2000 samples from a MC point set on the left hand side and from a QMC point set on the right-hand side. The lower discrepancy of the QMC points is directly visible due to the more symmetric point arrangement. Furthermore, QMC samples do not build clusters as in the case of MC points.

The situation differs for $t > 0$ in which ψ in (1.62) is not known. In the literature, there exist different schemes for the temporal evolution of random variables and their discrete realizations as an ensemble of M_s samples. Section 2.2.2 gives detailed information on different stochastic time discretization schemes. An important difference is the treatment of the Brownian motion. Two important approaches that deal with Brownian motion are

- a random walk discretization and
- a Brownian bridge discretization.

In this section, we compare both approaches for the mathematical treatment of Brownian motion and evolve the initial distribution of 2000 MC and QMC points for a given velocity field in time. As mentioned before, the initial distribution is shown in the first row of Fig. 2.5. The evolution in time is according to a homogeneous extensional flow in x-direction i.e. the velocity field is $\boldsymbol{u}(x, y, z) = (\varepsilon\, x, -\varepsilon/2\, y, -\varepsilon/2\, z)$ with $\epsilon \in \mathbb{R}$ for all $\boldsymbol{x} = (x, y, z) \in \mathcal{O}$. For MC points, both discretizations, the random walk and Brownian bridge, use normally distributed MC points to approximate the Wiener process. The bottom left partial image of Fig. 2.5 shows a typical sample distribution at time $t = 5.12$ close to the steady state. Both approaches for Brownian motion lead to interchangeable results if MC points are used.

The situation differs for the temporal evolution using QMC points. In this case, the normally distributed points that approximate the Wiener process are also created from a quasi random sequence. The dimensionality of the problem is the product of the number of Brownian motions (i.e. the number of three-dimensional spring segments) with the number of time-steps in the temporal discretization. In this example, we evolve a two-dimensional FENE dumbbell system up to $t = 5.12$ with a time-step width $\Delta t = 0.01$. Then, the full dimension of the Brownian motion is $d = 1024$. Consequently, we have to construct, for instance, Sobol points of dimensionality $d = 1024$ and use the inverse of the Gaussian cumulative distribution function to obtain normally distributed quasi random samples. However, for a discretization with

- a random walk the order of convergence is worse than for MC samples (below 0.5) and with
- a Brownian bridge discretization the order of convergence is roughly the same as in the MC case.

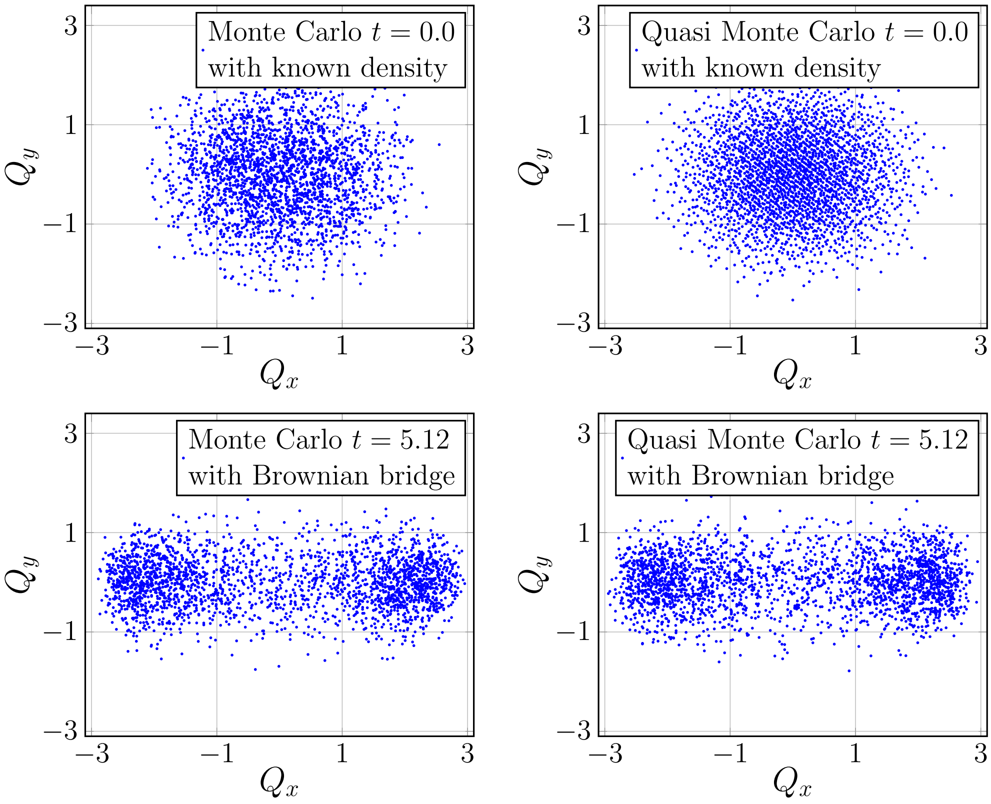


Figure 2.5.: Comparison of MC and QMC sampling of the initial FENE density (top row) and at $t = 5.12$ (bottom row) for a homogeneous extensional flow. The QMC approach at $t = 5.12$ has no advantage over the classical MC approach.

We visualize the Brownian bridge outcome with QMC points in the bottom right partial image of Fig. 2.5. Obviously, the low discrepancy of the point set at initial time has been lost at $t = 5.12$ due to the Brownian motion which leads to an overlap of the sampling points. In the end, the approximation of the QMC points in the final stage of the simulation is of similar accuracy as in the Monte Carlo case. For this reason, the order of convergence for the QMC points reduces from 0.92 as in Fig. 2.4 to roughly $1/2$. Of course, this reduction of accuracy depends on the problem and on the number of involved time-steps.

In Fig. 2.6 we give a descriptive illustration of the difficulties that occur in the Brownian path. Even though Sobol points are a low discrepancy point set, their distribution is inadequate in high dimensions. This leads to a pattern in the Gaussian point distribution on the right-hand side of Fig. 2.6. Consequently, this has an effect on the construction of the Brownian path. A random walk discretization gives equal importance to each dimension. On the other hand, a Brownian bridge discretization has a high variance in the first dimensions which then decreases in every step. For this reason, a Brownian bridge discretization can cope with the poor distribution of the 1024-dimensional point set whereas the random walk cannot. This is a first explanation for the observation in Fig. 2.5 for QMC point sets.

The problem of generating accurate point set with low discrepancy in high dimensions is still an active area of research; see e.g. Dick, Kuo and Sloan [35]. It is often the case that the MC order of $1/2$ can be improved up to an order that is between 0.5 and 1.0. Höök, Johnsen and Hellsten [60], for

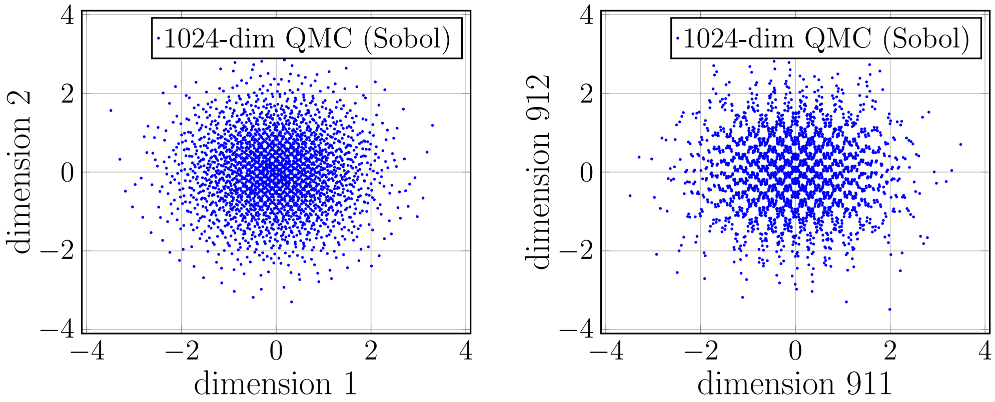


Figure 2.6.: Visualization of the dimensions 1, 2, 911 and 912 of a 1024-dimensional Sobol point set. The samples have been transformed with the inverse of the Gaussian cumulative distribution function.

instance, consider a problem of fast ion thermalization. In this case, the order of convergence directly correlates with the number of time-steps. For 200 time-steps the order of convergence is about 0.7 for the random walk and about 0.95 for the Brownian bridge. This decreases to about 0.3 for the random walk and 0.6 for the Brownian bridge if 1000 time-steps are used. As a result, the authors recommend a QMC approach only for moderate problem dimensions in the order $O(100)$. For problems of higher dimensionality that result from e.g. a large number of time steps, Monte Carlo approaches are still the methods of choice. Similar results are observed by Venkiteswaran and Junk [129]. These authors employ QMC points in the context of polymeric fluids but only for homogeneous flow fields in which the velocity field $\boldsymbol{u}$ does not depend on space $\boldsymbol{x}$ or time t.

2.2. Discretization of the multiscale Navier-Stokes-BCF system

This section covers the discretization of the coupled multiscale flow system from (1.59)–(1.63) in space and in time. Here, the spatial discretization refers to the physical space $O \subset \mathbb{R}^3$. The numerical treatment of the high-dimensional configuration space $D \subset \mathbb{R}^{3N}$ with N as the number of spring segments is considered in Section 2.1. The main result of this section is Algorithm 2 that allows to evolve the discrete microscopic and macroscopic unknowns over time.

2.2.1. Spatial discretization

A finite difference scheme is used for the spatial discretization of (1.59)–(1.63) with respect to $\boldsymbol{x} \in O$. For this purpose, we subdivide the physical domain O into rectangular grid cells. Then, the variables of interest are either discretized at the cell centers or on the cell faces. We place the macroscopic variables p and $\boldsymbol{\tau}_p$ in the cell centers and the three components of the velocity field $\boldsymbol{u}$ on the cell faces. This method is denoted as a staggered grid and ensures a strong coupling between p and $\boldsymbol{u}$. Otherwise, numerical instabilities, such as the checkerboard phenomena for the pressure field, might occur. The checkerboard phenomena describes a decoupling of adjacent discrete pressure values that then oscillate between two different values. Since the stress tensor results from a high-dimensional integral in configuration space D, see equation (1.62), it does make sense to also place the discrete

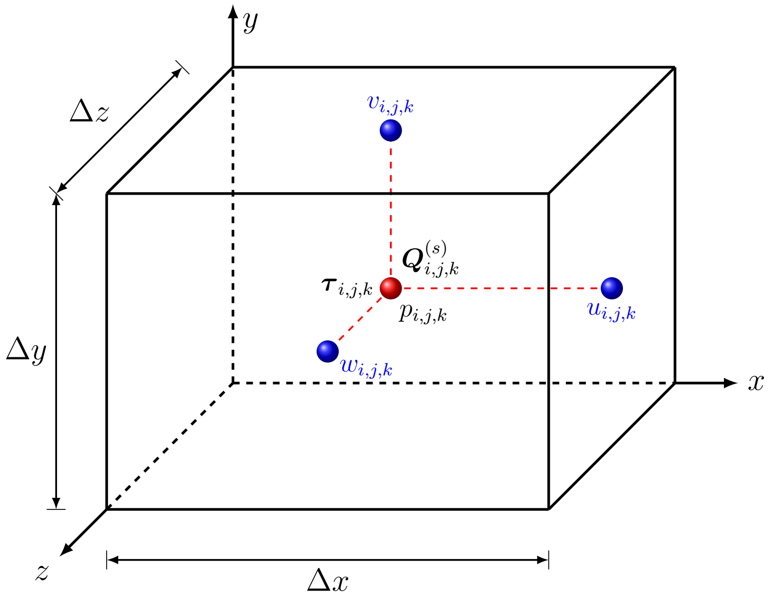


Figure 2.7.: Discrete unknowns at index position (i, j, k) on the staggered grid.

samples $\boldsymbol{Q}^{(s)}$, $s = 1, \dots, M_s$ of the $\boldsymbol{x}$-dependent random fields $\boldsymbol{Q}$ in the cell centers.

Let Δx, Δy and Δz denote the mesh widths in the x-, y- and z-direction, respectively. Then, the triple (i, j, k) with $i = 1, \dots, i_{\max}$, $j = 1, \dots, j_{\max}$ and $k = 1, \dots, k_{\max}$ fully determines the physical position of a discrete solution:

$$\begin{aligned}
u_{i,j,k} &= u(i\,\Delta x, (j-0.5)\Delta y, (k-0.5)\Delta z),\\
v_{i,j,k} &= v((i-0.5)\,\Delta x, j\,\Delta y, (k-0.5)\Delta z),\\
w_{i,j,k} &= w((i-0.5)\,\Delta x, (j-0.5)\Delta y, k\,\Delta z),\\
p_{i,j,k} &= p((i-0.5)\,\Delta x, (j-0.5)\Delta y, (k-0.5)\Delta z),\\
\boldsymbol{\tau}_{p,i,j,k} &= \boldsymbol{\tau}_p((i-0.5)\Delta x, (j-0.5)\Delta y, (k-0.5)\Delta z).
\end{aligned}$$

We visualize the position of the unknowns in grid cell (i, j, k) in Fig. 2.7.

Using second-order central differences we discretize all $\boldsymbol{x}$-dependent derivatives in (1.59)–(1.63) except for the convective velocity terms. Since the convective terms in the momentum equations (1.59) are prone for numerical instabilities, we discretize these terms with a 5th-order WENO scheme [63] to avoid oscillatory solutions. On the downside, the high-order of the WENO scheme requires a large discretization stencil of three grid cells in each coordinate direction. Consequently, the boundary of O has to be extended with an artificial strip of three grid cells, the so called ghost cells. As a result, the full grid for the velocity field including the ghost cells has the index positions $i = -2$ up to $i = i_{\max} + 3$ and analogously for j and k. This simplifies to $i = 0$ up to $i = i_{\max} + 1$ for the pressure field p as we only employ central differences here. More details on the spatial discretization of the macroscopic variables in NaSt3DGPF can be found in Griebel et al. [48] and in Croce et al. [31]. In Section 1.1, we discuss macroscopic constitutive equations which represent an alternative approach to model the stress tensor $\boldsymbol{\tau}_p$. The discretization of these macroscopic equations in NaSt3DGPF is explained in detail in

Claus [25].

On the micro-scale, we discretize the random field $\boldsymbol{Q}(\boldsymbol{x},t) = (\boldsymbol{Q}_1(\boldsymbol{x},t),\dots,\boldsymbol{Q}_N(\boldsymbol{x},t)) \in D_1 \times \dots \times D_N \subset \mathbb{R}^{3N}$ in space in the cell centers analogously to p and $\boldsymbol{\tau}_p$, i.e. at the positions $((i-0.5)\Delta x, (j-0.5)\Delta y, (k-0.5)\Delta z)$ with $i = 1,\dots,i_{\max}$, $j = 1,\dots,j_{\max}$ and $k = 1,\dots,k_{\max}$. This is necessary to avoid an interpolation of the random field $\boldsymbol{Q}(\boldsymbol{x},t)$ since $\boldsymbol{\tau}_p$ directly depends on $\boldsymbol{Q}(\boldsymbol{x},t)$ according to (1.62). As discussed in Section 2.1, each discrete random field $\boldsymbol{Q}(\boldsymbol{x}_{ijk},t)$ is represented by a number of $s = 1,\dots,M_s$ samples $\boldsymbol{Q}^{(s)}(\boldsymbol{x}_{ijk},t)$. Consequently, the total number of samples with dimensionality $\mathbb{R}^{3N}$ is of order $O(M_s\, i_{\max}\, j_{\max}\, k_{\max})$.

The discrete Brownian configuration fields $\boldsymbol{Q}^{(s)}_{i,j,k}(t)$ evolve according to the SPDE (1.61). We explicitly note that the Wiener process $\boldsymbol{W}(t)$ in (1.61) depends on time but not on space. This artificial spatial correlation since $\boldsymbol{W}(t)$ is uniform in space is a key idea of the Brownian configuration field method. It ensures a coupling between different Brownian configuration fields which strongly reduces the stochastic noise in space. We therefore observe a smooth stress tensor field in space in our simulations in Chapter 3. Nevertheless, the stress tensor still shows large oscillations over time. Furthermore, we discretize the convective part of (1.61) with a second-order QUICK scheme [85]. This has to be understood in the sense that derivatives with respect to $\boldsymbol{x}$ are performed individually for each sample $s = 1,\dots,M_s$ with the neighboring cell samples. A QUICK scheme has a difference stencil that covers two neighboring cells in the flow domain but only one neighboring cell on the boundary, i.e. we only require one artificial ghost cell for $\boldsymbol{Q}^{(s)}_{i,j,k}(t)$ on the boundary. As a result, the full grid for the Brownian configuration fields is described by an index set with range $i = 0,\dots,i_{\max}+1$, $j = 0,\dots,j_{\max}+1$ and $k = 0,\dots,k_{\max}+1$.

Boundary conditions

The boundary conditions for the velocity field $\boldsymbol{u}$ have been specified in (1.3a)–(1.3c) as

$$\begin{aligned} \boldsymbol{u}|_{\Gamma_1} &= \boldsymbol{u}_0 && \text{on the inflow boundary } \Gamma_1, \\ \boldsymbol{u}|_{\Gamma_2} &= 0 && \text{on the no-slip boundary } \Gamma_2, \\ \partial_{\boldsymbol{n}}(\boldsymbol{u}\cdot\boldsymbol{n})|_{\Gamma_3} = 0,\ \partial_{\boldsymbol{n}}(\boldsymbol{u}\cdot\boldsymbol{t})|_{\Gamma_3} &= 0 && \text{on the outflow boundary } \Gamma_3 \end{aligned}$$

and with $\partial O = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3$, $\boldsymbol{n}$ as outward pointing unit normal and $\boldsymbol{t}$ as tangential vector on ∂O. Furthermore, we use homogeneous Neumann boundary conditions for the pressure field p. These conditions can directly be discretized using finite differences.

Next, we turn to the boundary conditions for the Brownian configuration fields. We set

$$\boldsymbol{Q}^{(s)}(\boldsymbol{x},t)|_{\Gamma_1} = \boldsymbol{Q}^{0,(s)} \qquad \text{on the inflow boundary } \Gamma_1, \tag{2.19}$$

$$\nabla_{\boldsymbol{x}}\boldsymbol{Q}^{(s)}(\boldsymbol{x},t)|_{\Gamma_{2,3}} = 0 \qquad \text{on } \Gamma_2 \text{ and } \Gamma_3 \tag{2.20}$$

for all $s = 1,\dots,M_s$ realizations. One approach to generate configuration fields $\boldsymbol{Q}^{0,(s)}$ for a Dirichlet boundary condition on Γ_1 bases on previous simulations in a two-dimensional flow space. In this case, we first evolve a set of random samples $\boldsymbol{Q}^{(s)}|_\Gamma$ for a fixed inflow velocity $\boldsymbol{u}_0$ on Γ_1 from equilibrium towards steady state. The actual simulation in three-dimensional flow space $O \subset \mathbb{R}^3$ then reuses $\boldsymbol{Q}^{(s)}|_\Gamma$ on the two-dimensional inflow boundary. It is crucial to reuse the same number M_s of stochastic realizations as well as the same Brownian path to evolve the configuration fields towards a steady state. This ensures that $\boldsymbol{Q}^{0,(s)}$ represents an adequate inflow field on Γ_1.

2.2.2. Temporal discretization

We first consider the temporal discretization of the equations for conservation of momentum (1.59) and conservation of mass (1.60). The discretization bases on a semi-implicit scheme proposed by Bell et al. [9]. The implementation into NaSt3DGPF is described in Verleye et al. [130]. In this case, the diffusive velocity terms in (1.59) are discretized with an implicit 2nd-order Crank-Nicolson scheme to avoid severe CFL-type time step restrictions in the laminar flow regime. This is of practical importance since we primarily consider flow situations with low Reynolds numbers but with high Deborah numbers so that elastic effects are dominant. For the convective velocities in (1.59), we employ an explicit 2nd-order Adams-Bashforth scheme. This scheme leads to a restriction on the time-step width.

The temporal discretization of the stochastic partial differential equation (1.61) depends on the specific choice of the spring force term $\boldsymbol{F}$ in (1.63). The Hookean and the FENE-P spring in (1.43a) and (1.43c) do not restrict the possible spring extension so that the corresponding configuration space is $D = \mathbb{R}^{3N}$. For these spring models, we use an explicit Euler-Maruyama scheme for temporal discretization which is of first-order accuracy in time. In the applications, however, the first order accuracy in time is not a severe restriction. This is due to the fact that the accuracy of $\boldsymbol{\tau}_p$ is primarily restricted by the variance which reduces with an increasing number of samples M_s

The situation differs for the FENE spring (1.43b). Here, the configuration space D_i for each spring segment is an open sphere in $\mathbb{R}^3$ with radius $b(N)^{1/2}$ centered at the origin. Consequently, a physical solution has to fulfill $\|\boldsymbol{q}_i\|^2 < b(N)$ for all segments $i = 1, \dots, N$. This restriction, however, is not necessarily fulfilled by an explicit time-discretization scheme. For this reason, we use a time-discretization which is implicit in the spring force term. This approach was proposed by Öttinger [99]. The basic idea is that the extension of the spring segments $\|\boldsymbol{q}_i^{n+1}\|$ with $i = 1, \dots, N$ in the new time-step $n + 1$ is the only unknown that has to be determined. This leads to a cubic equation for $\|\boldsymbol{q}_i^{n+1}\|$ which has a unique solution in D_i.

Semi-implicit projection method

Let $\boldsymbol{u}^n$, p^n, $\boldsymbol{\tau}_p^n$ and $\boldsymbol{Q}^n$ denote the discretized variables in time step $t^n \in [0, T]$. Furthermore, $\boldsymbol{R}^n$ is a discrete equilibrium control variate of $\boldsymbol{Q}^n$ at t^n as described in Section 2.1.2. We then conduct the following steps to compute the discrete solution at $t^n + \Delta t^n = t^{n+1} \in (0, T]$:

1. In case of the Hookean and the FENE-P spring forces (1.43a) and (1.43c), respectively, we solve the stochastic ODE that results from the SPDE (1.61) after spatial discretization. Consequently, we apply a step of an explicit Euler-Maruyama method at all grid nodes $\boldsymbol{x}_j$, for all spring segments $i = 1, \dots, N$ and for all $s = 1, \dots, M_s$ stochastic samples according to

$$\boldsymbol{Q}_i^{n+1,(s)}(\boldsymbol{x}_j) = \boldsymbol{Q}_i^{n,(s)}(\boldsymbol{x}_j) + \Big[- \boldsymbol{u}^n \nabla \boldsymbol{Q}_i^{n,(s)}(\boldsymbol{x}_j) + (\nabla_{\boldsymbol{x}} \boldsymbol{u}^n)^T \boldsymbol{Q}_i^{n,(s)}(\boldsymbol{x}_j) - \frac{1}{4\,De(N)} \sum_{k=1}^{N} A_{ik} \cdot \boldsymbol{F}\left(\boldsymbol{Q}_k^{n,(s)}(\boldsymbol{x}_j)\right) \Big] \Delta t^n + \sqrt{\frac{\Delta t^n}{2De(N)}} \left[\boldsymbol{t}_{i+1}^{(s)} - \boldsymbol{t}_i^{(s)} \right] \quad \text{with } \boldsymbol{t}_i^{(s)}, \boldsymbol{t}_{i+1}^{(s)} \sim \boldsymbol{N}(0,1)^3. \tag{2.21}$$

We note that $\boldsymbol{x}_j$ is a short-hand notation for the three-dimensional grid as used in Fig. 2.7. Here, we only write $\boldsymbol{x}_j$ instead of $\boldsymbol{x}_{i,j,k}$ to avoid confusion since the index i denotes the spring segment in this case. Furthermore, A_{ik} are the components of the *Rouse matrix* (1.28).

If $\boldsymbol{F}$ is a FENE spring force (1.43b), we employ a semi-implicit Euler-Maruyama method

$$\begin{aligned}\left(1+\frac{A_{ii}\Delta t^n}{4\,De(N)}\right)\boldsymbol{F}\left(\boldsymbol{Q}_i^{n+1,(s)}(\boldsymbol{x}_j)\right) =& \boldsymbol{Q}_i^{n,(s)}(\boldsymbol{x}_j)+\Big[-\boldsymbol{u}^n\nabla\boldsymbol{Q}_i^{n,(s)}(\boldsymbol{x}_j)+(\nabla_x\boldsymbol{u}^n)^T\boldsymbol{Q}_i^{n,(s)} \\ &-\frac{1}{4\,De(N)}\sum_{\substack{k=1\\k\neq i}}^{N}A_{ik}\cdot\boldsymbol{F}\left(\boldsymbol{Q}_k^{n,(s)}(\boldsymbol{x}_j)\right)\Big]\Delta t^n \\ &+\sqrt{\frac{\Delta t^n}{2De(N)}}\left[\boldsymbol{t}_{i+1}^{(s)}-\boldsymbol{t}_i^{(s)}\right]\quad\text{with } \boldsymbol{t}_i^{(s)},\boldsymbol{t}_{i+1}^{(s)}\sim\boldsymbol{N}(0,1)^3\end{aligned}\tag{2.22}$$

and solve this equation for $\boldsymbol{Q}_i^{n+1,(s)}(\boldsymbol{x}_j)$.

In both equations, $\boldsymbol{t}_i^{(s)}$ and $\boldsymbol{t}_{i+1}^{(s)}$ denote a triple of independent Gaussian random variables with zero mean and variance one for $s=1,\ldots,M_s$.

2. Advance the equilibrium control variate according to

$$\boldsymbol{R}_i^{n+1,(s)} = \boldsymbol{R}_i^{n,(s)}-\frac{\Delta t^n}{4\,De(N)}\sum_{k=1}^{N}A_{ik}\cdot\boldsymbol{F}\left(\boldsymbol{R}_k^{n,(s)}\right)+\sqrt{\frac{\Delta t^n}{2De(N)}}\left[\boldsymbol{t}_{i+1}^{(s)}-\boldsymbol{t}_i^{(s)}\right]\quad\text{with } \boldsymbol{t}_i^{(s)},\boldsymbol{t}_{i+1}^{(s)}\sim\boldsymbol{N}(0,1)^3.\tag{2.23}$$

For an equilibrium control variate, it is crucial to use the same samples for the Brownian path of the Brownian configuration field $\boldsymbol{Q}^n$ as for the control variate $\boldsymbol{R}^n$. This ensures a high correlation of $\boldsymbol{Q}^n$ and $\boldsymbol{R}^n$ and therefore a better variance reduction.

3. We compute the new stress tensor according to Kramers' expression as

$$\begin{aligned}\boldsymbol{\tau}_p^{n+1}(\boldsymbol{x}_j)=\frac{\alpha_{b,d}(1-\beta)}{De(N)}\cdot\frac{3}{(N+1)^2-1}\sum_{i=1}^{N}\Big(\frac{1}{M_s}\sum_{s=1}^{M_s}\Big[&\boldsymbol{Q}_i^{n+1,(s)}(\boldsymbol{x}_j)\otimes\boldsymbol{F}(\boldsymbol{Q}_i^{n+1,(s)}(\boldsymbol{x}_j)) \\ &-\boldsymbol{R}_i^{n+1,(s)}\otimes\boldsymbol{F}(\boldsymbol{R}_i^{n+1,(s)})\Big]-\mathbf{Id}\Big).\end{aligned}\tag{2.24}$$

Note that equation (2.24) uses the fact that the expectation $\langle\boldsymbol{R}^{n+1}\otimes\boldsymbol{F}(\boldsymbol{R}^{n+1})\rangle$ is zero since $\boldsymbol{R}^{n+1}$ is an equilibrium control variate.

4. We solve a Helmholtz equation

$$\left(\boldsymbol{Id}-\frac{\Delta t^n\beta}{2Re}\Delta\right)\boldsymbol{u}^* = \boldsymbol{u}^n-\Delta t^n\left(\nabla p^n+\boldsymbol{u}^n\cdot\nabla\boldsymbol{u}^n-\frac{\beta}{2Re}\Delta\boldsymbol{u}^n-\frac{1}{Re}\nabla\cdot\boldsymbol{\tau}_p^{n+1}\right)\tag{2.25}$$

by using an SSOR preconditioned CG method and obtain an intermediate velocity field $\boldsymbol{u}^*$.

5. We use $\boldsymbol{u}^*$ on the right-hand side of a Poisson problem to compute a pressure correction ϕ^{n+1}

which ensures that $\boldsymbol{u}^{n+1}$ is divergence free, i.e.

$$\Delta\phi^{n+1} = \frac{1}{\Delta t^n}\nabla\cdot\boldsymbol{u}^* \tag{2.26}$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^* - \Delta t^n \nabla\phi^{n+1}. \tag{2.27}$$

The Poisson problem is solved with an AMG-preconditioned BiCGStab solver; see Metsch [96].

6. We obtain the new pressure field by computing

$$p^{n+1} = p^n + \phi^{n+1} - \frac{\Delta t^n\beta}{2Re}\Delta\phi^{n+1}. \tag{2.28}$$

Time-step restriction

It is generally known that explicit time-discretization schemes are only stable for limited time-step widths. We employ a semi-implicit scheme which is implicit in the diffusive velocity terms in the momentum equation and also implicit in some spring force contributions. For this reason, larger time-steps than in a fully explicit scheme are possible. We next discuss the restrictions on the time-step width Δt for our specific implementation. The results can be derived, for instance, by a von Neumann stability analysis [24].

Compared to purely Newtonian fluids, the momentum equations (1.59) contain an additional stress tensor $\boldsymbol{\tau}_p$ which further restricts the size of Δt. Trebotich et al. [126] state the restriction on Δt for the convective velocity term in x-direction, which we denote as Δt_x, is

$$\Delta t_x \leq \frac{\Delta x}{\max_{i,j,k}\left[|u_{i,j,k}| + \sqrt{2(\tau_{xx} + \frac{1-\beta}{De})/Re}\right]}. \tag{2.29}$$

Analogously, we derive restrictions Δt_y and Δt_z in the coordinate directions y and z, respectively. As a result, the restriction in time due to the contribution of the convective velocities is

$$\Delta t_c = \min(\Delta t_x, \Delta t_y, \Delta t_z). \tag{2.30}$$

The restriction in (2.30) is the famous Courant-Friedrich-Lewy (CFL) condition [28].

If our scheme was explicit in the diffusive velocity terms, we would obtain a similar restriction as for the convective velocities. Other restrictions can occur due to body forces such as gravity; see Croce [29] for a detailed explanation.

Next, we consider time-step restrictions for the SPDE (1.61). For this purpose, we cite a theorem that connects the stability of an SPDE with the stability of its corresponding deterministic PDE.

Theorem 2.3 [Stability of stochastic differential equations]
Let $d\boldsymbol{X}_t = \mu(\boldsymbol{X}_t, t)\,dt + \sigma(\boldsymbol{X}_t, t)\,d\boldsymbol{B}_t$ be a stochastic differential equation with drift term μ and diffusion term σ. A finite difference discretization of the previous stochastic differential equation is stable if and only if the corresponding discretization of the deterministic equation $d\boldsymbol{X}_t = \mu(\boldsymbol{X}_t, t)\,dt$ is stable.

Proof: The proof is given as Corollary 3.1 in Roth [116].

As a result, we only analyze the stability of the deterministic equivalent of (1.61). Furthermore,

Algorithm 2: Algorithm for variance reduced multiscale simulations of viscoelastic fluids

```
   Data: Initial conditions $\boldsymbol{u}^0$ and $p^0$ and $N$-segment chain density $\psi_{\text{eq}}$ according to (1.58a)–(1.58d).
   Result: Discrete solution $\boldsymbol{u}$, $p$, $\boldsymbol{\tau_p}$ and $\boldsymbol{Q}^{(s)}$ with $s = 1, \ldots, M_s$ at time $t \leq t_{\max}$.
 1 Set $t \leftarrow 0$, $n \leftarrow 0$;
 2 Initialize velocity and pressure field with $\boldsymbol{u}^0$ and $p^0$;
 3 Generate $s = 1, \ldots, M_s$ realizations $\boldsymbol{Q}^{(s)} = (\boldsymbol{Q}_1^{(s)}, \ldots, \boldsymbol{Q}_N^{(s)}) \sim \psi_{\text{eq}}$ // von Neumann rejection
      sampling according to Algorithm 1
 4 Initialize all BCFs with the equilibrium configuration, i.e. $\boldsymbol{Q}_{i,j,k}^{0,(s)} = \boldsymbol{Q}^{(s)}$ for all $i, j, k$;
 5 Reuse samples for the equilibrium control variate $\boldsymbol{R}^{0,(s)} \leftarrow \boldsymbol{Q}^{(s)}$ for $s = 1 \ldots, M_s$ ;
 6 Compute initial stress tensor values $\boldsymbol{\tau_p}^0$ by using (2.24) // is zero up to machine accuracy
 7 while $t \leq t_{max}$ do
 8     Compute new time-step size $\Delta t^n$ according to (2.30) without diffusive restriction;
 9     Set boundary values for $\boldsymbol{u}^n$ and $p^n$;
10     Set boundary values for the BCFs $\boldsymbol{Q}_{i,j,k}^{n,(s)}$;
11     Compute new configuration field values $\boldsymbol{Q}_{i,j,k}^{n+1,(s)}$ according to (2.21) or (2.22);
12     Compute new control variate $\boldsymbol{R}^{n+1,(s)}$ by using (2.23);
13     Obtain new stress tensor $\boldsymbol{\tau_p}^{n+1}$ as in (2.24) // Kramers' expression
14     Set boundary values for intermediate velocities $\boldsymbol{u}^*$;
15     Solve the Helmholtz equation for $\boldsymbol{u}^*$ according to (2.25);
16     Solve the Poisson problem (2.26) for the the pressure difference $\phi^{n+1}$;
17     Compute the new velocity field $\boldsymbol{u}^{n+1}$ by using the pressure difference (2.27);
18     Obtain the new pressure field $p^{n+1}$ according to (2.28);
19     $t \leftarrow t + \Delta t^n$, $n \leftarrow n + 1$
20 end
```

we employ a discretization of (1.61) which is implicit in the spring force F. Consequently, there is no stability restriction due to F in (1.61). As a result, only a convection equation with reaction term remains. Since we employ the same finite difference scheme for the Navier-Stokes equations as for the stochastic equation, the time-step restriction is very similar to the restrictions in (2.29) and (2.30). In fact, a Neumann stability analysis shows that the restriction in time-step width for the Navier-Stokes equations is more severe than for the Brownian configuration field equation (1.61). For this reason, it is sufficient to limit Δt according to (2.29) and (2.30) in the applications.

Algorithm for multiscale viscoelastic flow solver

Finally, we state Algorithm 2 which summarizes the findings of Section 2.2.1 and Section 2.2.2 on the temporal and spatial discretization, respectively. The implementation in NaSt3DGPF bases on this algorithm.

We note that the steps 11 and 13 in Algorithm 2 for the evolution of the Brownian configuration fields $\boldsymbol{Q}^n$ are the most demanding with respect to the computing time. This is due to the fact that $\boldsymbol{Q}^n$ is a short-hand notation for M_s samples in the center of the grid cell (i, j, k). One simple approach to reduce the computing time bases on parallelization. This topic will be covered in the next section.

2.3. Complexity and parallelization

In Section 2.3.1, we first discuss the complexity of our multiscale viscoelastic flow solver. Furthermore, we state the CPU and memory requirements for the numerical simulations in Chapter 3. These requirements cannot be fulfilled on single-core workstations. Instead of that, massively parallel computations are necessary to reduce the computing time. After that, parallelization will be considered in Section 2.3.2.

However, even on massively parallel computers the computing time is in the order of days up to weeks for typical problems of interest. Therefore, we require a more advanced approach to reduce the computational complexity of the problem. Our method of choice to overcome these limitations uses a sparse grid instead of a full grid discretization. Sparse grid discretizations will be discussed in Chapters 4–6.

2.3.1. Complexity of full grid approach

Chapter 1 covers two different approaches for modeling viscoelastic fluids: these are macroscopic approaches on the one hand, see Section 1.1, and multiscale approaches on the other hand, see Section 1.2. Macroscopic methods have a computational complexity that is comparable to the complexity of a classical Newtonian fluid. In contrast to this, multiscale methods have a higher computational complexity. This is due to the more complex modeling of the polymer chains.

The computational complexity of a non-Newtonian macroscopic flow solver can be estimated as the following:

- Let M_g be the number of grid cells per coordinate direction, then the grid complexity is $O(M_g^3)$ unknowns for the velocity, the pressure and the stress tensor field.
- If Algorithm 2 requires M_t time-steps, then the computational complexity of a full simulation is $O(M_t \cdot M_g^3)$ operations.

The complexity increases in the case of a multiscale polymer model due to the enormous complexity for the Brownian configuration fields:

- An N-segment Brownian configuration field $\boldsymbol{Q}$ is approximated with $s = 1, \ldots, M_s$ samples $\boldsymbol{Q}^{(s)} \in \mathbb{R}^{3N}$. After discretization, a sample in $\mathbb{R}^{3N}$ can be stored as a sequence of N samples in $\mathbb{R}^3$. Then, the grid complexity of a Brownian configuration field is of order $O(N \cdot M_s \cdot M_g^3)$ with the same constant of proportionality as for the macroscopic variables.
- The total number of operations in Algorithm 2 is of order $O(N \cdot M_t \cdot M_s \cdot M_g^3)$ where M_t is again the number of time steps.

The complexity estimate shows that the stochastic approximation which bases on random fields leads to an enormous increase in computing time (number of operations) and in memory requirement (number of unknowns). In Table 2.1 we compare the grid complexity of three multiscale simulations from Chapter 3. In Section 3.1.2 we simulate a homogeneous extensional flow, i.e. the solution does not depend on the position in space $\boldsymbol{x} \in \mathcal{O}$. Therefore, the memory requirement is in the order of several hundred megabytes only although the number of samples $M_s = 10^6$ is high. Next, in Section 3.2.1 a planar contraction flow is simulated. In this case, the flow space $\mathcal{O}$ is a subset of $\mathbb{R}^2$. Since NaSt3DGPF

Table 2.1.: Grid complexity of the Brownian configuration fields for three simulations in Chapter 3.

	extensional flow	planar contraction flow	4 : 1 contraction flow
section	3.1.2	3.2.1	3.2.2
spatial grid M_g^3	1 cell (homogeneous)	$640 \times 128 \times 3$ cells	$380 \times 64 \times 64$ cells
samples per cell M_s	$1.5 \cdot 10^5$	4000	1200
spring segments N	5	1	5
random field samples	$1.5 \cdot 10^6$	$983 \cdot 10^6$	$1.87 \cdot 10^9$
dimension of samples	15-dim	3-dim	15-dim
memory [GB]	0.1	22	208

only creates three-dimensional grids, we emulate a two-dimensional grid by using only three grids in the third coordinate direction and applying periodic boundary conditions there. As a result, the memory requirement for the Brownian configuration fields is about 22 GB. Finally, the last column of Table 2.1 lists the memory requirements for a three-dimensional 4 : 1 contraction flow. This problem in Section 3.2.2 is the most demanding in this thesis with respect to the computational complexity. This is due to the fine spatial grid with $380 \times 64 \times 64$ cells. For this reason, the memory requirement is about 208 GB. We note that simulations for the second and third problem of Table 2.1 were conducted on a parallel computer.

Complexity of a control variate for PDE system In Section 2.1.3 we discuss control variates that allow us to decrease the statistical error. As mentioned before, the complexity of the Brownian configuration fields is of order $O(N\, M_s\, M_g^3)$ in each time step. Next, we analyze how this complexity increases for an equilibrium control variate on the one hand and for a macroscopic model control variate on the other hand. The complexity noticeably differs for these two control variates since

- an equilibrium control variate requires only M_s samples of $\boldsymbol{Y}_t$ in (2.17) independent of space $\boldsymbol{x}$, i.e. there is practically no increase in cost, and
- a macroscopic model control variate requires the same number of samples of order $O(N\, M_s\, M_g^3)$ as used for Brownian configuration fields, i.e. the total computational cost increases to $O(N\, M_g^3\, (M_s + M_s))$ with the same constant of proportionality.

Consequently, a macroscopic model control variate only makes sense if it reduces the root-mean-square error in (2.5) by a factor larger than $\sqrt{2}$. Otherwise, it is better to simply double the number of samples $M_s \to 2M_s$. For this reason, all complex simulations in Chapter 3 with the three-dimensional flow solver *NaSt3dGPF* employ an inexpensive equilibrium control variate for which the complexity analysis in Table 2.1 remains the same. Furthermore, Algorithm 2 contains an equilibrium control variate.

2.3.2. Parallelization

A natural approach for parallelization bases on domain decomposition; see e.g. Smith et al. [120]. This method decomposes the flow space $\mathcal{O}$ into subdomains. Each subdomain is assigned to a different processor / CPU core. Processor cores that share a common boundary face have to exchange data in

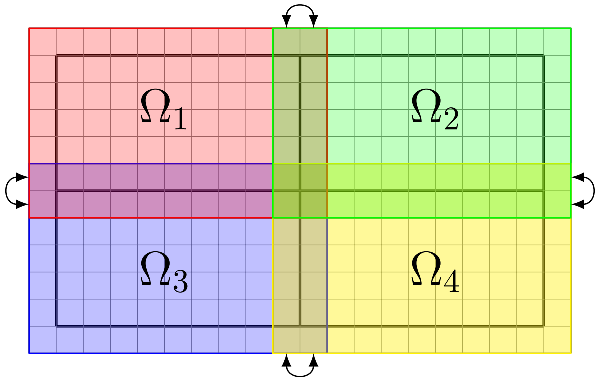


Figure 2.8.: Two-dimensional illustration of the domain decomposition approach for parallelization.

Table 2.2.: Parallel code performance of Algorithm 2 on CPU cluster *Siebengebirge*.

#cores	*#nodes*	time [s]	speedup	efficiency
32	1	13531	1.0	100 %
64	2	7693	1.76	88 %
128	4	3726	3.63	91 %
160	5	2937	4.61	92 %

every time step. We illustrate the principle of data exchange in two dimensions in Fig. 2.8 which is taken from Rüttgers et al. [118].

The data exchange is performed with the Message Passing Interface (MPI). For this purpose, every local subdomain has additional ghost rows adjacent to the cell faces which do not belong to ∂O. These ghost cells store the flow field from the neighboring cores. For an optimal efficiency of the parallelization it is essential to minimize the size of these additional ghost cells. We therefore decide for cubic shaped subdomains for the individual cores. Data exchange is required for the velocity field, the pressure field and for the Brownian configuration field but not for the stress tensor $\boldsymbol{\tau}_p$. This is due to the fact that $\boldsymbol{\tau}_p$ directly depends on the stochastic fields according to (1.62). Since the data exchange is time-consuming, it can be more efficient to directly compute $\boldsymbol{\tau}_p$ if the number of samples M_s is low. This is the case for the complex multiscale simulations in Chapter 3. Analogously to the sequential code, data exchange of the stochastic samples requires most of the time. More details on the data exchange of the macroscopic variables in *NaSt3DGPF* is given in Croce [29].

The parallel results in Chapter 3 have been computed on the parallel CPU cluster *Atacama*. The cluster is operated by the Institute for Numerical Simulation and the Sonderforschungsbereich 1060 at University of Bonn. *Atacama* consists of 78 Dell PowerEdge M620 compute nodes with 16 Intel Xeon CPU E5-2650 2.60 GHz CPUs per node. Therefore, the total CPU number is 1248. Furthermore, the system has a main memory of 4992 GB and MPI communication is performed with 56 Gb/Section (4X FDR) Infiniband. The Linpack performance of the system is 20,630 GFlop/s which is 80% of the theoretical peak performance.

In the literature, several articles investigate the parallel code performance of *NaSt3DGPF*. Zaspel and Griebel [140] study the scaling behavior of a *NaSt3DGPF* GPU portation using NVIDIA's CUDA framework. Furthermore, the parallel code performance of *NaSt3DGPF* on *Atacama* is analyzed by

Rüttgers et al. [118] for a Newtonian fluid simulation through a complex geometry from nature, the scopa of an oil collecting bee. Finally, the scaling behavior of an implementation of the multiscale Algorithm 2 in *NaSt3DGPF* is investigated by Griebel and Rüttgers [51]. The scaling results for a simple test problem are listed in Table 2.2. We note that these results have been obtained on a smaller cluster called *Siebengebirge*. This cluster consists of 160 Intel Xeon X7560 2.226 GHz CPUs on 5 computing nodes and is operated by the Institute for Numerical Simulation. As expected, the scaling behavior on *Siebengebirge* is close to optimal. The efficiency is always close to or above 90%. The high parallel efficiency results from the large number of stochastic samples, see Table 2.1, that can be parallelized efficiently. We expect similar results for the larger cluster *Atacama* since the same code is used for a similar CPU architecture. Therefore, we conclude that *NaSt3DGPF* is perfectly adapted for massively parallel multiscale simulations of viscoelastic fluids. Better results can only be achieved by reducing the complexity of Algorithm 2. For this purpose, we employ a discretization on a sparse grid which will be the considered in the second part of this thesis.

3. Numerical results on full grids

This chapter contains numerical results on full grids that were computed with the multiscale viscoelastic flow solver described in Chapter 2. In the following applications, we distinguish

- homogeneous flows and
- complex multiscale flows.

Homogeneous flow fields refer to stationary velocity fields with a velocity gradient that does not depend on space. In this case, it is not necessary to solve the Navier-Stokes equations. Furthermore, the Brownian configuration fields for the underlying polymer structure simplify from a random field to a stochastic process. The numerical treatment of this process is described in Section 2.1. We present numerical results for homogeneous flow fields in Section 3.1.

Complex multiscale flows denote the general case in which the coupled multiscale flow system in (1.59)–(1.63) is considered. A discrete approach for the solution of the multiscale system is given in Algorithm 2 in Section 2.2. These flow fields exhibit an enormous computational complexity and require an efficient parallelization approach; see Section 2.3. The numerical results on complex flows are presented in Section 3.2.

3.1. Homogeneous flows

In non-Newtonian fluid mechanics, homogeneous shear and extensional flows are used to characterize the fluid's elasticity behavior. We here concentrate on extensional flows and refer to Section 6.1 or to Rüttgers [117] for a description of shear flows.

Extensional flows

An extensional flow stretches a fluid element in one or two coordinate directions and compresses it in the remaining ones. The rate of constant elongation is characterized by a dimensionless scalar elongation rate $\dot{\epsilon} > 0$. Then, the corresponding velocity field is

$$\boldsymbol{u} = (\dot{\epsilon}x, -\frac{\dot{\epsilon}}{2}y, -\frac{\dot{\epsilon}}{2}z) \qquad \text{for an uniaxial extensional flow and} \qquad (3.1a)$$

$$\boldsymbol{u} = (\dot{\epsilon}x, -\dot{\epsilon}y, 0) \qquad \text{for a planar extensional flow.} \qquad (3.1b)$$

We visualize both velocity fields in Fig. 3.1. This figure is taken from Rüttgers [117].

In the following, we focus on uniaxial extensional flows in x-direction as defined in (3.1a). The velocity field gradient for this flow field is $\nabla\boldsymbol{u}(\boldsymbol{x}, t) = \operatorname{diag}(\dot{\epsilon}, -\dot{\epsilon}/2, -\dot{\epsilon}/2) := \boldsymbol{\kappa}$, i.e., it does not depend

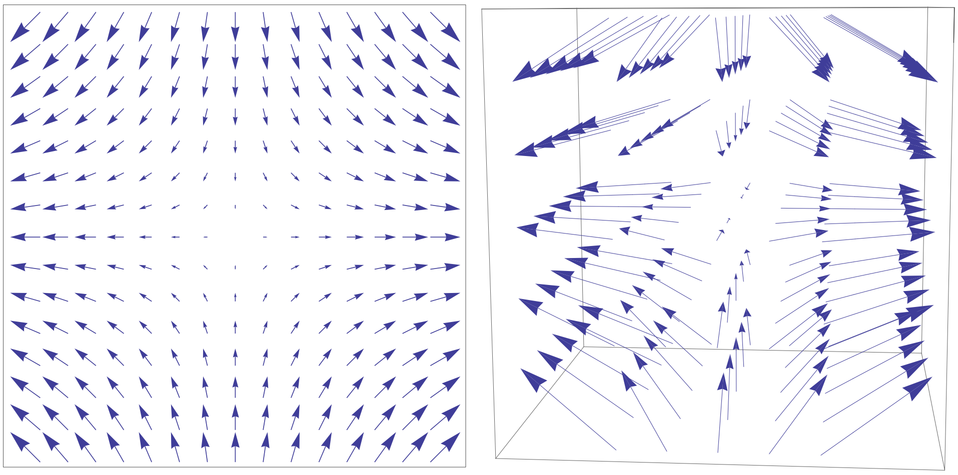


Figure 3.1.: Comparison of stationary planar and uniaxial extensional flow.

on space and time, and the coupled system in (1.59)–(1.63) simplifies to

$$d\boldsymbol{Q}(t) = \Big[\boldsymbol{\kappa}^T \boldsymbol{Q}(t) - \frac{1}{4\,De(N)} \boldsymbol{A} \cdot \boldsymbol{F}\,(\boldsymbol{Q}(t))\,\Big]\,dt + \sigma d\boldsymbol{W}(t) \tag{3.2}$$

$$\boldsymbol{\tau_p}(t) = \frac{3(1-\beta)\,(b(N)+5)}{b(N)\,De(N)\,((N+1)^2-1)} \sum_{i=1}^{N} \left(\mathbb{E}[\boldsymbol{Q}_i(t) \otimes \boldsymbol{F}_i(\boldsymbol{Q}_i(t))] - \boldsymbol{Id}\right). \tag{3.3}$$

In this simplified case, the only quantities of interest are the stochastic process $\boldsymbol{Q}(t) = (\boldsymbol{Q}_1(t), \ldots, \boldsymbol{Q}_N(t))$ and the corresponding stress tensor $\boldsymbol{\tau_p}(t)$. Furthermore, only the diagonal components τ_{xx}, τ_{yy} and τ_{zz} of $\boldsymbol{\tau_p}(t)$ are non-zero.

3.1.1. Dumbbell models

The first application is an extensional flow for the simplified dumbbell model case $N = 1$. For this model, we compare the outcome of the Hookean spring force, the FENE spring force and the FENE-P spring force. The initial condition for the probability density function $\psi_{eq}(\boldsymbol{q})$ in the dumbbell model is given in (1.56a)–(1.56d). Note again that the initial stress tensor $\boldsymbol{\tau_p}(t)$ that is computed from the random variable $\boldsymbol{Q}(0) \sim \psi_{eq}(\boldsymbol{q})$ in (3.3) has an expectation of zero. The steady state stress tensor in a stationary velocity field is known for some dumbbell models. We discuss these analytical results in the following.

Hookean dumbbell solution in an extensional flow

The linear Hookean dumbbell model can be written as a macroscopic stress tensor equation; see e.g. Lozinski [88]. This constitutive equation for $\boldsymbol{\tau_p}$ results from (1.8) and (1.9) for the special case $\epsilon = \xi = 0$. According to Owens and Phillips [100], the steady state stress tensor in an x-directional extensional flow is

$$\tau_{xx} = \frac{2\,\dot{\epsilon}}{1 - 2De\,\dot{\epsilon}}, \qquad \tau_{yy} = \tau_{zz} = -\frac{\dot{\epsilon}}{1 + De\,\dot{\epsilon}} \tag{3.4}$$

for a Deborah number $De < 1/(2\dot{\epsilon})$. Furthermore, the Hookean dumbbell model predicts an unlimited first normal stress component for $De \geq 1/(2\dot{\epsilon})$.

FENE dumbbell solution in an extensional flow

In homogeneous flow fields there exists a steady state solution for the FENE probability density function ψ in flow fields with a symmetric and constant velocity gradient $\boldsymbol{\kappa}$. This solution also applies to homogeneous extensional flows. The solution is

$$\psi_{\text{analytic}} = C\left(1 - \frac{\|\boldsymbol{q}\|^2}{b}\right)^{b/2} \exp(De\, \boldsymbol{\kappa} : \boldsymbol{q} \otimes \boldsymbol{q}) \tag{3.5}$$

with $\boldsymbol{q} \in \mathcal{B}(0, b(1)^{1/2}) \subset \mathbb{R}^3$ and C as a normalization constant. For further information, Bird et al. [12] state this solution as equation (13.2-14). The steady state solution ψ_{analytic} allows to compute the expectation in (3.3).

Spring model comparison for moderate extensional flow

In Table 3.1 we first list the parameters for the following simulations. Using these parameters, we compute the analytic steady state stress tensor values for the Hookean and the FENE dumbbell model with equation (3.4) and (3.5). The parameters are chosen such that the first normal stress component and the extensional viscosity tend to infinity for the Hookean dumbbell model.

Furthermore, the analytical formula for the FENE model yields

- $\tau_{xx}^{\text{analytic}} \approx 9.3991$,
- $\tau_{yy}^{\text{analytic}} \approx -0.6643$,
- $\tau_{zz}^{\text{analytic}} \approx -0.6643$,
- zero for all shear stresses.
- $(\tau_{xx} - \tau_{yy})/\dot{\epsilon} \approx 10.634$

Next, we numerically solve the system of equations (3.2) and (3.3) with the parameters listed in the last rows of Table 3.1. We plot the extensional viscosity $\eta(\dot{\epsilon}) = (\tau_{xx} - \tau_{yy})/\dot{\epsilon}$ over time in Fig. 3.2. For the Hookean dumbbell model there is a linear increase of $\eta(\dot{\epsilon})$ over the full time range of the simulation. An analysis of the stochastic samples shows that the Hookean dumbbells become the more stretched the longer the simulation is performed. This results in an unlimited stress growth in this flow field.

On the other hand, the FENE and FENE-P model predict a finite steady state result for $\eta(\dot{\epsilon})$ after about $t = 8.0$. The numerical result for the FENE model tends to the analytical result of ≈ 10.634

Extensional flow with dumbbell model		
Deborah number	De	0.5
viscosity ratio	β	0.0
extensional rate	$\dot{\epsilon}$	1.0
maximum spring extension	b	60.0
stochastic samples	M	$2 \cdot 10^6$
time-step size (const.)	Δt	$5 \cdot 10^{-4}$

Table 3.1.: Parameters for the mathematical model and for the numerical simulations in which a dumbbell model evolves according to a three-dimensional extensional flow.

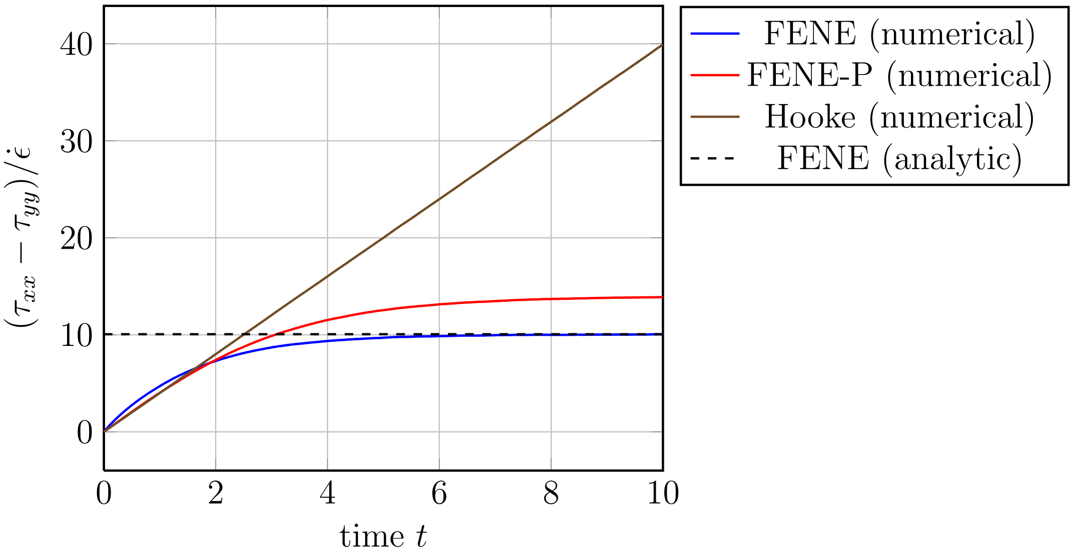


Figure 3.2.: Time-dependent extensional viscosity $\eta(\dot{\epsilon}) = (\tau_{xx} - \tau_{yy})/\dot{\epsilon}$ with $\dot{\epsilon} = 1$ for the FENE, FENE-P and Hookean dumbbell model.

that is indicated with a black dotted line in Fig. 3.2. Further investigations show that the error of the approximation reduces with order $O(M^{-1/2})$. We do not investigate the error reduction depending on M for the dumbbell case since this analysis is already performed in Section 6.2.1 of Rüttgers [117].

Moreover, the FENE-P model reaches a larger steady state result of about $\eta(\dot{\epsilon}) \approx 13.9$. This is expected since the FENE-P model restricts only the average dumbbell extension by $b^{1/2}$ such that $\mathbb{E}(\boldsymbol{Q}) \leq b^{1/2}$ for the FENE-P model instead of $\boldsymbol{Q} \leq b^{1/2}$ as for the FENE model. This results in higher stress tensor values for the FENE-P model in moderate and strong velocity fields. Consequently, the FENE-P model is an accurate approximation of the FENE model only in flow fields with low elongation rate $\dot{\epsilon}$.

Fig. 3.2 shows a further interesting aspect that separates the FENE model from the Hookean and the FENE-P model. The Hookean model predicts a linear increase of $\eta(\dot{\epsilon})$. In the initial phase of the simulation up to about $t = 1.8$, the FENE-P model also predicts a linear increase of $\eta(\dot{\epsilon})$. Both models assume a Gaussian initial distribution which is not the case for the FENE model. For this reason, the FENE model has a nonlinear behavior over the complete time range of the simulation.

3.1.2. Spring-chain models

In this section we consider an extensional flow for more general spring-chain systems with $N = 1, \ldots, 5$ segments. Each segment is connected with a nonlinear FENE spring. The physical system is fully determined by the system of equations (3.2) and (3.3). Moreover, we now analyze how the stress tensor development over time depends on the spring segment number N. This dependency, however, also depends on the chosen spring-chain gauge; cf. Section 1.2.3. In Table 3.2 we list the mathematical and numerical parameters for the following simulation. In contrast to the simulations in Section 3.1.1 for the dumbbell model, we now prescribe a stronger extensional rate $\dot{\epsilon} = 2.0$ and employ a larger maximum spring extension $b = 120$. For this reason, we expect larger stress tensor values as in the previous simulation.

Extensional flow with spring-chain model		
Deborah number	De	1.0
viscosity ratio	β	0.0
extensional rate	$\dot{\epsilon}$	2.0
spring model	FENE	
maximum spring extension	b	120.0
spring extension gauge	$b(N)$	de Gennes (1.48)
spring relaxation gauge	$\lambda(N)$	rel. time scale (1.51)
stochastic samples	M	$1.5 \cdot 10^5$
time-step size (const.)	Δt	$5 \cdot 10^{-4}$

Table 3.2.: Parameters for the FENE multi-bead-spring chain systems in a 3D extensional flow.

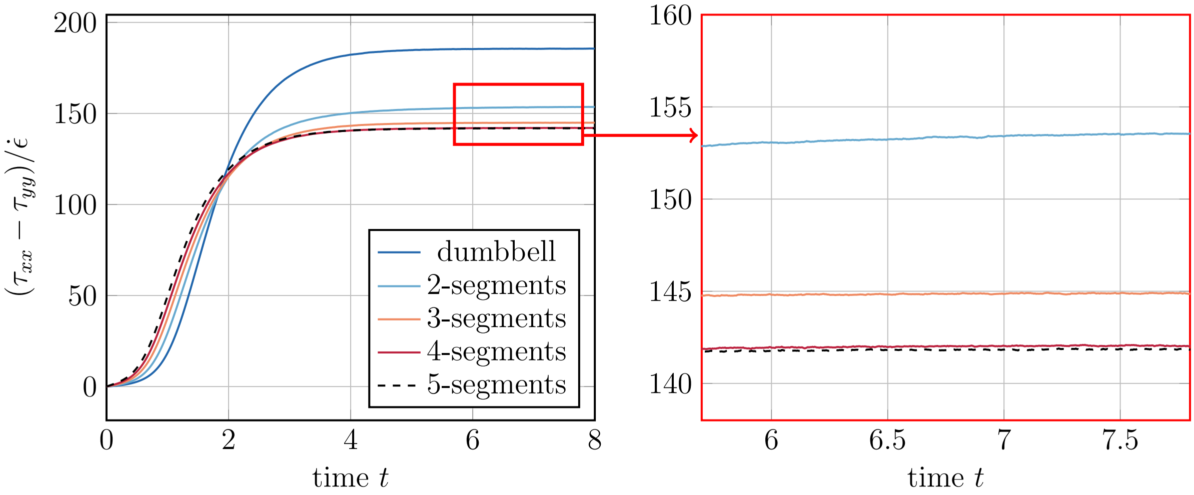


Figure 3.3.: Extensional viscosity $\eta(\dot{\epsilon}) = (\tau_{xx} - \tau_{yy})/\dot{\epsilon}$ with $\dot{\epsilon} = 2$ over time for FENE spring chains with different segment numbers N.

In Fig. 3.3 we visualize $\eta(\dot{\epsilon} = 2, t)$ for $t \in [0, 8.0]$ with $N = 1, \dots, 5$ spring segments. More precisely, on the left hand side (LHS) of Fig. 3.3 we plot $\eta(\dot{\epsilon} = 2, t)$ over the full temporal grid and on the right-hand side (RHS) we focus on a zoomed extract close to the steady state of the system. The zoomed extract is indicated with a red rectangle on the LHS of the figure.

The simulated steady state results are about 186 (dumbbell case), 154 ($N = 2$ segments), 145 ($N = 3$ segments) and about 142 for $N = 4$ and $N = 5$. We observe that the dumbbell model predicts the lowest extensional viscosity in the early stage of the simulation up to about $t = 2.0$ but the largest extensional viscosity in the end. The situations is reversed for $N = 5$. Here, the stress tensor values are largest in the initial phase but lowest at steady state. The early stage behavior is primarily caused by the mapping of the relaxation time $\lambda(N)$ according to (1.51). Due to this mapping, the relaxation time $\lambda(1)$ for the single dumbbell segment is larger than, for instance, the relaxation time $\lambda(5)$ for each of the five segments. This leads to a later increase in $\eta(\dot{\epsilon} = 2)$ for the dumbbell model as shown in Fig. 3.3. The steady state behavior seems to be more affected by the maximum spring segment extension $b(N)$ according to (1.48). The figure indicates that the dumbbell segment with maximum extensibility $b(1) = 120$ seems

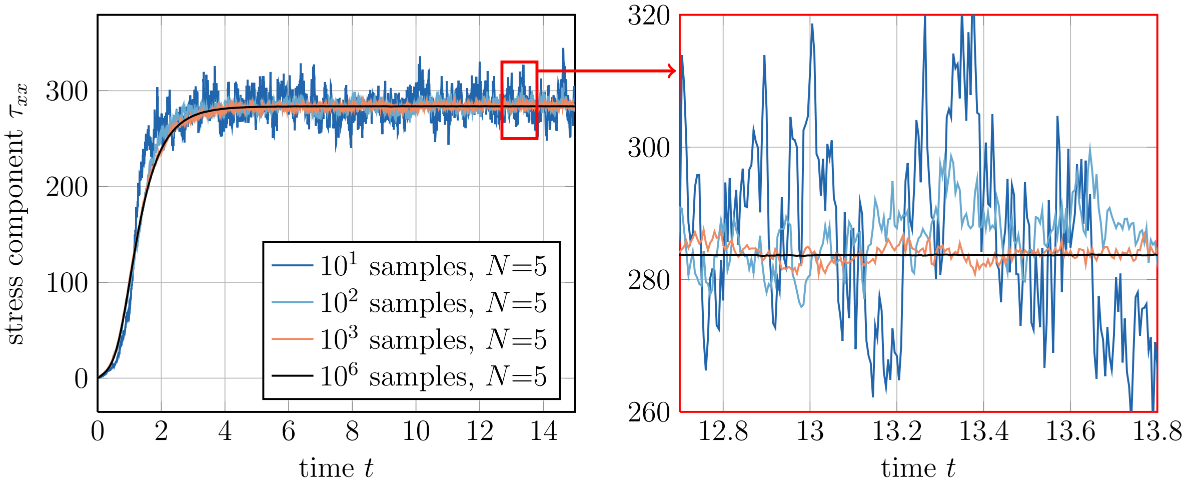


Figure 3.4.: Temporal evolution of τ_{xx} for a five segment spring-chain with different numbers of sample points M.

to be more elongated in average than a five segment chain with $b(5) = 24$ for each segment.

According to Fig. 3.3, the dumbbell model and the 2-segment chain strongly differ in their steady state result for $\eta(\dot{\epsilon} = 2)$. In contrast to this, the differences for $N = 4$ and $N = 5$ are negligible. Note that this effect is specific to the current application and to the gauge of the spring-chains and cannot be generalized to other flow situations. However, in this application the four-segment chains delivers a suitable approximation to the five-segment chain but with a lower computational complexity. This is an idea that will be used in Chapter 6 to reduce the computational complexity of multiscale polymer flow simulations.

Finally, we analyze the order of convergence for the first normal stress component τ_{xx} for a chain with $N = 5$ spring segments. Fig. 3.4 shows the time-dependent evolution of τ_{xx} depending on the number M of stochastic realizations. The numerical steady state result is $\tau_{xx} \approx 284$. The left-hand side (LHS) of Fig. 3.4 visualizes τ_{xx} for $t \in [0, 15]$. Again, a small red rectangle indicates a zoomed extract that is shown on the right-hand side (RHS) of the figure. The stress component increases from its initial value of zero at $t = 0.0$ up to about $t \approx 4.0$ when the system reaches its steady state. Then, τ_{xx} oscillates around its steady state result and the size of the oscillation, more precisely the variance of the solution, strongly depends on the number of samples M. As expected, the variance is largest for $M = 10$ samples and, in average, the more decreases the more M is increased. In total, we perform six simulations $l = 1, 2, \ldots, 6$ with different sample numbers $M_l = 10^l$. The numerical solution on the finest grid (black solid line) for level $l = 6$ does not show any visible variance with the chosen scaling of the x-axis in Fig. 3.4. We note that Fig. 3.4 does not include the results of the levels $l = 4$ and $l = 5$ since the results can hardly be distinguished from the result on level $l = 6$.

Next, we compare the expectation $\langle \tau_{xx} \rangle$ for the different levels $l = 1, \ldots, 6$ in Table 3.3. For this purpose, we approximate $\langle \tau_{xx} \rangle$ as

$$\langle \tau_{xx} \rangle = \frac{1}{\# M_T} \sum_{t_i \in M_T} \tau_{xx}(t_i) \quad \text{for } M_T \equiv \{t_i | \, t_i \geq 10.0\} . \tag{3.6}$$

Table 3.3.: Average of the normal stress component τ_{xx} for different numbers of stochastic realizations M_l and the corresponding rate of convergence. The same results are specified for the variance $\sigma^2(\tau_{xx})$.

l	M_l	$\langle\tau_{xx}\rangle$	$\varepsilon_{\text{rel}}(\tau_{xx})$	$p_l(\varepsilon_{\text{rel}})$	$\sigma^2(\tau_{xx})$	$p_l(\sigma^2)$
1	10^1	286.5353	1.0_{-2}		245.29	
2	10^2	285.8048	7.4_{-3}	0.13	25.73	0.98
3	10^3	283.9659	9.4_{-4}	0.90	1.90	1.13
4	10^4	283.8157	4.1_{-4}	0.36	0.17	1.04
5	10^5	283.7308	1.1_{-4}	0.56	0.02	0.93
6	10^6	283.6987			1.6_{-3}	1.10
$\langle p_l\rangle$				0.49		1.03

In this case, M_T contains all discrete points in time ≥ 10.0 to ensure that the variance error is dominant.

Moreover, the 4th column of Table 3.3 specifies an approximation to the relative error $\varepsilon_{\text{rel}}(\tau_{xx}) = |\tau_{xx} - \tau_{xx,\text{analytic}}|/|\tau_{xx,\text{analytic}}|$. In this case, an analytic formula for the steady state probability density function ψ results in a 15-dimensional integral for the expectation in (3.3) due to the high-dimensional configuration space. Since this integral cannot be easily solved with grid-based methods we approximate $\tau_{xx,\text{analytic}} \approx \tau_{xx}$ on level $l = 6$. We use this approximation of $\varepsilon_{\text{rel}}(\tau_{xx})$ for the levels $l = 1, \dots, 5$. This approximation of $\varepsilon_{\text{rel}}(\tau_{xx})$ now allows us to investigate the order of convergence p_l for the different sample levels l. For this purpose, we define the order of convergence between the levels $l - 1$ and l as

$$p_l(\varepsilon_{\text{rel}}) = \frac{\log(\varepsilon_{l-1}/\varepsilon_l)}{\log(10)} \tag{3.7}$$

where ε_l is a short-hand notation for $\varepsilon_{\text{rel}}(\tau_{xx})$ on level l. We list the order of convergence of the relative error in the 5th column of Table 3.3. As expected for a stochastic approach, the order of convergence strongly varies between 0.13 and 0.90. In the last row, we compute an average rate of convergence of 0.49. This result is very close to the expected result of 0.5.

Next, we investigate the variance of the solution $\sigma^2(\boldsymbol{\tau}_p)$. Similarly to definition (3.6) we compute $\sigma^2(\boldsymbol{\tau}_p)$ as

$$\sigma^2(\boldsymbol{\tau}_p) = \frac{1}{\#M_T} \sum_{t_i \in M_T} (\boldsymbol{\tau}_p(t_i) - \boldsymbol{\tau}_{\text{fine\_grid}})^2 \quad \text{for } M_T \equiv \{t_i | \, t_i \geq 10.0\} \tag{3.8}$$

where $\boldsymbol{\tau}_{\text{fine\_grid}}$ again denotes the numerical approximation on level $l = 6$. Note that it is necessary to change the denominator in (3.8) from M_T to $M_T - 1$ to obtain an unbiased variance estimator. The second last row of Table 3.3 specifies the variance of the approximation. Furthermore, the last row lists the order of convergence of $\sigma^2(\boldsymbol{\tau}_p)$. In this case, $p_l(\sigma^2)$ is analogously defined to (3.7). We observe for $p_l(\sigma^2)$ on different levels l that the results have a comparatively small variance with respect to the expected rate of 1.0.

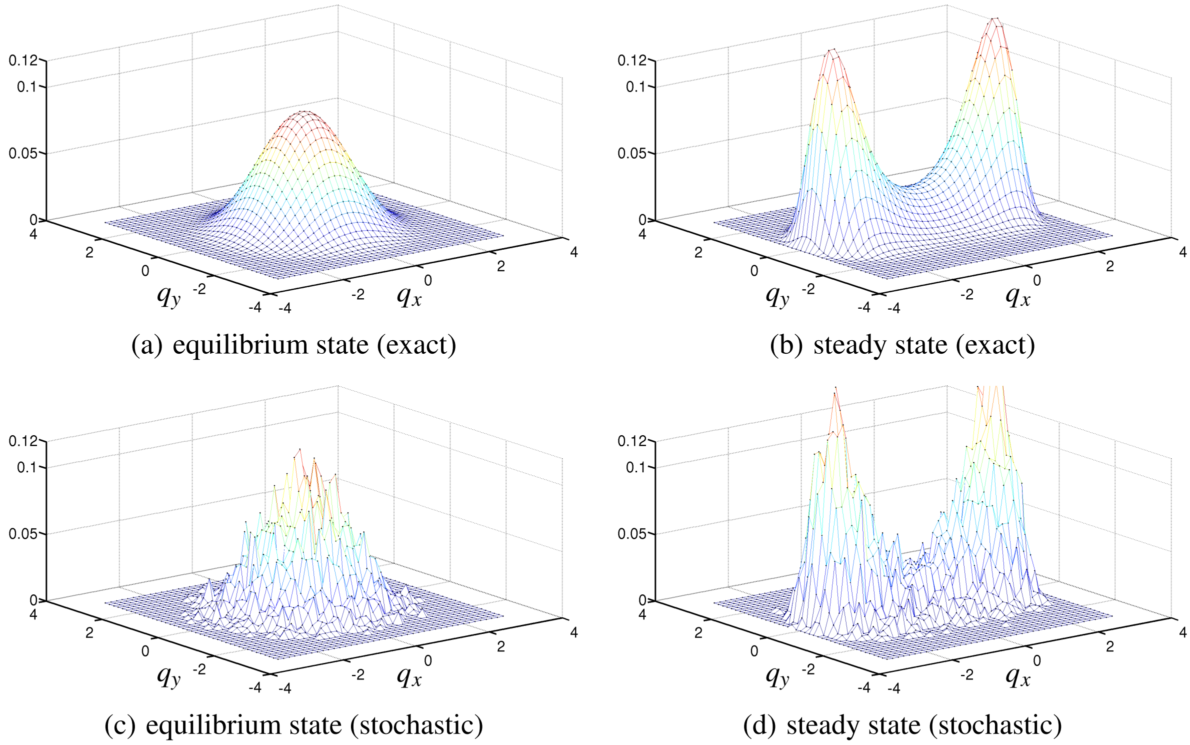
Finally, we conclude that the numerical scheme of Chapter 2 is able to obtain the expected rates of convergence for the mean and for the variance, respectively, for spring-chain models up to $N = 5$ segments. The next subsection focuses on the question how the stochastic approximation can be used to approximate the corresponding probability density function.

3.1.3. Reconstruction of the probability density function

An important difference between macroscopic approaches as described in Section 1.1 and multiscale approaches as considered in this thesis is that a multiscale description delivers additional information on the underlying polymeric structure. This information can be used to better characterize the dominant type of flow in a certain region of the flow domain. For this purpose, it is necessary to somehow visualize the distribution of the random variable $\boldsymbol{Q}$. In contrast to the Fokker-Planck based approach, see equation (1.36) in Section 1.2.2, the probability density function ψ is not directly computed in a stochastic approach. However, it is possible to recover the density ψ from the stochastic realizations. The corresponding numerical approach is known as *kernel density approximation*; see Parzen [101]. Usually, this approach suffers from a low order of convergence. On the other hand, it is sufficient to obtain a coarse approximation of ψ to get an idea of the dominant type of flow.

First, we try to reconstruct a three-dimensional probability density function for a FENE dumbbell model as considered in Section 3.1.1. In this case, we approximate $\psi : \mathcal{B}(0, b(1))^{1/2} \times [0, T] \to \mathbb{R}^+$, $(\boldsymbol{q}, t) \mapsto \psi(\boldsymbol{q}, t)$ for a moderate extensional flow with the parameters $De = 1.0$, $\dot{\epsilon} = 1.0$, $b = 10$ and $M = 10^5$.

In Fig. 3.5 we visualize a two-dimensional cut of ψ with $z = 0$ that is taken from Rüttgers [117]. The first row of Fig. 3.5 shows the analytic result at initial and at steady state. These two plots are visualizations of equation (1.56b) and (3.5). The equilibrium configuration at $t = 0.0$ has no dominant



(a) equilibrium state (exact)

(b) steady state (exact)

(c) equilibrium state (stochastic)

(d) steady state (stochastic)

Figure 3.5.: Comparison of a 2D cut from a simulated and a reconstructed 3D probability density function with the analytic solution. The density function evolves according to an extensional flow.

spring orientation since the density function is rotationally symmetric with respect to the origin. This situation has changed at steady state. Here, ψ is oriented along the flow direction x. The maximum of ψ is close to the maximum spring extension of $b^{1/2} \approx 3.2$. Therefore, an extensional flow stretches the molecules and rotates them in the direction of flow. We further note that the solution is point symmetric with respect to the origin since the orientation of the configuration vector $\boldsymbol{q}$ can be changed to $-\boldsymbol{q}$ without affecting the physical state of the system. The second row of Fig. 3.5 shows the reconstruction of ψ with a kernel density estimator by Botev et al. [16]. Although the stochastic simulation is performed with $M = 10^5$ stochastic realizations, only a few thousands samples have a z-coordinate close to zero and can be used to reconstruct the 2D cut of ψ at $z = 0$. This results in a coarse approximation of the probability density function. Nevertheless, it is still possible to characterize the type of flow as the general shape of ψ can be recognized.

Next, we analyze the polymer configurations for a more general spring-chain with $N > 1$ as considered in Section 3.1.2. In this case, the probability density function is a map $\psi : D_1 \times \cdots \times D_N \times [0, T] \rightarrow \mathbb{R}^+$, $(\boldsymbol{q}_1, \ldots, \boldsymbol{q}_N, t) \mapsto \psi(\boldsymbol{q}_1, \ldots, \boldsymbol{q}_N, t)$ with $D_i = \mathcal{B}(0, b(N)^{1/2}) \subset \mathbb{R}^3$ for $i = 1, \ldots, N$. Consequently, ψ has 15 dimensions in configuration space for the five segment case that was analyzed in Fig. 3.4 and in Table 3.3. Therefore, we cannot completely visualize the density function in a single plot but concentrate on marginal densities that represent individual spring segments.

In Fig. 3.6 we compare the components that belong to the spring segments $i = 1$ and $i = 3$ for 2000 samples in 15-dimensional space. This is a discrete approximation of the marginal densities $\psi|_{D_1} : D_1 \times [0, T] \rightarrow \mathbb{R}^+$ and $\psi|_{D_3} : D_3 \times [0, T] \rightarrow \mathbb{R}^+$. We further note that the third row of Fig. 3.6 shows the x-y-components of both segments only and skips the z-component. Since the equilibrium solution in (1.58b) at $t = 0$ is the N-fold tensor product of the dumbbell equilibrium density from (1.56b), we can use the same stochastic samples for each spring segment; see first column of Fig. 3.6. The situation changes with ongoing time. At $t = 2.0$ the stochastic samples in both spring segments have been stretched along the x-axis. Furthermore, most of the samples are close to their maximum extension $b(N)^{1/2} \approx 4.9$. Nevertheless, we still detect stochastic samples which are not extended at all. Interestingly, the samples for spring segment $i = 3$ (red color) seem to be more extended than the samples for segment $i = 1$ (blue color) as shown in the third row of Fig. 3.6. This effect is more pronounced at $t = 10.0$ close to the steady state. The physical interpretation of this finding is that the spring segments in the center of the spring-chain are better aligned with the flow and that there is a higher degree of freedom for the spring segments at the end of the chain.

The different distributions of these two segments of a five segment spring-chain can be better compared with the corresponding probability density functions. As seen in Fig. 3.6, we can only visualize two-dimensional cuts of the marginal distributions $\psi|_{D_1}$ and $\psi|_{D_3}$ for the random variables $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_3$ or alternatively visualize some specific 3D contour lines. To avoid this limitation, we instead consider the random variables $||\boldsymbol{Q}_1||^2$ and $||\boldsymbol{Q}_3||^2$ and try to reconstruct the corresponding densities. An approach is to construct a histogram of the 2000 samples in Fig. 3.6 and then normalize the result. The outcome is shown in Fig. 3.7. Here, we compare the densities of the squared extension of both spring segments $i = 1$ (blue solid line) and $i = 3$ (red solid line). We obtain similar results as in Fig. 3.6. The probability density function for $i = 3$ has its peak position at ≈ 21 which is close to the maximum extension of $b(5) = 24$. Furthermore, the density for $i = 3$ is more localized.

As a result, the reconstruction of the density function ψ gives valuable insights into the flow behavior. This becomes important in the following section on complex multiscale flows where regions with different dominant flow field types occur.

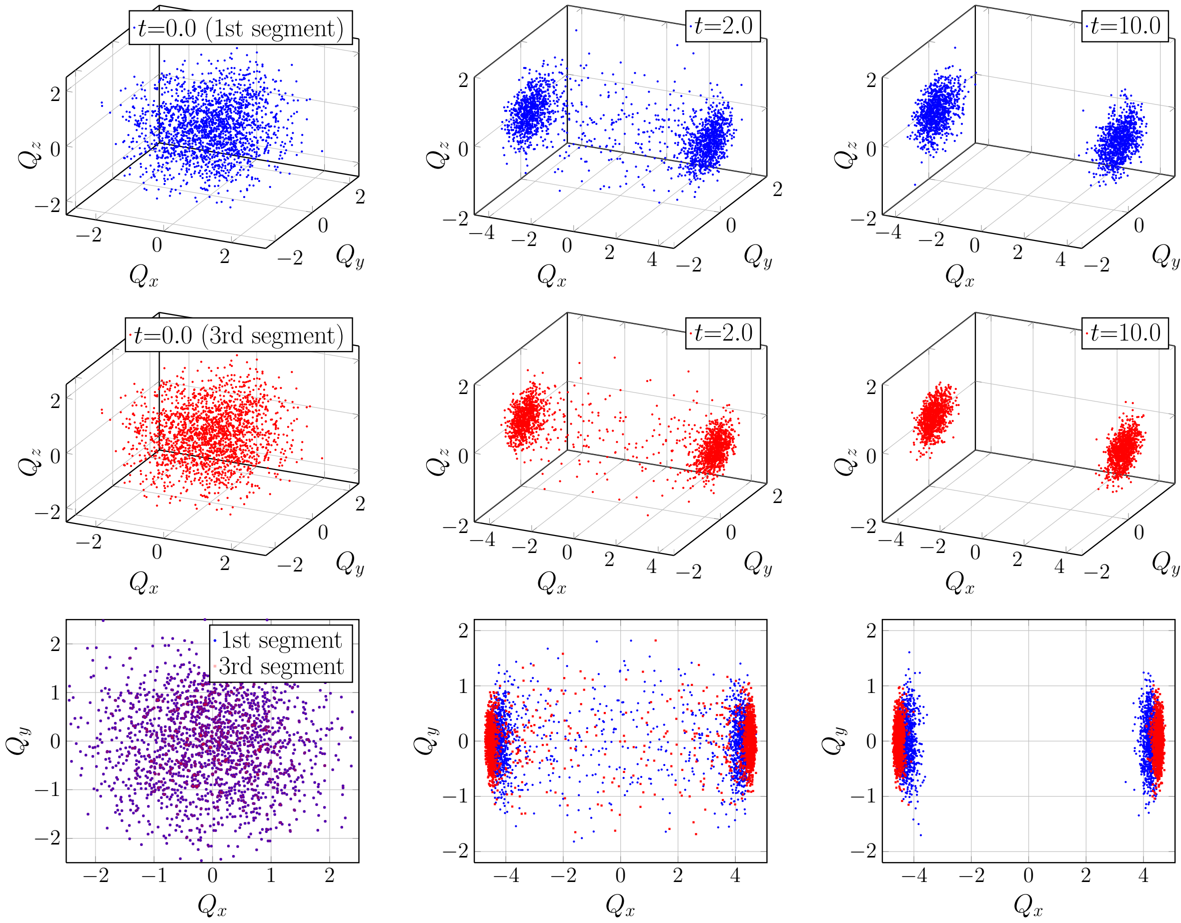


Figure 3.6.: Comparison of the discrete polymer configurations for the segments $i = 1$ and $i = 3$ of a 5-segment chain in an extensional flow. The third row compares the x/y-components of both segments.

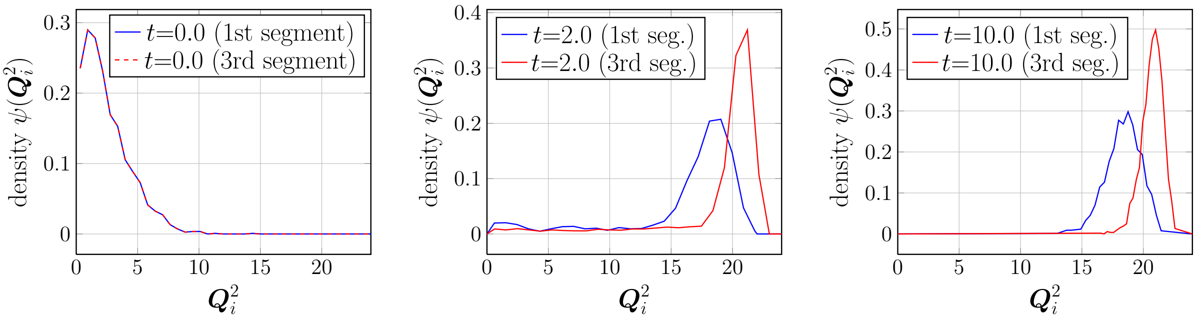


Figure 3.7.: Comparison of the $\boldsymbol{Q}_i^2$ density functions for the spring segments $i = 1$ (blue) and $i = 3$ (red) of a 5-segment chain.

3.2. Complex multiscale flows

A complex multiscale flow problem denotes the case in which the coupled system (1.59)–(1.63) is considered, i.e. the macroscopic Navier-Stokes equations are coupled with a stochastic PDE for the underlying polymeric structure. In contrast to homogeneous flows in Section 3.1, the spring segment variable $\boldsymbol{Q}$ now represents a time-dependent random field $\boldsymbol{Q} = \boldsymbol{Q}(\boldsymbol{x}, t)$.

In the following, we present numerical results for different complex flows. These are

- a planar contraction flow and
- a $4 : 1$ square-square contraction flow.

First, we consider a planar contraction flow for the multiscale FENE dumbbell model $N = 1$ in Section 3.2.1. Here, we reuse the flow parameters from previous simulations in the literature and are able to compare the simulation outcomes.

In Section 3.2.2 we present numerical results for a $4 : 1$ square-square contraction flow. For this purpose, we employ the multiscale FENE spring-chain model with $N = 1, 3, 5$ spring segments. In this case, we compare our results with experimental measurements from the literature.

Furthermore, we note that we have validated our multiscale viscoelastic flow solver in Griebel and Rüttgers [51] by comparing the simulation outcome with the analytic solution for a benchmark problem and by determining the rate of convergence. A non-Newtonian flow problem for which an analytic solution is known is, for instance, an *unsteady Poiseuille flow of a Hookean dumbbell fluid*. For this problem, an analytic solution for the velocity field was first derived by Waters and King [136]. Furthermore, Carew et al. [23] found an analytical solution for the stress tensor components.

3.2.1. Planar contraction flow

Problem specification and choice of simulation parameters

The first multiscale flow problem is a $4 : 1$ contraction flow in a two-dimensional channel that we visualize in Fig. 3.8. Simulation results for this type of flow have been published, for instance, by Wapperom, Keunings and Legat [133] for the FENE and FENE-P model with $N = 1$ spring segment. We reuse their simulation parameters to allow for comparisons. The results have been published in Griebel and Rüttgers [51] so that we focus on the most important findings. This simulation is of high practical relevance for the subsequent simulation in Section 3.2.2 since the 2D problem features the same corner singularity as the 3D case.

Fig. 3.8 shows the two-dimensional channel geometry. The length of the upstream and downstream channel is both 20.0. The upstream channel's height is 8.0 and reduces to a downstream channel with height 2.0 which leads to a contraction ratio of $4 : 1$. Furthermore, we select the fluid parameters in accordance with Wapperom et al. [133] as $\rho = 0.098$, $\eta_s = 0.05$, $\eta_p = 0.65$ and $\lambda = 0.6$. On the upper and low channel walls we prescribe no-slip boundary conditions. Furthermore, we assume a parabolic velocity profile on the inflow and outflow domain. With $U_c = 23/3$ as average outflow velocity, L_c as half of the channel height and the definitions in (1.10) we obtain the following dimensionless units to characterize the flow: Reynolds number $Re \approx 1.07$, Deborah number $De = 4.6$ and viscosity ratio $\beta = 1/14$. Furthermore, the FENE and FENE-P model both require a maximum spring extensibility which is chosen as $b = 10$.

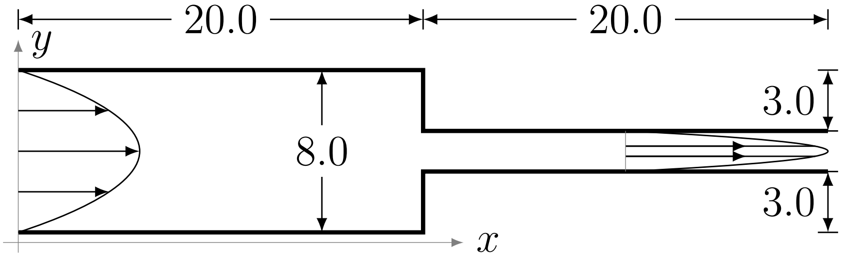


Figure 3.8.: Two-dimensional contraction flow geometry.

Table 3.4.: Mesh width on different levels l used for the FENE and FENE-P simulations with $N = 1$ spring segment.

l	Δx_l , Δy_l	Δz_l	cells/direction	total cells
1	0.5	0.5	$80 \times 16 \times 3$	3,840
2	0.25	0.5	$160 \times 32 \times 3$	15,360
3	0.125	0.5	$320 \times 64 \times 3$	61,440
4	0.0625	0.5	$640 \times 128 \times 3$	245,760

In Table 3.4 we list four different mesh resolutions $l = 1, 2, 3, 4$ for the following simulation. As mentioned in Chapter 2 the flow solver NaSt3DGPF only allows three-dimension grids. For two-dimensional simulations the three-dimensional grid is built with periodic boundary conditions in the z-direction to emulate a two-dimensional domain.

The simulations with the FENE and FENE-P model both use the discrete multiscale scheme described in Algorithm 2 in Section 2.2.2 with $M = 4000$ Brownian configuration fields. We note, however, that the FENE-P model is actually a macroscopic model. Therefore, the same results as with Algorithm 2 could be achieved with a purely macroscopic approach. Nevertheless, we use the FENE-P model in its multiscale formulation to allow for a better comparability with the FENE model since all discrepancies result from the modeling and not from the implementation differences.

The temporal discretization of the Navier-Stokes equations is performed with a semi-implicit Crank-Nicolson scheme; see Section 2.2.2. The maximum time-step width that ensures stability depends on the grid level l. The time-step width is of order $O(10^{-2})$ for $l = 1$ and of order $O(10^{-3})$ for $l = 4$. Finally, we summarize the essential simulation parameters in Table 3.5.

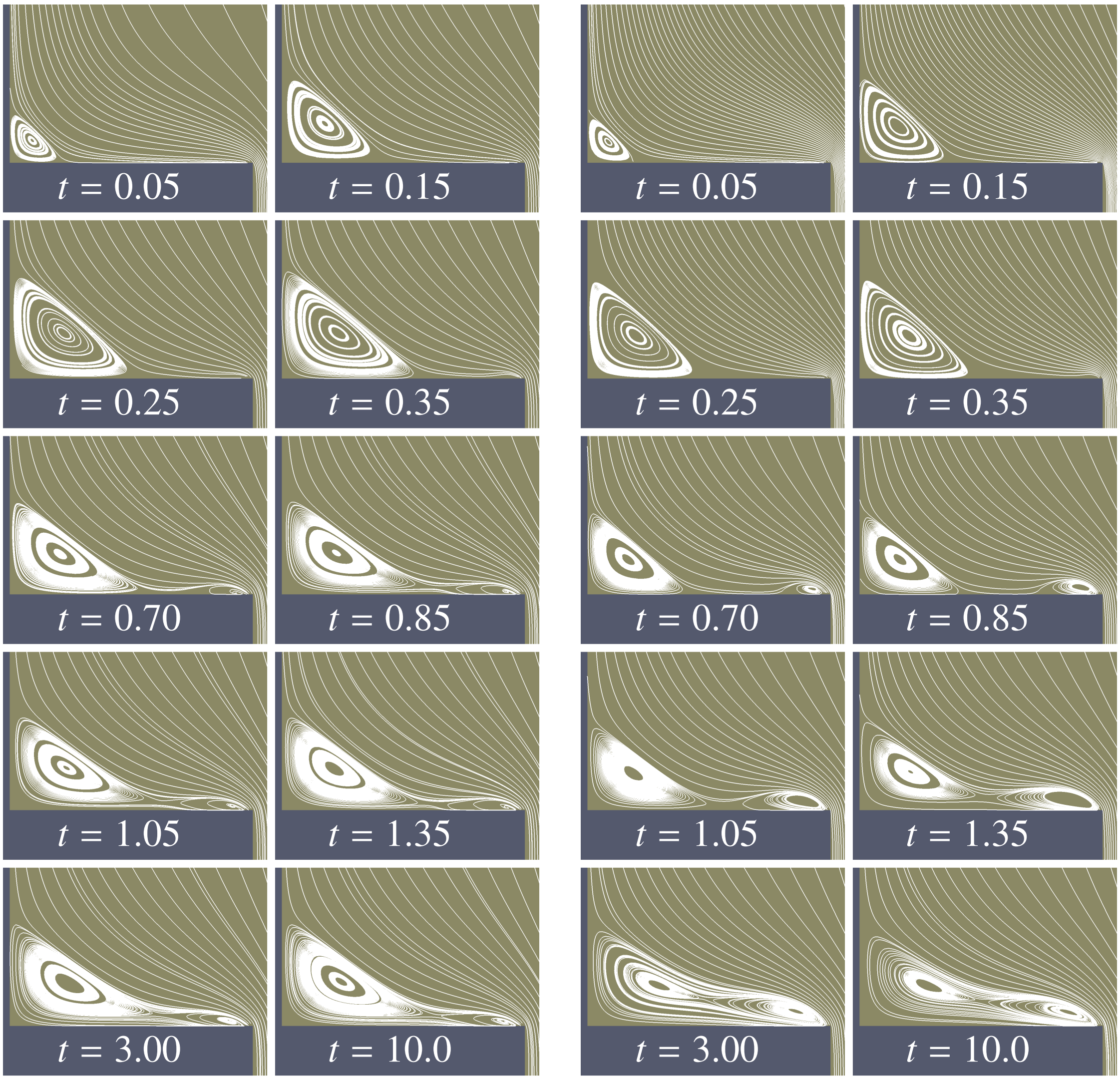
Planar contraction flow with dumbbell models		
Reynolds number	Re	1.07
Deborah number	De	4.6
viscosity ratio	β	1/14
maximum spring extension	b	20
number of segments	N	1 (dumbbell)
stochastic samples / cell	M	4000
temporal scheme (macro)	Crank-Nicolson	

Table 3.5.: Parameters for a planar contraction flow with a FENE and FENE-P dumbbell model.

2D simulation results

In Fig. 3.9, the main result, we compare the temporal evolution of the lower corner vortices for the FENE model (left) and FENE-P model (right) on the finest grid level $l = 4$. Furthermore, the points in time for which the vortices have been visualized are chosen such that they coincide with the results in Figure 15 and Figure 20 in Wapperom et al. [133]. These two figures are reprinted in Fig. 3.10 in this thesis.

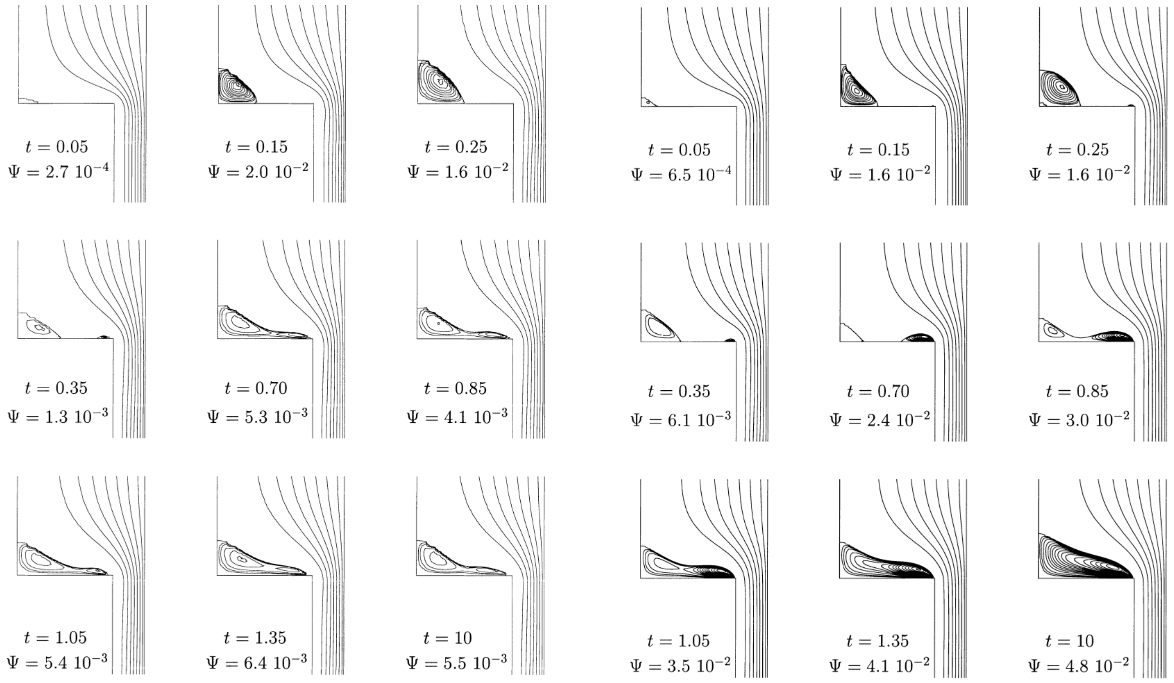
Both models predict a small initial corner vortex that starts to grow over time. At about $t = 0.35$ the corner vortices have reached their maximum size and a lip-vortex at the re-entrant corner forms. Then,



(a) Evolution of the FENE model

(b) Evolution of the FENE-P model

Figure 3.9.: Comparison of the corner vortex evolution over time for the FENE and the FENE-P dumbbell model on grid level $l = 4$. See Fig. 3.10 for a comparison with the literature result.



(a) Evolution of the FENE model as shown in Figure 20 of Wapperom et al. [133]

(b) Evolution of the FENE-P model as shown in Figure 15 of Wapperom et al. [133]

Figure 3.10.: Vortex evolution for the FENE and FENE-P models according to the literature result.[1]

the further evolution differs between the FENE and the FENE-P model.

At $t \approx 0.7$, the lip vortex in the FENE model attaches to the corner vortex. Next, the vortex intensity of the combined structure further increases up to $t \approx 3.0$ when the system reaches a steady state. Although a combined vortex structure is formed, both vortex cores can still be separated during the complete simulation.

On the other hand, the lip vortex that is predicted in the FENE-P model is larger in size than in the FENE model. Interestingly, the vortices in the FENE-P model are separated for a longer period of time up to about $t = 0.85$. Then, they build one larger vortex structure that differs in vortex intensity and shape from the FENE model. Although both vortex cores can also be separated in the FENE-P model, they seem to be more concentrated than in the FENE model. This is due to the fact that the interior region between the vortex cores in the FENE-P model has a much higher intensity. Furthermore, the corner vortex at steady state has a smaller size than in the FENE model. The FENE-P model also reaches its steady state at $t \approx 3.0$.

Compared to the literature, our FENE simulation results show an excellent agreement with the results in Figure 20 in Wapperom et al. [133] (reprinted in Fig. 3.10 (a) in this thesis). But, our results for the FENE-P model differ from the literature result at the final stage of the simulation. Wapperom et al. [133] observe that the two separate vortex cores in the FENE-P model merge at $t \approx 1.35$ and form one large lip vortex; cf. Fig. 3.10 (b).

[1] Reprinted from Journal of Non-Newtonian Fluid Mechanics, 91 (2), Wapperom, Keunings and Legat, *The backward-tracking Lagrangian particle method for transient viscoelastic flows*, 273–295, 2000, with permission from Elsevier.

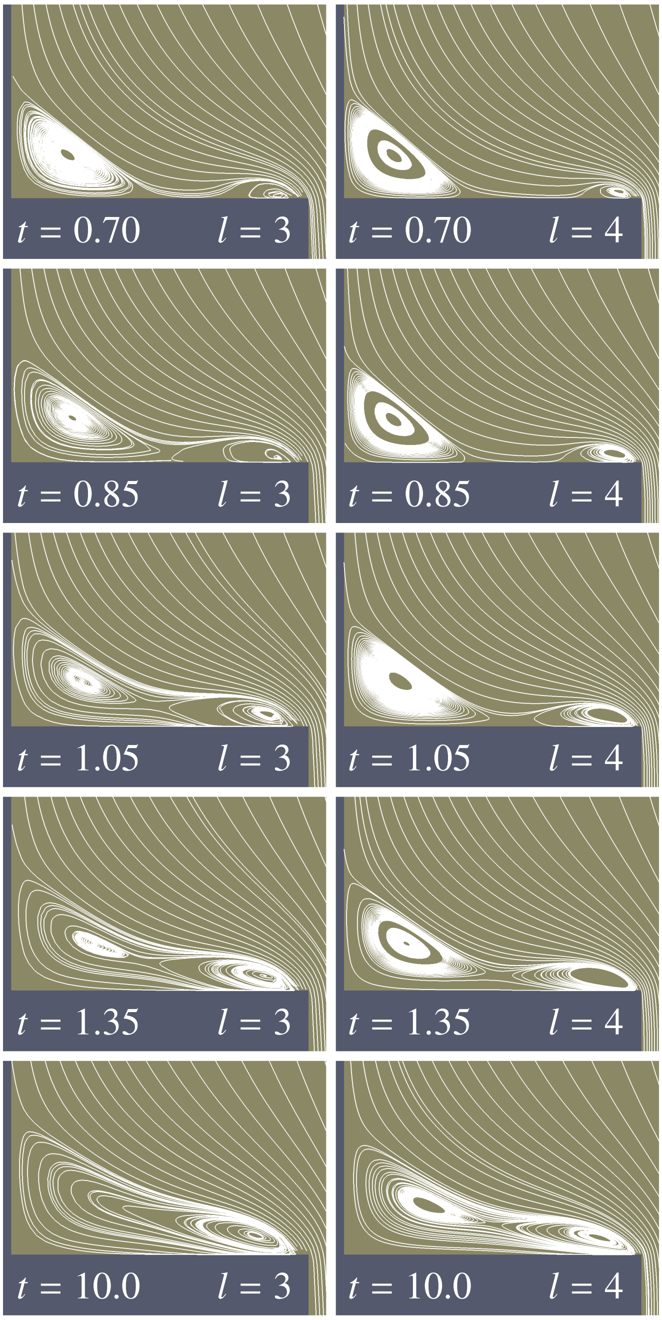


Figure 3.11.: Numerical result for the FENE-P dumbbell model depending on the mesh width l.

An explanation for this difference in the FENE-P model between our simulation and the findings by Wapperom et al. [133] can be found in Fig. 3.11. In this figure, we compare the temporal evolution of the FENE-P model on our finest grid $l = 4$ (right) and on the next coarser grid with level $l = 3$ (left). The results on both grids are nearly identical up to $t \approx 0.85$. Then, at $t \approx 1.35$ the two vortex cores on $l = 3$ merge and build one large vortex structure. Consequently, the coarser grid does not seem to be

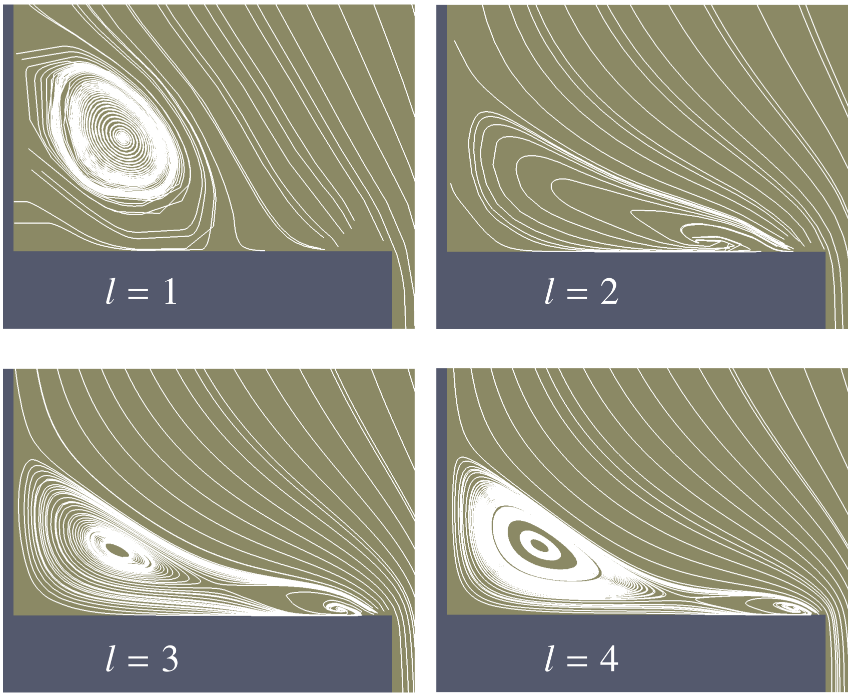
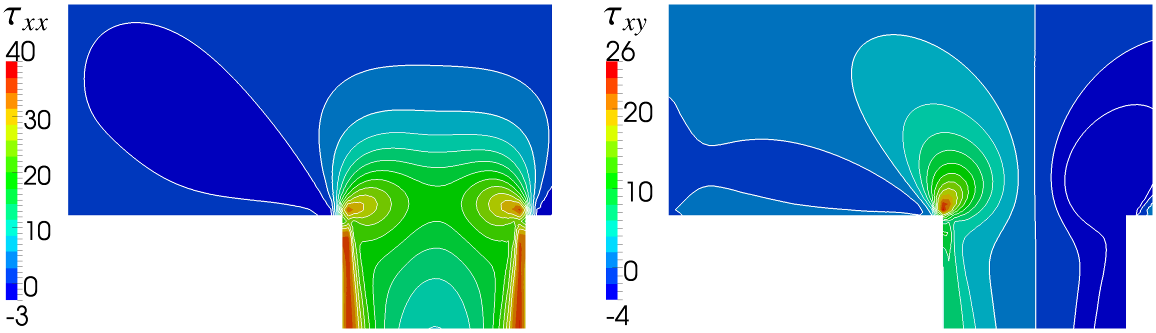


Figure 3.12.: Steady state corner vortex for the FENE dumbbell model depending on the mesh width l.

sufficient to resolve both vortex cores. Interestingly, the result for $l = 3$ closely resembles the literature results by Wapperom et al. [133] that we have reprinted in Fig. 3.10 (b) in this thesis. Furthermore, the authors employ an adaptive mesh that is relatively fine close to the corner singularity but coarse otherwise. For this reason, it seems plausible that the grid used by Wapperom et al. [133] is fine enough to resolve the lip vortex core close to the contraction corner but not fine enough to resolve the core of the corner vortex. Note that we have also performed a FENE-P simulation on the next finer mesh level $l = 5$ with $1280 \times 256 \times 3$ grid cells. The steady state result is very close to the result on level $l = 4$. This verifies that our result on level $l = 4$ is grid independent. We do not show the level $l = 5$ as it does not reveal any further vortex characteristics.

The results for the FENE-P model indicate that with an insufficient grid resolution the simulation outcome strongly varies with the mesh width. Fig. 3.12 gives a similar analysis for the FENE dumbbell model at $t = 10.0$ (steady state). On the coarsest grid level $l = 1$ the mesh only resolves one large corner vortex. A lip vortex does not occur. The next grid $l = 2$ is still unable to resolve both vortex cores. Nevertheless, the shape of the vortex structure already resembles the fine grid results. Finally, the finest grids $l = 3$ and $l = 4$ are both able to show a separate lip and corner vortex. Although there are still minor differences in the vortex shapes, the grid on level $l = 3$ is able to resolve all important vortex characteristics. Therefore, at least for the FENE model a grid accuracy on level $l = 3$ seems to be sufficient. This is important as the resolution of the subsequent 3D simulations in Section 3.2.2 resemble the grid solution on $l = 3$. Our 2D grid convergence study indicates that this resolution seems to be sufficient to capture the relevant flow phenomena.

Finally, in Fig. 3.13 we display the first normal stress component τ_{xx} and the first shear stress component τ_{xy} at $t = 10.0$ (steady state). Although the solution is symmetric with respect to the channel's



(a) Contour lines of stress component τ_{xx} (b) Contour lines of stress component τ_{xy}

Figure 3.13.: Steady state stress components for the FENE dumbbell model.

centerline, we nevertheless show a slightly larger part of the simulation domain. This illustrates the accuracy of our numerical stress tensor scheme that ensures the symmetry of the solution even though it depends on stochastic samples. On a larger scale there are stochastic oscillations of the stress tensor over time. For fixed time, however, hardly any oscillations are visible for the stress tensor. This is a result of the Brownian configuration field approach that is described in Chapter 2.2.

The perspective in Fig. 3.13 was chosen such that it coincides with Figure 19 in the article by Wapperom et al. [133]. Both stress components have their maximum at the corner singularity. We note that the numerical scheme in NaSt3DGPF does not place grid points in the cell corners. As a consequence, this leads to a slightly rounded corner geometry. Therefore, the τ_{xx} stress component at the corner becomes the larger the more the mesh level l is increased since the singularity is better approximated. The contour lines of both stress components closely resemble the results obtained by Wapperom et al. [133]. The only difference we note is that the maximum normal stress values are slightly larger in the literature result. We explain this difference by the adaptive mesh geometry refinement at the corner singularity. This difference diminishes with further mesh refinement.

As a result, we conclude for our multiscale flow solver:

- The simulated corner vortex structure in a two-dimensional contraction flow with a FENE dumbbell model closely resembles the literature results; cf. Fig. 3.9 (left) with Figure 20 in Wapperom et al. [133]. Note that the literature result is reprinted in Fig. 3.10 (a) in this thesis.

- The macroscopic FENE-P closure delivers only a coarse approximation to the FENE spring force model; see vortex structure in Fig. 3.9. This verifies that multiscale simulation approaches deliver insights into the micro-scale that cannot be approximated by macroscopic methods.

- Our simulation results for the FENE-P model differed with the results in Wapperom et al. [133]; cf. Fig. 3.10 (b). But, our simulation results indicate that these differences result from a coarse mesh in Wapperom et al. [133] in the region far away from the contraction corner. This was derived from the finding that our result on the coarse grid level $l = 3$ agrees with the literature result; see Fig. 3.11.

3.2.2. $4:1$ square-square contraction flow

Problem specification and choice of simulation parameters

In this section, we extend the two-dimensional contraction flow simulations from Section 3.2.1 to its three-dimensional analogon. Due to the enormous increase in computational complexity for three-dimensional multiscale polymer simulations, there exist no further multiscale simulation results for this three-dimensional flow problem in the literature.

In Griebel and Rüttgers [51] some first-time results for this problem have been published. The following results extend these findings in the following aspects:

- We consider the general FENE spring-chain with $N = 1, 3, 5$ spring segments. The simulation in Griebel and Rüttgers [51] concentrates on the FENE dumbbell model with $N = 1$ only.
- We analyze the effect of different maximum spring extensibility parameters b. In Griebel and Rüttgers [51] only simulations with $b = 20$ were conducted.
- We employ more grid cells to discretize the 3D flow space and more stochastic samples to approximate the $3N$-dimensional configuration space as in Griebel and Rüttgers [51]. For this reason, the simulation domain includes a longer downstream channel to allow for an analysis of the stress relaxation.

Although we are not aware of comparable simulation results, there exists various experimental measurements for square-square contraction flows in the literature. In the following, we employ measurements published by Sousa et al. [123] for a non-Newtonian solution composed of 40.0% (w/w) glycerol, 59.9% (w/w) water and polyacrylic acid (PAA) with a weight concentration of 600 ppm for comparison. This publication extends previous results from Sousa et al. [122] for a Newtonian and a Boger fluid to shear-thinning fluids. The reason to decide for this literature result is that shear-thinning fluid is considered and that the FENE model is able to describe such fluids.

Fig. 3.14 visualizes the simulation domains which will be used in the following. Depending on the model parameters, we decide for one of two similar simulation domains (a) or (b). In both cases, the flow geometry consists of a quadratic upstream channel with side length $2H_1 = 24$ mm which contracts into a smaller downstream channel with side length $2H_2 = 6$ mm, i.e. the contraction ratio is 4:1. The

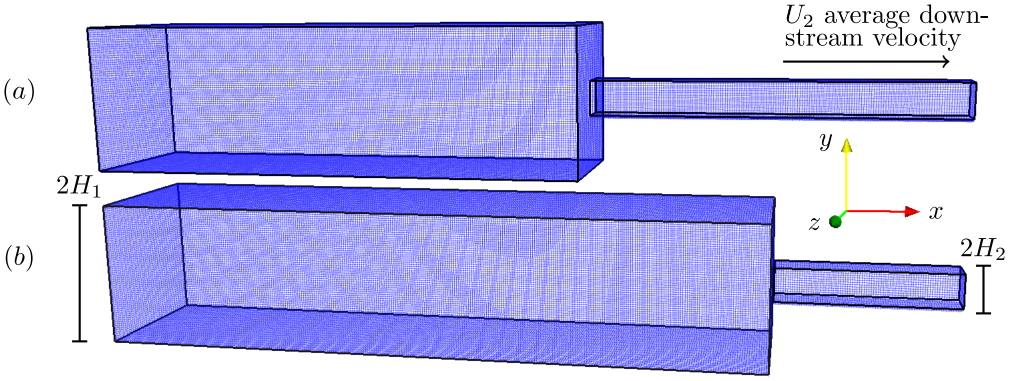


Figure 3.14.: Visualization of the 3D meshes for the contraction flow problem.

Three-dimensional contraction flow with spring-chain models		
solvent shear viscosity	η_s	0.03 [Pa s]
zero polymer shear viscosity	η_p	1.62 [Pa s]
fluid's relaxation time	λ	32 [s]
Reynolds number	Re	$\ll 1.0$
Deborah number	De	$24.1, 108, 157$
viscosity ratio	β	0.0182
spring models	FENE-chain	
maximum spring extension	b	20, 60, 120
spring segment nr.	N	1, 3, 5
spring extension gauge	$b(N)$	de Gennes (1.48)
spring relaxation gauge	$\lambda(N)$	rel. time scale (1.51)
number of grid cells		$380 \times 64 \times 64$
stochastic samples / cell	M	up to 1200
total sample nr.		up to $\approx 1.87 \cdot 10^9$
memory requirement		up to $\approx$ 209 GB
temporal scheme (macro)	Crank-Nicolson	

Table 3.6.: Parameters for a 3D contraction flow simulation with FENE spring-chain models.

length of the upstream channel in domain (a) is 82.5 mm. The corresponding downstream channel's length is 60 mm. This slightly differs for the other simulation domain (b) shown in the lower part of Fig. 3.14. Here, the lengths of the upstream and downstream channels are 112.5 mm and 30 mm, respectively. In total, the length of the simulation domain in x-direction is 142.5 mm in both cases.

The reason for these two different simulation domains becomes clear from the simulation parameters that we list in Table 3.6. The simulations are conducted with maximum spring extensibility values $b = 20, 60, 120$. Usually, the larger the spring extensibility b is chosen the larger is the stress tensor $\boldsymbol{\tau}_p$ in the model. Since larger normal stresses often lead to stronger corner vortices in the upstream channel, we try to maximize the upstream channel's length for $b = 120$ without increasing the computational complexity. Therefore, we use the simulation domain (b) for $b = 120$. Since we do not expect corner vortices with a comparable size for $b = 20$ and $b = 60$, it is beneficial to spend more grid points on the downstream channel. Therefore, we perform the simulations with $b = 20$ and $b = 60$ on the simulation domain (a) as shown in Fig. 3.14.

Both computational domains are discretized on a grid with $380 \times 64 \times 64$ cells; see Table 3.6. In Section 3.2.1 we demonstrate that this resolution is sufficient to resolve the relevant vortex structures in the FENE model. For the stochastic description of the polymer models we employ $M = 1200$ samples in each cell. Consequently, without considering the boundary cells, the total number of Brownian configuration fields is $380 \times 64 \times 64 \times 1200 \approx 1.87 \cdot 10^9$. This leads to an enormous computational complexity especially for the simulation with $N = 5$ spring segments. The samples for a Brownian configuration field of dimensionality $3N$ require $3N$ double variables which each have an 8 byte size. To sum it up, the memory requirement is $\approx$ 209 GB for $N = 5$ spring segments and about 125 GB and 42 GB for $N = 3$ and $N = 1$, respectively. For this reason, it is essential to use massively parallel computers to allow for simulations of this complexity on full grids, see Section 2.3, or perform the same simulations on sparse grids, see Chapter 6.

We choose the parameters of the flow in correspondence with those used by Sousa et al. [123]. Beside experimental measurements, these authors also perform macroscopic simulations with a simplified Phan-Thien Tanner model (SPTT) that is able to predict shear-thinnig. Sousa et al. [123] state the parameters for their fluid fit as $\epsilon = 0.06$ for the extensibility parameter, $\eta_p = 1.62\,\mathrm{Pa\,s}$ for the zero polymer shear viscosity, $\eta_s = 0.03\,\mathrm{Pa\,s}$ for the solvent shear viscosity, and $\lambda = 32\,\mathrm{s}$ for the fluid's relaxation time, respectively. Again, we transform these variables into their dimensionless form by using (1.10). In this case, we obtain $\beta = 0.0182$ for the viscosity ratio. Next, we decide for $L_c = H_2 = 3\,\mathrm{mm}$ as a characteristic length scale and for $U_c = U_2$, with U_2 as average downstream velocity, as a characteristic velocity. With $U_2 = 2.26\,\mathrm{mm/s}, 10.1\,\mathrm{mm/s}$ and $14.2\,\mathrm{mm}/s$ and (1.10) we then obtain the Deborah numbers $De = 24.1, 108$ and 157. The corresponding Reynolds number are about $Re = 0.003, 0.014$ and $0.02 < 1$. Consequently, the low Reynolds number indicates that the occurrence of corner vortices for high flow rates are an effect of the fluid's elasticity, characterized by the Deborah number, and not due to inertia as characterized by the Reynolds number.

On the inflow boundary, we prescribe the analytical velocity field solution for a constant viscosity fluid as boundary condition; see White [138]. Theoretically, this boundary condition only applies to Newtonian fluids and viscoelastic fluids without shear-thinning. But, at a sufficient distance from the contraction corner, the velocity profile can be assumed as Newtonian even for shear-thinning fluids. Since our simulations do not show deviations from the inflow profile in the first part of the upstream channel, we conclude that this simplification does not strongly effect the behavior in the contraction region.

The FENE model contains one parameter, the maximum spring-chain extensibility b, to match the shear-thinning behavior of the experimental fluid. There is, however, no guarantee that the specific experimental fluid can be described in all details by the FENE model. Furthermore, we cannot directly evaluate the viscometric functions of the experimental fluid which is a fluid characterization in shear and extensional flows. But, since Sousa et al. [123] stated the extensibility of the SPTT model as $\epsilon = 0.06$, we can fit b such that it matches the SPTT model. For this purpose, we plot in Fig. 3.15 the viscosity $\eta(\dot\gamma) = \eta_s + \eta_p(\dot\gamma)$ and the first normal stress coefficient $\Psi_1 = (\tau_{xx} - \tau_{yy})/\dot\gamma$ over the shear rate $\dot\gamma$ for the 3D FENE dumbbell model with $b = 5, 10, 20, 50$.

In the first row of Fig. 3.15 the apparent viscosity $\eta(\dot\gamma)$ is visualized for different shear rates in the range $\dot\gamma \in [10^{-2}, 10^3]s^{-1}$. In the limit $\dot\gamma \to 0$, the apparent viscosity matches the specified zero shear result $\eta(\dot\gamma) = \eta_s + \eta_p = 1.65\,\mathrm{Pa\,s}$. For $\dot\gamma \to \infty$, the polymer viscosity $\eta_p(\dot\gamma)$ tends to zero so that all plots in Fig. 3.15 converge to the solvent's viscosity value $0.03\,\mathrm{Pa}\,s$. In the following simulations, however, we focus on shear rates in the range $[0.5 \cdot 10^{-1}, 10^1]s^{-1}$ since relevant shear rates $\dot\gamma$ are of order $\dot\gamma \approx U_2/H_2 = 0.75\,\mathrm{s}^{-1}$ for $De = 24.1$ and of order $\dot\gamma \approx U_2/H_2 = 4.9\,\mathrm{s}^{-1}$ for $De = 157$. For this reason, we show the corresponding range of $\dot\gamma$ in a zoomed extract on the right-hand side (RHS). Furthermore, this extract is indicated with a red rectangle on the left-hand side (LHS) of Fig. 3.15. We note that all four choices for b in the FENE dumbbell model deviate from the SPTT model prediction (black solid line) for $\eta(\dot\gamma)$. This outcome underlines that the FENE and the SPTT model used by Sousa et al. [123] differ in the prediction of the shear-rate dependent viscosity.

Since the analysis of the shear-thinning behavior leads to indecisive results for an estimate of the model parameter b, a further approach was used in Griebel and Rüttgers [51]. Here, a plot of the first normal stress coefficient defined as $\Psi_1 = (\tau_{xx} - \tau_{yy})/\dot\gamma^2$ is compared. We show corresponding plots in the second row of Fig. 3.15. In this case, the FENE model with $b = 20$ is in high accordance with the SPTT model fit for the relevant range of $\dot\gamma$. For this reason, all 3D contraction flow simulations in

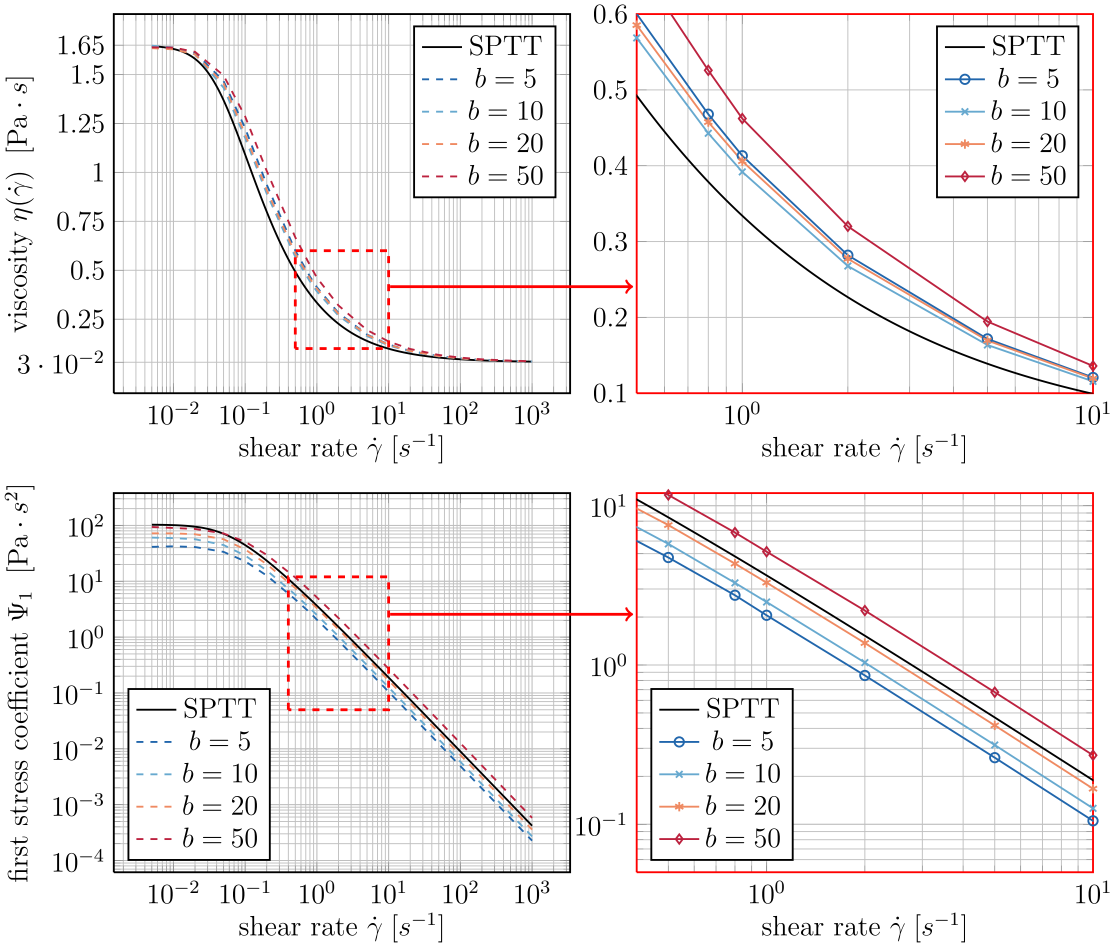


Figure 3.15.: Apparent viscosity and first stress coefficient Ψ_1 over shear rate $\dot{\gamma}$ for the SPTT and the 3D FENE dumbbell model with $b = 5, 10, 20, 50$.

Griebel and Rüttgers [51] were performed with this parameter choice.

One finding in Griebel and Rüttgers [51] was that the FENE dumbbell model with $b = 20$ accurately predicts the size of the corner vortices for low and medium Deborah numbers, compared to the experimental measurements in Sousa et al. [123], but underestimates the vortex size for high Deborah number flows. Furthermore, a comparison with the SPTT model was not possible in Griebel and Rüttgers [51] since Sousa et al. [123] report stability issues for higher Deborah numbers. These stability issues did not occur for the multiscale Brownian configuration field approach in Griebel and Rüttgers [51] and motivated the use of this computationally demanding approach.

In this thesis, we investigate other choices for the maximum spring extensions b as in Griebel and Rüttgers [51] to better match the experimentally measured vortex size. More precisely, we perform simulations with the values $b = 20$, $b = 60$ and $b = 120$ even though Fig. 3.15 indicates that $b = 20$ gives the best match to the fluid in a shear flow. One explanation for this approach is that we expect an extensional flow-type behavior on the centerline. Sousa et al. [123] do not deliver a characterization of the fluid in an extensional flow. Therefore, it does make sense to also investigate other choices for

b. Usually, the stress tensor grows with the molecular extension that is characterized by b. Since the corner vortices are an effect of the normal stress, an increase in b should lead to larger corner vortices. Nevertheless, we decide for $b = 120$ as largest possible extension in the following simulations for two reasons:

- Herrchen and Öttinger [59] analyzed various FENE dumbbell and closure models with extensions $b = 20, 50, 100$. These authors give a physical meaning to b interpret this values roughly as the number of monomer units that build the polymer molecule. They conclude that $b \in [20, 100]$ leads to a model that corresponds with the physical representation of a real polymer.
- As mentioned before, for $b \to \infty$ the FENE spring converges to the Hookean spring model. Since a Hookean spring-chain molecule (Rouse model) can be fully described by a macroscopic model, see Section 1.1, a multiscale approach is not required. We therefore concentrate on choices for b in which the FENE model differs from the Oldroyd-B model.

3D simulation results

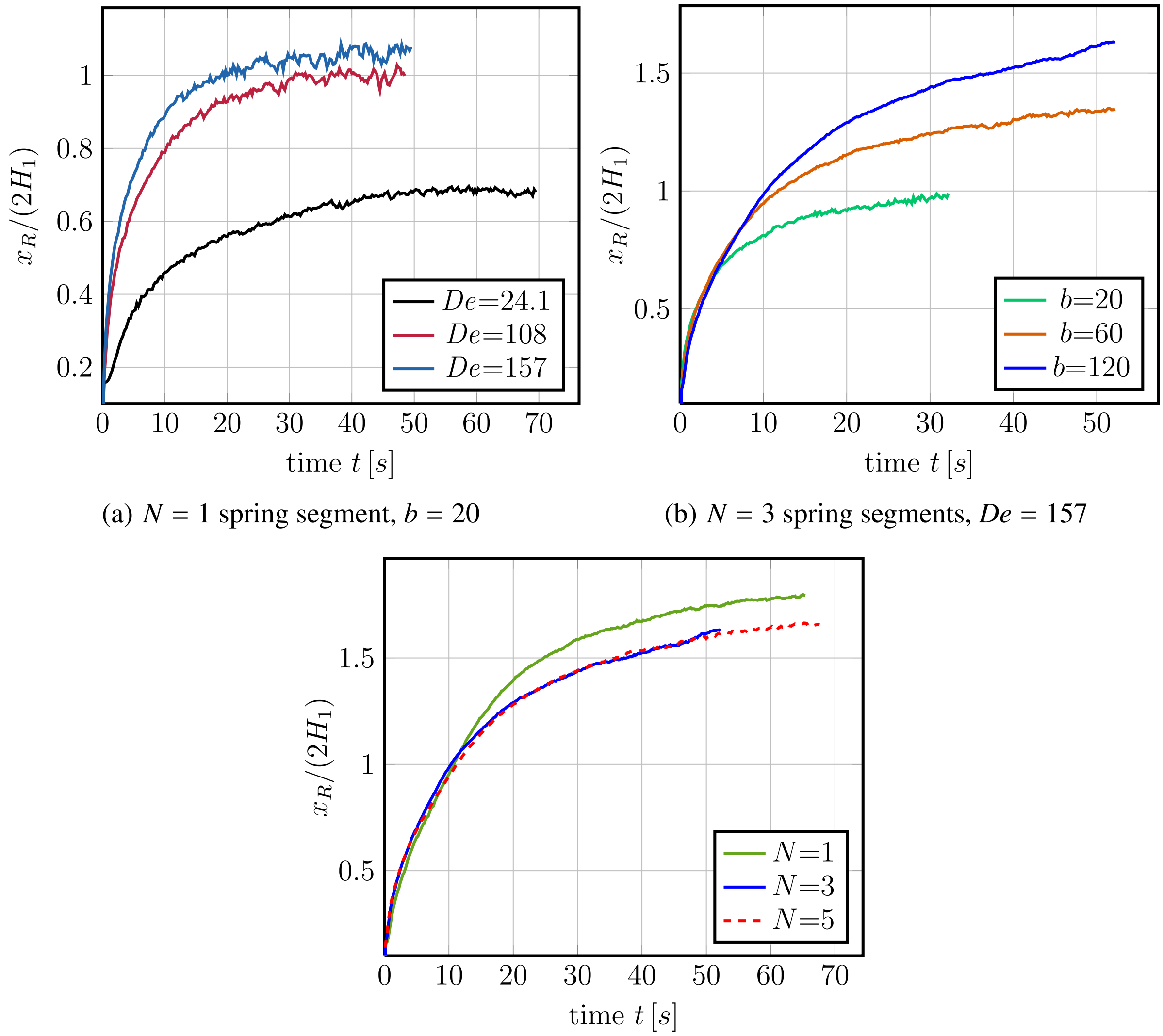
As listed in Table 3.6, we consider contraction flows with different flow rates characterized by the Deborah number ($De = 24.1, 108, 157$), flows with different spring segments numbers ($N = 1, 3, 5$) and with different maximum spring extensibility parameters ($b = 20, 60, 120$). Since each multiscale simulation requires massively parallel computations, we do not simulate all possible 27 parameter combinations but restrict ourselves to 8 representative simulations. These simulations use the parameters

- $De = 24.1, N = 1, b = 20$,
- $De = 108, N = 1, b = 20$,
- $De = 157, N = 1, b = 20$,
- $De = 157, N = 1, b = 120$,
- $De = 157, N = 3, b = 20$,
- $De = 157, N = 3, b = 60$,
- $De = 157, N = 3, b = 120$,
- $De = 157, N = 5, b = 120$

and the remaining flow parameters as specified in Table 3.6.

In all eight simulations a vortex occurs in the corner regions of the upstream channel and starts to grow over time, compare with Section 3.2.1 for the two-dimensional analogon. We plot the size of the corner vortices for the three dumbbell simulations with $N = 1$ in Fig. 3.16 (a). This plot resembles Fig. 22 in Griebel and Rüttgers [51]. We observe that the size of the $2H_1$-normalized corner vortex increases with ongoing time and reaches its steady state value at $t \approx 50\,s$ for $De = 108$, 157 and at $t \approx 70\,s$ for $De = 24.1$. Furthermore, the size of the corner vortex increases with De. The steady state results for the $2H_1$-normalized corner vortex size are ≈ 0.7 for $De = 24.1$, ≈ 1.0 for $De = 108$ and ≈ 1.1 for $De = 157$. The longer period of time for the simulation with $De = 24.1$ to reach a steady state can be comprehended with a physical interpretation of the polymer system. Since larger Deborah numbers correspond with higher flow rates in the channel, the spring-chain molecules are extended much faster in a shorter period of time. This explains the immediate increase of τ_{xx} for $De = 108$ and $De = 157$ in Fig. 3.16 (a) compared to the flow with $De = 24.1$.

Fig. 3.16 (b) shows the temporal evolution of the corner vortex sizes for the 3-segment spring-chain ($N = 3$). In this case, we concentrate on the highest Deborah number but vary the maximum spring extensibility b. As expected, the normal stresses increase with b and the higher elasticity leads to larger corner vortices. In all cases, the 3-segment chains have not yet fully reached their steady state. The



(a) $N = 1$ spring segment, $b = 20$

(b) $N = 3$ spring segments, $De = 157$

(c) $b = 120$, $De = 157$

Figure 3.16.: Comparison of the $2H_1$-normalized vortex length on the central plane for different Deborah numbers, spring extensibility b and number of spring segments N.

reason for this limitation is the enormous computation time for such a simulation in the order of weeks. As a preliminary result, the $2H_1$-normalized corner vortex size is about 1.0 at $t = 30\,s$ for $b = 20$ and about 1.35 and 1.6 for $b = 60$ and $b = 120$, respectively. Interestingly, the plot of the τ_{xx} stress component in Fig. 3.16 (b) for $N = 3$ has a lower variance than the corresponding plot in Fig. 3.16 (a) for $N = 1$. There are two explanations for this observation. First, as mentioned before, the simulations with $N = 1$ use $M = 800$ samples per grid cell instead of $M = 1200$ for $N = 3$ and $N = 5$. Second, the stress component for $N = 3$ is the average of the three spring-segment stress components; cf. (1.62) for the stress tensor definition. This averaging further reduces the variance of the solutions for $N = 3$ and for $N = 5$.

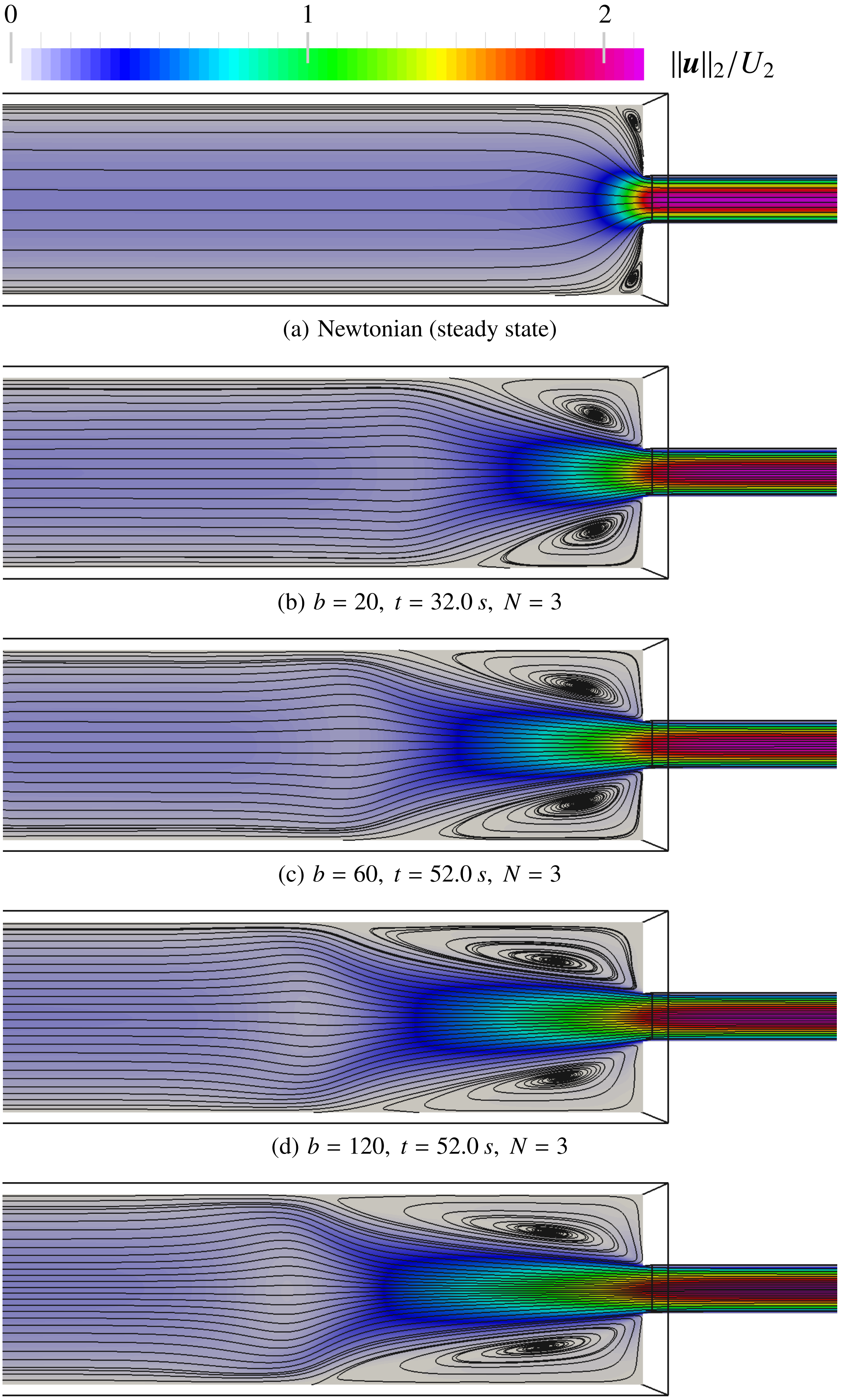
At last, in Fig. 3.16 (c) we compare the vortex growth depending on the spring-segment number N. Here, all simulations use the highest flow rate with $De = 157$ and have a maximum spring extension of $b = 120$. Up to $t = 10\,s$, the three simulations are almost identical and predict a corner vortex size of ≈ 1.0. Then, the situation differs between the dumbbell model on the one hand and the 3-segment and 5-segment chain on the other hand. Interestingly, the dumbbell model predicts a larger corner vortex of about 1.7 at $t = 40\,s$ compared to about 1.5 for $N = 3$ and $N = 5$. In contrast to this, both spring-chain models show a high agreement in the predicted vortex size. The simulation with the 5-segment chain is conducted up to $t \approx 68\,s$ with a final vortex size of about 1.7 which matches the result of the dumbbell system at $t \approx 40.0\,s$. Our results clearly indicate a substantial difference between a dumbbell and a spring-chain model. Up to now, however, it is not clear which model is more accurate even though the multi-segment model more closely resembles a real polymeric molecule.

Next in Fig. 3.17, we compare the steady state results of several selected simulations. For a better visualization of the flow patterns, we project the three-dimensional streamlines onto a slice that goes through the channel center. The projection is performed with the *Surface Vectors* filter in the post processing program *ParaView*; see Squillacote and Ahrens [124]. In agreement with Fig. 3.16 (b), the smallest corner vortices occur for $b = 20$ and increase for larger values of b. The velocity field on the slice is visualized with a rainbow color scheme and normalized by the average velocity of the downstream channel $U_2 = 14.2\,\mathrm{mm}/s$. As expected, the velocity is largest in the center of the downstream channel. For a Newtonian fluid, the ratio of the maximum to the average velocity in a downstream channel with quadratic shape is ≈ 2.0963 and this value is reached at a sufficient distance from the contraction. This distance can be estimated from the elastic stress tensor field which is shown in Fig. 3.25 in the end of this section.

Fig. 3.17 shows another interesting phenomena that is typically associated with highly elastic flows, the so called *streamline divergence*. In a Newtonian fluid, the streamlines in a contraction geometry run parallel to the channel walls and then move inwards into the downstream channel. The situation differs for high Deborah number flows as in Fig. 3.17. Here, the streamlines in the upstream channel diverge towards the channel walls shortly before they reach the corner vortex. This effect is more pronounced if larger corner vortices occur as for $b = 120$ in Fig. 3.17 (d) and Fig. 3.17 (e). We also note that the horizontal velocity component has a local minimum $u/U_2 < 0.1$ on the channel's centerline at the position with the strongest streamline divergence.

The effects of *large corner vortices* and a *streamline divergence* for high Deborah number flows are not only observed in simulations but also occur in experiments. Fig. 3.18 shows, as an illustrative example, both phenomena in an experiment by Sousa et al. [123] for a Deborah number $De = 123.59$. The flow patterns in the experiment were visualized with long time exposure streak line photography. For this purpose, a 3 mW 523 nm laser diode (green) illuminates a plane in the flow domain. Apart from a slightly lower flow rate, Fig. 3.18 resembles the simulation results in Fig. 3.17 and visualizes both flow phenomena.

In Fig. 3.19 we visualize the temporal evolution of the streamlines, again projected on a slice through the channel's center, for the 5-segment spring chain at time $t = 4.0\,s$, $t = 10.0\,s$ and $t = 68.0\,s$. In the early stage of the simulation at $t = 4.0\,s$, a corner vortex with a circular shape occurs. The $2H_1$-normalized extension of the corner vortices is 0.5. At this time, a streamline divergence is not yet visible. Then, at $t = 10\,s$ both corner vortices have doubled their extension in x-direction so that the $2H_1$-normalized length is about 1.0; cf. Fig. 3.16 (c). Furthermore, a slight deformation of the streamlines becomes visible. Finally, at $t = 68\,s$ the corner vortex structures have obtained an extremely



(a) Newtonian (steady state)

(b) $b = 20,\ t = 32.0\,s,\ N = 3$

(c) $b = 60,\ t = 52.0\,s,\ N = 3$

(d) $b = 120,\ t = 52.0\,s,\ N = 3$

(e) $b = 120,\ t = 52.0\,s,\ N = 1$

Figure 3.17.: Comparison of streamlines that are projected on a slice through the channel's center for a Newtonian and several non-Newtonian fluids.

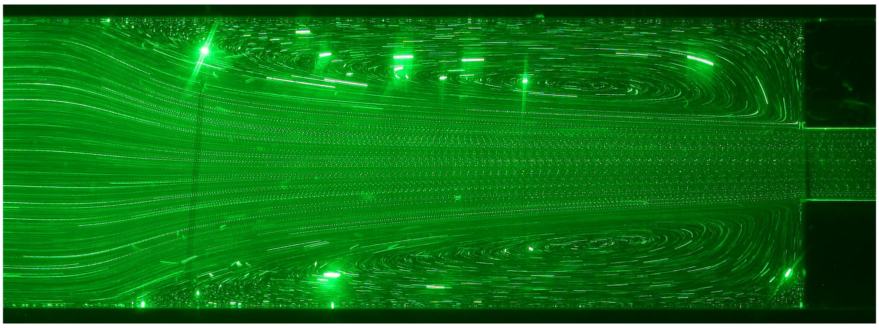
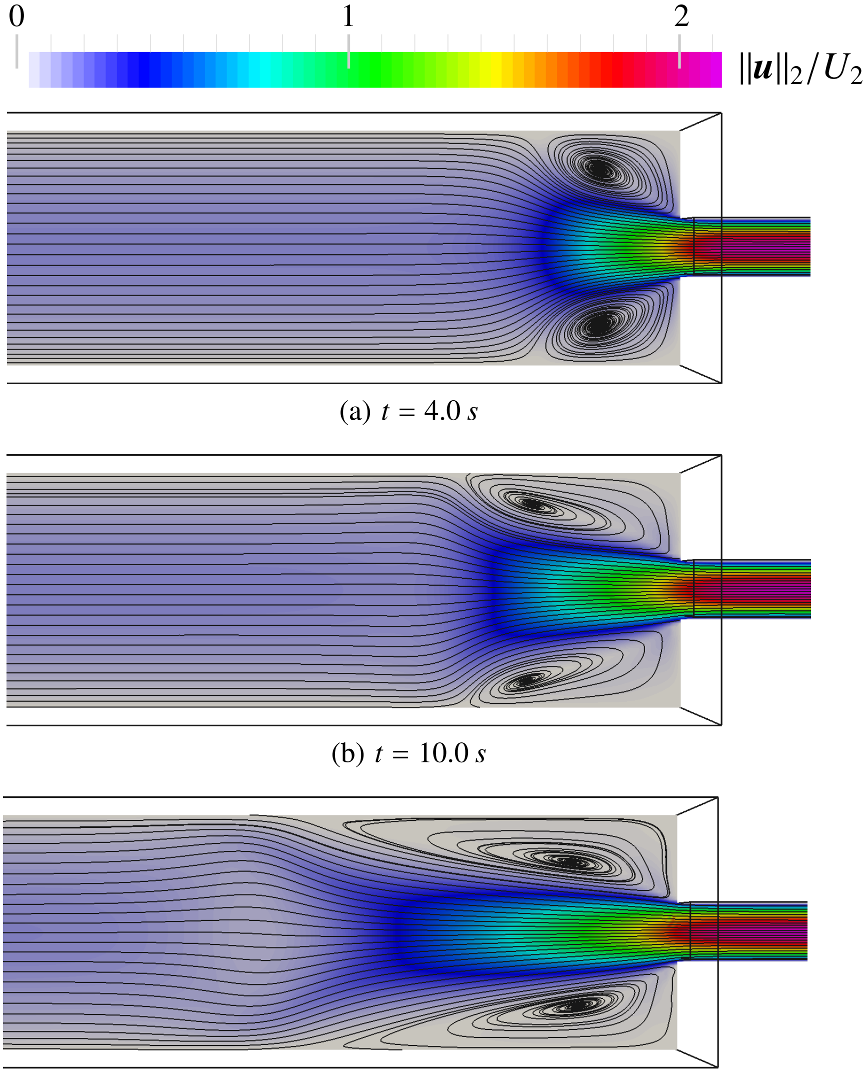


Figure 3.18.: Experimental measurement of a shear-thinning fluid in a contraction geometry with the same physical dimensions as considered in this thesis. The figure was provided by Dr. P.C. Sousa from University of Porto, Portugal.

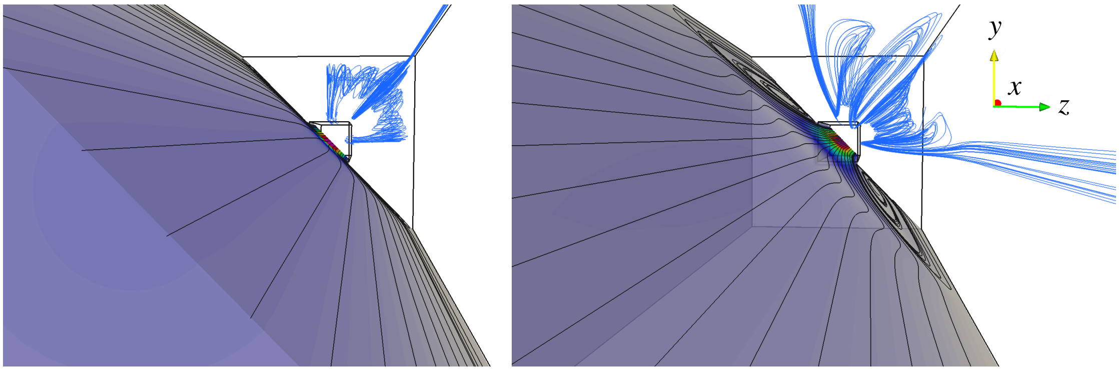


(a) $t = 4.0\,s$

(b) $t = 10.0\,s$

(c) $t = 68.0\,s$

Figure 3.19.: Temporal evolution of selected streamlines for the 5-segment spring chain with $b = 120$.



(a) De = 1.0 (PTT)

(b) De = 157 (FENE 5-segment chain)

Figure 3.20.: Three-dimensional corner vortex patterns for low Deborah number / Newtonian-like flows (left) and for highly elastic flows (right).

ellipsoidal shape and the vortex extension is ≈ 1.7 times the upstream channel's height. Moreover, a divergence of the streamline is clearly visible in the region shortly before the corner vortex starts. Again, this effect is caused by the elastic stress tensor and does not occur in the Newtonian case since we are in a laminar flow regime with Reynolds number $Re < 1$.

Up to now, we have solely focused on streamlines that run on a slice through the channel center. The results resemble the simulation outcomes from Section 3.2.1 for a planar contraction flow but, due to the higher elasticity of the fluid, larger corner vortices occur. However, in contrast to the planar case, the streamlines in the three-dimensional geometry run along a complex three-dimensional path. We illustrate this complex three-dimensional vortex structure in Fig. 3.20 for a low Deborah number flow on the left-hand side (LHS) and for a high Deborah number flow with $De = 157$ on the right-hand side (RHS). We note that the perspective in Fig. 3.20 coincides with the direction of flow, i.e. the x-direction. In Fig. 3.20 (a) a Newtonian-like flow was simulated with a macroscopic PTT model with $De = 0.5$, compare with Griebel and Rüttgers [51]. First, black solid lines show the flow field on a diagonal slice. Furthermore, blue solid lines visualize selected three-dimensional streamlines. These streamlines indicate that particles which enter the flow domain on a diagonal plane, such as $(y - H_1) + (z - H_1) = 0$, rotate by 45° towards one of the central planes $y = H_1$ or $z = H_1$, respectively. Various findings from the literature show that an inversion of this 3D vortex pattern occurs for highly elastic flows. Fig. 3.20 (b) illustrates this *flow inversion* phenomena for the simulation with $N = 5$, $De = 157$ and $b = 120$. In this case, particles enter the flow domain on the horizontal and vertical planes $y = H_1$ and $z = H_1$, respectively, and rotate in the opposite direction before they reach the contraction. Sousa et al. [123] observe that a flow inversion occurs in experiments for Deborah numbers larger than ≈ 1.5. This phenomena is reproduced in Griebel and Rüttgers [51] and is also shown in Fig. 3.20.

In Fig. 3.21 we compare the simulated steady state vortex lengths from Fig. 3.16 with the experimental measurements by Sousa et al. [123]. Again, we note that we are not aware of any other simulations of these highly elastic flows with $De = 108$ and $De = 157$. This is due to the fact that Sousa et al. [123] report stability issues for the macroscopic PTT model for Deborah numbers in the order of $O(10)$ and only multiscale approaches can be applied. First, Fig. 3.21 contains the experimental measurements by Sousa et al. [123] colored in red. These values show some variation due to measurement errors and minor oscillations in the flow. Next, Fig. 3.21 shows all findings from Griebel and Rüttgers [51] in dark

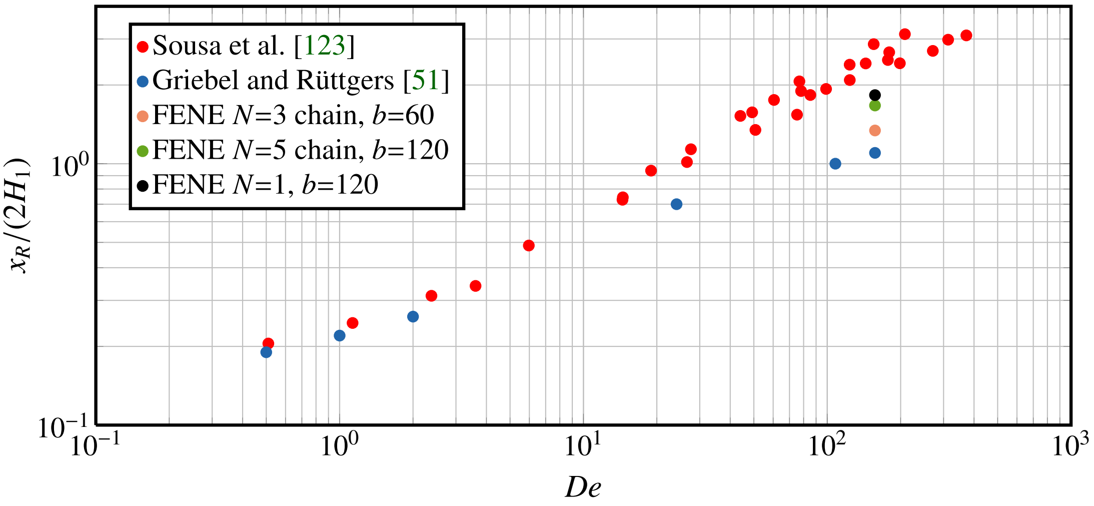


Figure 3.21.: Comparison of the $2H_1$-normalized vortex lengths on the central plane obtained in the simulations with experimental measurements.

blue. The simulation outcomes in Griebel and Rüttgers [51] for the dumbbell model with $N = 1$ are in high agreement with the experimental measurements for Deborah numbers up to $De \approx 24.1$. Then, the numerically predicted vortex sizes for $De = 108$ and $De = 157$ are by a factor of 2 smaller than the experimental measurements. According to Griebel and Rüttgers [51], two possible modifications of the simulation parameters to overcome these limitations are

- to use a FENE multi-bead-spring model instead of a FENE dumbbell model and
- to better investigate the influence of the parameter for the maximum spring extensibility b.

As already shown in Fig. 3.16, we conduct both possible extensions in this thesis. Therefore, Fig. 3.21 also contains numerically predicted vortex sizes for the simulations with a 3-segment chain with $b = 60$, a 5-segment chain with $b = 120$ and for a dumbbell model with $b = 120$. All these simulations employ a flow rate such that the corresponding Deborah number is $De = 157$. In agreement with Fig. 3.16, the steady state vortex sizes strongly increase with the parameter b. On the downside, the 3-segment and 5-segment chains predict slightly smaller steady state vortex sizes than the dumbbell model. As a preliminary result, the dumbbell model with $N = 1$ and $b = 120$ seems to better match the experiments than the more complex multi-segment chains with respect to the experimental vortex sizes. But, it has to be taken into account that $b = 120$ is already larger than the sensible parameter range $b \in [20, 100]$ that is suggested by Herrchen and Öttinger [59]. Therefore, the quality of the different spring-chain models cannot be estimated by the prediction of the steady state vortex sizes alone so that we compare further flow features in the following.

As mentioned before, Fig. 3.21 shows a high agreement between the multiscale polymer simulations and the experimental measurements by Sousa et al. [123] for low and medium Deborah numbers up to about 24.1. To better illustrate this, Fig. 3.22 compares the velocity component u in flow direction on the channel's centerline, i.e. along the line with $y = H_1$ and $z = H_1$. The simulation uses a FENE dumbbell model and is described in detail in Griebel and Rüttgers [51]. As expected, the flow velocity in the upstream channel is comparatively small in the region far away from the contraction. Then, at a $2H_1$-normalized distance of about $x = -0.8$ with respect to the contraction, the fluid's velocity increases.

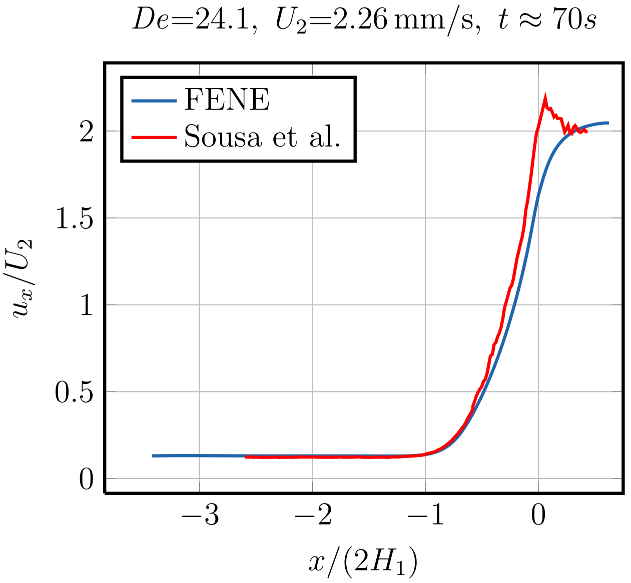


Figure 3.22.: Velocity profiles on the channel's centerline with a dumbbell model and $De = 24.1$.

We note that the position at which the fluid starts to accelerate roughly corresponds with the maximum extension of the corner vortices. Furthermore, we observe an excellent agreement of this position between experiment and simulation in Fig. 3.22. The simulation then predicts a maximum U_2-normalized velocity of about 2.1 which closely matches the analytical value of ≈ 2.0963 for a quadratic channel. In this case, there are some minor differences with the experimental measurement. Unfortunately, the experimental measurement ends shortly after the contraction. Therefore, it is not clear when the experimental fluid has relaxed and when its velocity profile adapts the analytic value. Furthermore, we observe some minor oscillations in the experimental measurements from Sousa et al. [123]. Since the authors do not give an error analysis for their measured velocity profile, it is difficult to distinguish measurement errors and elastic fluid behavior. Nevertheless, the general accordance with the measurements is high.

Fig. 3.23 shows that the difference and uncertainty between the measurement and the simulation is larger for $De = 157$. First, the experimental measurements by Sousa et al. [123] strongly oscillate for $De = 157$ (black solid line) and also, for comparison, for an intermediate Deborah number $De = 108$ (light blue line). Since the predicted vortex sizes for these Deborah numbers can differ by more than 50 % for repeated measurements, see Fig. 3.21, this uncertainty should be taken into account when the quality of the simulation outcome is evaluated. Nevertheless, all four simulations in Fig. 3.23 underestimate the experimental vortex size and therefore predict a later fluid acceleration on the centerline. As mentioned before, the simulation results become better with larger extensibility values b. Interestingly, the velocity profile does not change significantly between the 3-segment and the 5-segment chain. Furthermore, the simulation result with $N = 1$ and $b = 120$ (green line) has the highest accordance with the experimental measurement.

Next, all simulations in Fig. 3.23 reproduce a phenomena that is known as *velocity undershoot*, a local minimum of the velocity profile. More precisely, this term denotes the effect that a minimum of the horizontal velocity profile occurs shortly before the fluid starts to accelerate. This minimum does not

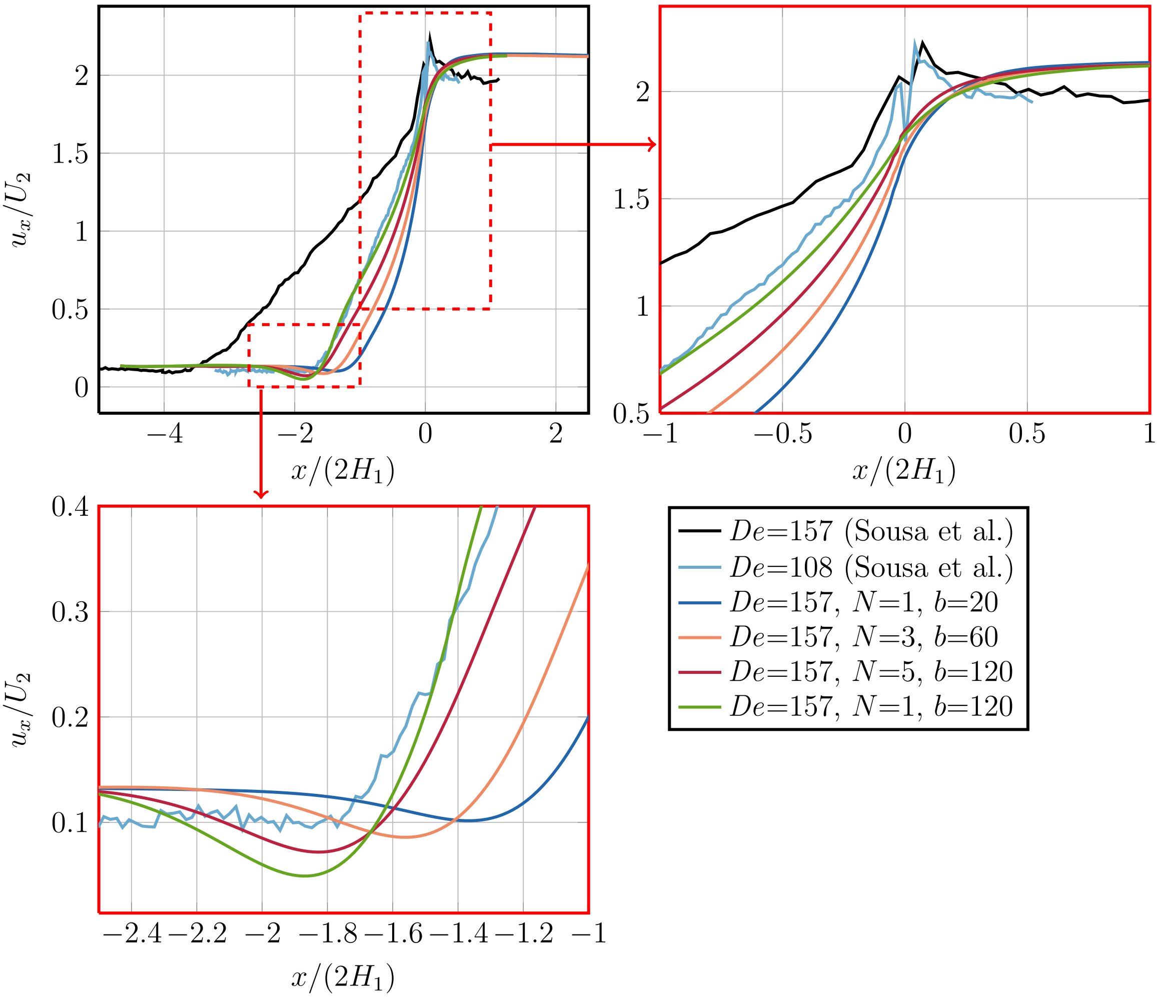


Figure 3.23.: Spring-chain model comparison of the central velocity profiles for a high Deborah number flow with $De = 157$.

occur for Newtonian and for low Deborah number flows. The effect is connected with the streamline divergence shown, for instance, in Fig. 3.17 and Fig. 3.19. The velocity undershoot is shown in a zoomed extract in Fig. 3.23. The undershoot is also observed by Sousa et al. [123] but more difficult to detect in Fig. 3.23 since the lower fluid measurements in the upstream channel are prone to oscillations.

In Fig. 3.24, the dimensionless pressure gradient $(\nabla_{x/H_2} p)/((\eta_s + \eta_p)U_2/H_2)$ is plotted on the channel's centerline. Similarly to Fig. 3.23, the origin of the horizontal axis has been shifted such that it coincides with the beginning of the contraction. Normally, the pressure gradient is negative for a fluid flow in positive coordinate direction. Far away from the contraction, the dimensionless pressure gradient is ≈ -0.023 in the upstream channel and ≈ -3.85 in the downstream channel. But, in the interval $[-H_2, H_2]$ the pressure gradient varies strongly in all four multiscale flow simulations. First, we observe a *pressure recovery* such that $(\nabla_{x/H_2} p)/((\eta_s + \eta_p)U_2/H_2) > 0$ in all models. This pressure recovery correlates with the velocity undershoot phenomena in Fig. 3.23. According to Fig. 3.24, the pressure recovery for the model parameters $(N = 3,\ b = 20)$ and $(N = 3,\ b = 60)$ is much lower than for

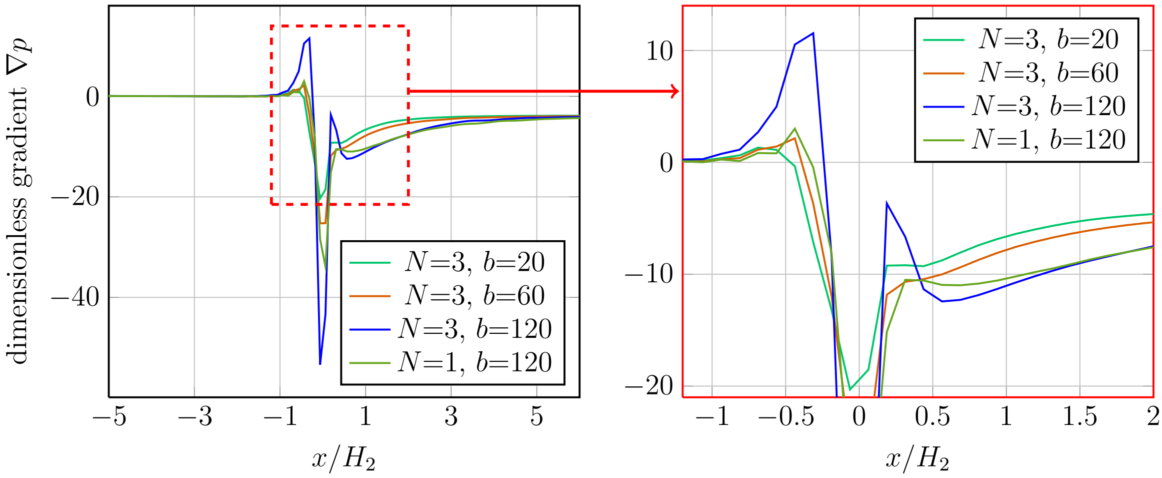


Figure 3.24.: Pressure gradient over the centerline for different spring chain models.

(N = 3, b = 120). Furthermore, this is not only affected by the spring extensibility b but also influenced by the number of spring segments. More precisely, for (N = 1, b = 120) the pressure recovery is much lower than for the 3-segment chain. Moreover, we note that the pressure profile for the 5-segment chain with b = 120 closely resembles the result of the 3-segment chain. Shortly after the pressure recovery, all models predict a large pressure drop with a global minimum at $x/H_2 \approx 0$. Then, the pressure gradient has a further local maximum at $x/H_2 \approx 0.5$ for both models with $b = 120$. Finally, the pressure gradient asymptotically reaches the Newtonian value in the downstream channel. In contrast to this, a Newtonian pressure profile shows only one large pressure drop exactly at the contraction.

With respect to the pressure prediction, there seems to be a significant difference between a dumbbell model on the one hand and multi-segment chains on the other hand. Without an experimental comparison it cannot be decided which model achieves better results. There are, however, pressure comparisons by Koppol et al. [75] for a two-dimensional 4:1:4 contraction-expansion flow. In this publication, the predictions of the pressure drop for a macroscopic FENE-P closure model, a multiscale FENE dumbbell and a multiscale FENE three-bead-spring model are compared with experimental results. The authors conclude that the results of the 3-segment model had the best accordance with the measurement. Therefore, the results by Koppol et al. [75] indicate that the pressure profile of the 3-segment chain in Fig. 3.24 better matches experimental data. Since large pressure drops are also expected for this type of flow, it is probable that our multi-spring chain results also better coincide with the behavior in nature.

Finally, we focus on the stress tensor and on the stochastic polymer configurations. For this purpose, Fig. 3.25 shows the τ_{xx} stress component on a central plane for the 5-segment spring chain model with De = 157 close to the steady state at t = 68 s. In agreement with the findings by Keunings [70] for the Brownian configuration field (BCF) method, the stress field is smooth in space at a fixed point in time. At a first glance, this result is unexpected for a stochastic approach. But the smoothness in space is caused by a locally correlated Brownian force as described in Chapter 2. On the downside, we note that stochastic noise still occurs over time; see e.g. Fig. 38 in Griebel and Rüttgers [51]. Fig. 3.25 shows that τ_{xx} has two stress peaks close to the corner singularity. High normal stress values are also observed along the streamlines that run close to the corner singularity. This observation illustrates that streamline

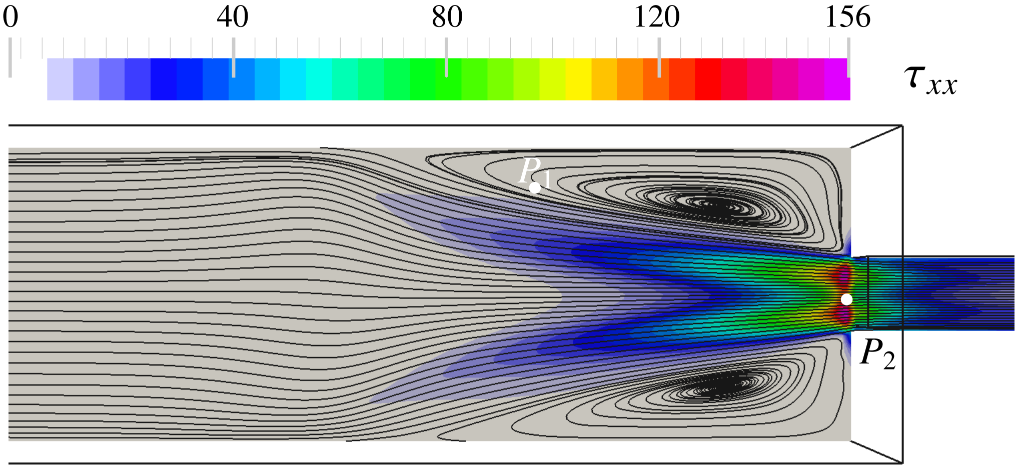


Figure 3.25.: Simulation of the τ_{xx} stress component for the 5-segment spring chain model with $De = 157$ at steady state. The points P_1 and P_2 illustrate the position of the configurations fields that are analyzed in Fig. 3.26 and Fig. 3.27.

divergence as shown in Fig. 3.19 is an elastic effect that is caused by the stress tensor.

Fig. 3.25 contains two additional points P_1 and P_2 at which we visualize the polymer chain orientation. A similar analysis can be found in Prieto et al. [108] for a 2D planar contraction flow. This analysis allows to better characterize the fluid at this position. The exact positions of P_1 and P_2 are

$$P_1 = (86.0625\,\text{mm}, 20.4375\,\text{mm}, 11.8125\,\text{mm})$$
$$P_2 = (112.3125\,\text{mm}, 11.8125\,\text{mm}, 11.8125\,\text{mm}).$$

These positions coincide with the cell centers of the finite difference cells $(i, j, k) = (230, 55, 32)$ and $(i, j, k) = (300, 32, 32)$. The distance of both points to the central plane $z = H_1$ is half the size of the equidistant mesh width.

In Fig. 3.26 the Brownian configuration fields are visualized at position P_1 close to the upper corner vortex and at position P_2 close to the centerline. As listed in Table 3.6, every grid cell contains $M = 1200$ samples of dimensionality $3N = 15$. The first column of Fig. 3.26 plots the first three dimensions of the samples in blue and, analogously, the second column shows the dimensions that represent the third spring segment in red, i.e. plots the dimensions 7, 8 and 9. Furthermore, the last column superimposes the sample dimensions $(1, 2)$ and $(7, 8)$ in one figure which corresponds to the x/y-component of the first and the third spring segment. Obviously, the distribution of the samples differs between P_1 and P_2. At position P_1, the samples are rotated against the horizontal axis and are moderately extended. Furthermore, the third segment is more extended, in average, than the first segment. The distribution of the samples is typical for a shear flow that is expected at this flow position. Next, the second row of Fig. 3.26 analyzes the configuration fields at position P_2. In this case, the configuration fields are aligned with the flow direction. Furthermore, most samples have an extension close to the maximum value of $b(5)^{1/2} = (120/5)^{1/2} \approx 4.9$ and the samples are more localized. This corresponds with results from the literature which state a dominant extensional flow on the centerline.

Analogously to Fig 3.7 in Section 3.1.3, Fig. 3.27 plots the density function of the squared configuration vector for the samples at position P_1 and P_2. This allows us to better characterize the strength of the flow but does not deliver information on the flow orientation any more. We further note that

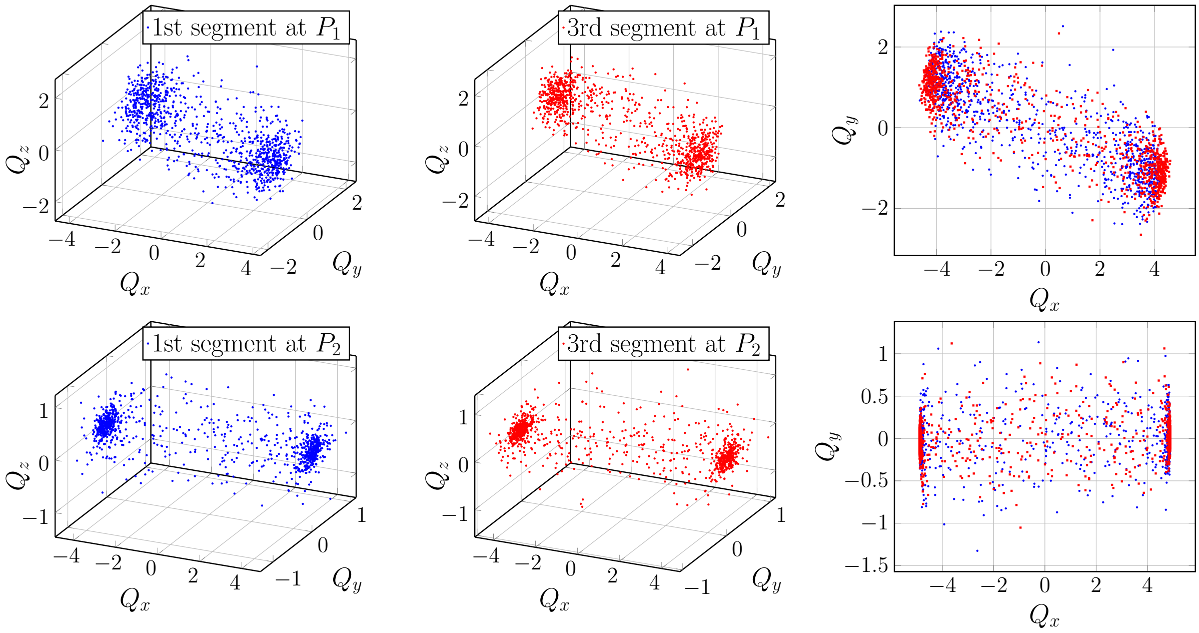


Figure 3.26.: Visualization of the first and the third component of the 15-dimensional Brownian configuration fields at two positions in flow space.

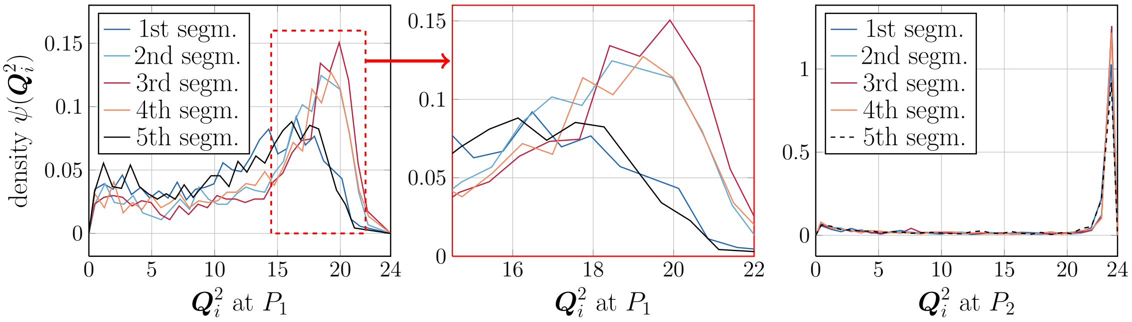


Figure 3.27.: Comparison of the density function for the squared configuration vector of a 5-segment chain at two positions P_1 (left, center) and P_2 (right).

our temporal scheme, which is implicit in the spring force term, ensures a valid maximum extension $\leq b(5)^{1/2}$ for each sample in configuration space. Consequently, the probability density function is zero for squared extensions equal to or larger than $b(5) = 24$. At position P_1, the third spring segment in the middle of the polymer chain is extended most. On the other hand, the end of the polymer chain which corresponds to the first and fifth segment is slightly less extended. The second and fourth spring segment have an extension between these two regimes. We therefore conclude that the shear flow at P_1 leads to a molecular extension that is strongest at the polymer chain's center. The situation differs at position P_2 on the right-hand side of Fig. 3.27. Here, we observe a much stronger polymer extension since nearly all samples are close to their maximum extension. Furthermore, there is no difference between the five spring segments. This is an effect of the strong extensional flow on the centerline. The strong fluid field reduces the degrees of freedom for the polymeric molecules.

As a conclusion, our simulations were able to reproduce all flow phenomena that are reported in the literature for highly elastic fluid flows in contraction geometries. These phenomena are

- streamline divergence, cf. Fig. 3.19,
- an inversion of the flow field, cf. Fig. 3.20,
- an undershoot of the velocity component on the centerline, cf. Fig. 3.23,
- a pressure recovery shortly before the contraction, cf. Fig. 3.24, and
- an enormous corner vortex increase compared to the Newtonian case, cf. Fig. 3.21.

Furthermore, our simulations reveal the differences between dumbbell models on the one hand and spring-chain models on the other hand. According to Koppol et al. [75], the pressure drop predicted by spring-chain models better corresponds with experimental measurements than the prediction by a simple dumbbell model. This justifies the increase of computational complexity that occurs for more complex multi-spring models.

As reported in Section 2.3, the multiscale simulations in this chapter required weeks of computing time on massively parallel clusters such as *Atacama*. It is therefore essential to reduce the computational complexity of these simulations if even more complex problems shall be considered. For this reason, the second part of this thesis will concentrate on the reduction of the computational complexity. In the following, we will show that sparse grids can be ideally used to weaken or even break the curse of dimensionality for our applications in polymer physics.

Part II.

Sparse grids for polymeric fluids

4. Sparse grids

In Chapter 2 we have presented an algorithm to allow for multiscale simulations of viscoelastic fluids. Theoretically, all dilute polymeric flow problems of practical interest can be solved by using this approach. In practice, however, our simulations with the flow solver *NaSt3DGPF* suffer from the so called *curse of dimensionality*; cf. Section 2.3. Therefore, concepts are required that dramatically reduce the complexity of the physical problem.

In this chapter, we discuss the *combination technique* as one viable method to cope with multiscale viscoelastic flow problems. The combination technique is a variant of a *sparse grid* discretization. We first discuss the general concepts of sparse grids in Section 4.1. Then, we focus on the sparse grid combination technique in Section 4.2. Furthermore, in Section 4.3 the relation between dimension-adaptive sparse grids on the one hand and the Multilevel Monte Carlo method on the other hand is discussed. Using these concepts in Chapter 5, we derive a dimension-adaptive algorithm that applies the combination technique to the multiscale fluid equations. This adaptive algorithm is used in Chapter 6 to decrease the computing time of multiscale polymeric fluid simulations enormously.

4.1. General concepts

Let d denote the dimensionality of a PDE problem and N the number of grid points in each dimension of a tensor-product grid-based discretization scheme, then the complexity with respect to the number of grid points is of order $O(N^d)$. The term *curse of dimensionality* is usually applied to describe this exponential growth of complexity. This growth underlines the importance of more elaborated approaches to cope with high-dimensional problems. One possible approach to reduce the complexity is given by the sparse grid method that was introduced to PDE problems by Zenger [141]. Discretizations on a sparse grid have a significantly reduced complexity of order $O(N \cdot (\log N)^{d-1})$. Furthermore, they preserve nearly the same accuracy as a tensor product grid approach provided that the solution is sufficiently smooth. The basic principle of sparse grids, a high-dimensional multiscale basis, was first used by Smolyak [121] for integration problems. We here describe the basic concepts and refer to Bungartz and Griebel [18] and the references given therein for a more detailed description. Moreover, our summary of the general concepts is based on Pflüger [103] and on Garcke [40].

Let $u : \boldsymbol{x} \in \bar{\Omega} = [0,1]^d_{i=1} \subset \mathbb{R}^d \mapsto u(\boldsymbol{x}) \in \mathbb{R}$ denote a real-valued function defined on a d-dimensional hypercube. This function u is the solution of a PDE problem of the form

$$\begin{aligned} \mathcal{L}(u) &= f && \text{on } \Omega \\ u &= g && \text{on } \partial\Omega \end{aligned} \tag{4.1}$$

on a domain Ω with a general second-order differential operator $\mathcal{L}(\cdot)$ and a right-hand side f. In our application, the coupled multiscale system in (1.59)–(1.63), the unknown u might represent the pressure field, a velocity field component or a component of the stress tensor. Furthermore, this interpretation requires a suitable smooth transformation from the high-dimensional flow and polymer configuration space on which u is defined to the d-dimensional unit cube on which sparse grids are defined.

After discretization the problem is given on a discrete mesh $\Omega_{\boldsymbol{l}}$ with level $\boldsymbol{l} \in \mathbb{N}^d$ and a corresponding mesh size $h_{\boldsymbol{l}} = (h_1, \ldots, h_d) = (2^{-l_1}, \ldots, 2^{-l_d})$. In general, the mesh widths differ in each coordinate direction. In a single direction it is, however, assumed to be equidistant. Consequently, the approximation space is defined on discrete points

$$\boldsymbol{x}_{\boldsymbol{l},\boldsymbol{j}} = (x_{l_1,j_1}, \ldots, x_{l_d,j_d}) \tag{4.2}$$

with $j_i = 0, \ldots, 2^{l_i}$ in each dimension $i = 1, \ldots, d$.

Depending on the discretization scheme applied to the PDE problem (4.1) the discrete approximation spaces $V_{\boldsymbol{l}}$ associated with the grid $\Omega_{\boldsymbol{l}}$ differ. In the Finite Element method (FEM) with homogeneous Dirichlet boundary conditions the approximation space associated with the interior grid consists of polynomial basis functions of degree p on a compact support. The most simple choice is given by one-dimensional hat functions $\phi_{l,j}$ for $p = 1$ which are defined for $x \in [0, 1] \subset \mathbb{R}$ as

$$\phi_{l_i,j_i}(x) = \begin{cases} 1 - |x/h_{l_i} - j_i| & \text{for } x \in [(j_i - 1)h_{l_i}, (j_i + 1)h_{l_i})] \cap [0, 1] \\ 0 & \text{else} \end{cases} \tag{4.3}$$

for $i = 1, \ldots, d$. The tensor product of the one-dimensional functions

$$\phi_{\boldsymbol{l},\boldsymbol{j}}(\boldsymbol{x}) = \prod_{i=1}^{d} \phi_{l_i,j_i}(x_i), \quad \boldsymbol{x} \in \Omega, \tag{4.4}$$

then defines a basis. The approximation space $V_{\boldsymbol{l}}$ as linear combination of the basis

$$V_{\boldsymbol{l}} = \operatorname{span}\{\phi_{\boldsymbol{l},\boldsymbol{j}} | j_i = 0, \ldots, 2^{l_i},\ i = 1, \ldots, d\} \tag{4.5}$$

is the standard *nodal point basis*.

4.1.1. Hierarchical increment space

Using the approximation space $V_{\boldsymbol{l}}$ we define the *hierarchical increment space* or *hierarchical difference space* $W_{\boldsymbol{l}}$ as

$$W_{\boldsymbol{l}} = \operatorname{span}\{\phi_{\boldsymbol{l},\boldsymbol{j}} | j_i = 1, \ldots, 2^{l_i} - 1,\ j_i \text{ odd for all } i = 1, \ldots, d\}. \tag{4.6}$$

With hierarchical increment spaces it is possible to rewrite the full grid space $V_{\boldsymbol{l}}$ as a direct sum of subspaces in the form

$$V_{\boldsymbol{l}} = \bigoplus_{k_1=1}^{l_1} \cdots \bigoplus_{k_d=1}^{l_d} W_{k_1}^{(1)} \otimes \ldots \otimes W_{k_d}^{(d)} = \bigoplus_{k_1=1}^{l_1} \cdots \bigoplus_{k_d=1}^{l_d} W_{k_1,\ldots,k_d} = \bigoplus_{\boldsymbol{k} \le \boldsymbol{l}} W_{\boldsymbol{k}} \tag{4.7}$$

where the relation $\le$ on the right-hand side of (4.7) is a component-wise operation. In the special case $l_1 = \cdots = l_d = n$ we rewrite (4.7) in a simplified form as

$$V_n^{(\infty)} = \bigoplus_{|\boldsymbol{k}|_\infty \le n} W_{\boldsymbol{k}}. \tag{4.8}$$

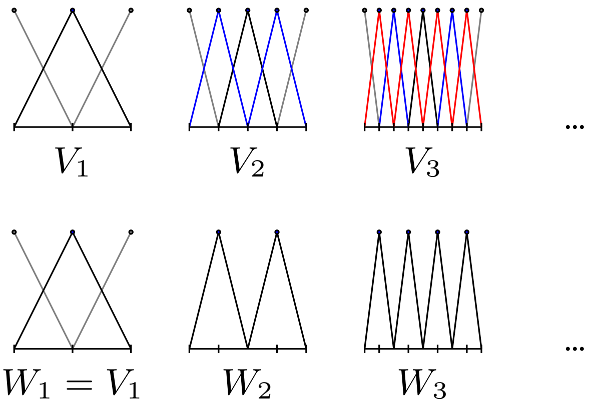


Figure 4.1.: Nodal and hierarchical basis functions for the levels $l = 1, 2, 3$ and $d = 1$.

As a result, the increment space lead to a multilevel subspace decomposition. The elements of $W_{\boldsymbol{l}}$ in (4.6) form a basis which is called a *hierarchical basis*. In Fig. 4.1 we compare the basis functions of V_l and W_l for the levels $l = 1, 2, 3$ and $d = 1$. When the hierarchical increment space is used as ansatz space in a FEM discretization, the discrete solution $\tilde{u}$ on $V_{\boldsymbol{l}}$ is approximated as

$$\tilde{u}(\boldsymbol{x}) = \sum_{\boldsymbol{k} \le \boldsymbol{l}} u_{\boldsymbol{k}}(\boldsymbol{x}) = \sum_{\boldsymbol{k} \le \boldsymbol{l}} \sum_{\substack{1 \le \boldsymbol{j} \le 2^{\boldsymbol{l}}-1 \\ \boldsymbol{j} \text{ is odd}}} c_{\boldsymbol{l},\boldsymbol{j}} \cdot \phi_{\boldsymbol{l},\boldsymbol{j}}(\boldsymbol{x}) \tag{4.9}$$

with coefficients $c_{\boldsymbol{l},\boldsymbol{j}} \in \mathbb{R}$ and partial solutions $u_{\boldsymbol{k}}(\boldsymbol{x})$ in $W_{\boldsymbol{k}}$. The coefficients $c_{\boldsymbol{l},\boldsymbol{j}}$ in a hierarchical representation are called *hierarchical surplus*. Both basis representations, the nodal and the hierarchical basis, require $\prod_{i=1}^{d}(2^{l_i} - 1)$ inner basis functions when the boundary conditions are not taken into account. For $l_1 = \cdots = l_d = n$ the full grid $V_n^{(\infty)}$ has a complexity

$$|V_n^{(\infty)}| = (2^n - 1)^d = \sum_{i=0}^{n} \binom{n}{i} 2^{nd-i} \cdot (-1)^i \approx \mathcal{O}(N^d) \tag{4.10}$$

with $N = 2^n$. The major difference of the hierarchical representation is the decay of the hierarchical surplus $c_{\boldsymbol{l},\boldsymbol{j}}$ on higher levels $\boldsymbol{l}$; cf. Bungartz and Griebel [18].

In a finite difference (FD) discretization we employ the same discrete grid points $\boldsymbol{x}_{\boldsymbol{l},\boldsymbol{j}}$ as in (4.2). In contrast to a FEM discretization, the solution is only described on these grid points. Consequently, the approximation spaces $V_{\boldsymbol{l}}$ and $W_{\boldsymbol{l}}$ are the discrete point spaces defined on these grid points. A drawback in comparison to a FEM discretization is the stronger regularity assumption on the solution. Typically, the solution u is an element of a Hölder space instead of a Sobolev space. Next, we employ the hierarchical increment spaces $W_{\boldsymbol{l}}$ to construct an approximation space of level l with a reduced complexity compared with the full grid spaces in (4.7) and (4.8).

4.1.2. Sparse grids

The basic idea of sparse grids is to construct an approximation space that consists of a combination of increment spaces from a cost-benefit optimized index set $\mathcal{I} \subset \mathbb{N}^d$, i.e. $V^{\text{approx}} = \bigoplus_{\boldsymbol{k} \in \mathcal{I}} W_{\boldsymbol{k}}$. For each

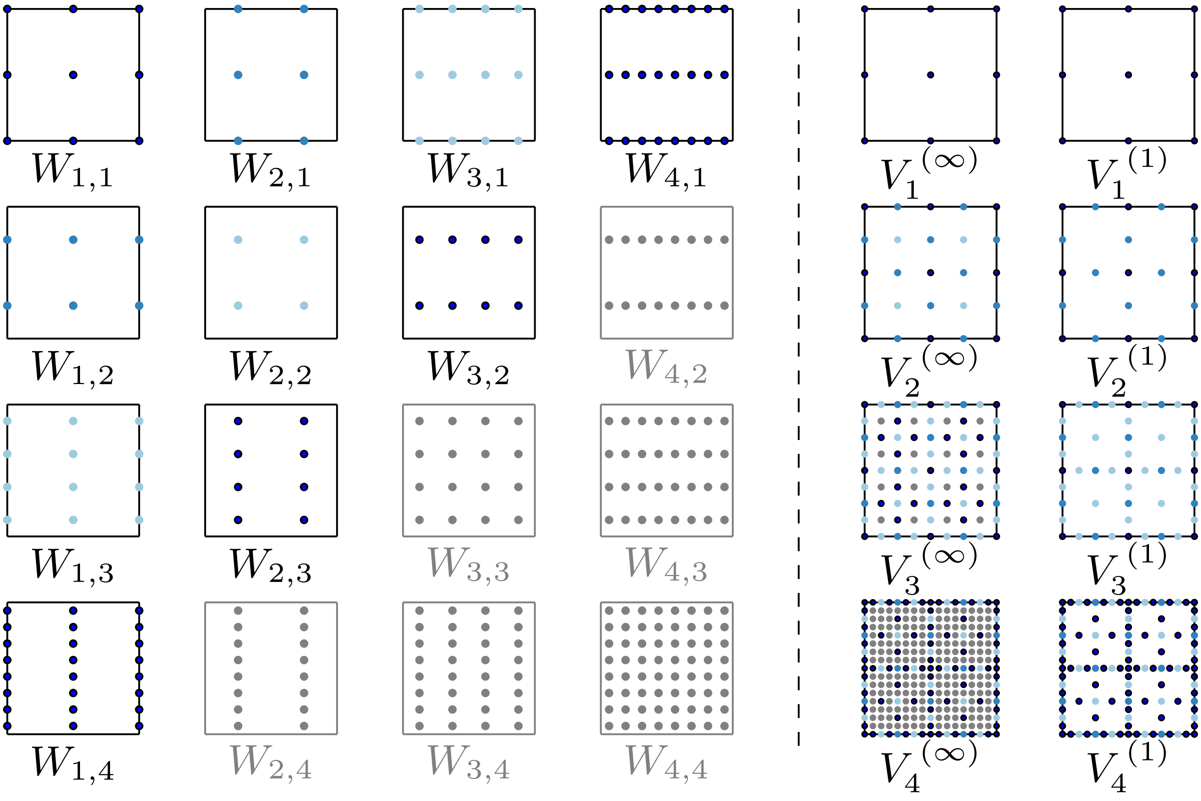


Figure 4.2.: Illustration of the two-dimensional hierarchical increment spaces $W_{\boldsymbol{k}}$ up to level $|\boldsymbol{k}|_\infty = 4$ and the corresponding full grid and L^2 sparse grid spaces $V_n^{(\infty)}$ and $V_n^{(1)}$ for $n = 1, 2, 3, 4$.

increment space $W_{\boldsymbol{k}}$, which is uniquely characterized by its index $\boldsymbol{k} \in \mathbb{N}^d$, a local cost function $c(\boldsymbol{k})$ is defined by

$$c(\boldsymbol{k}) = 2^{|\boldsymbol{k}|_1 - 1} \tag{4.11}$$

with $|\boldsymbol{k}|_1 = \sum_{i=1}^d k_i$. Moreover, a local benefit function $b_*(\boldsymbol{k})$ is given by upper bounds for $\|u_{\boldsymbol{k}}\|_*^2$ with $u_{\boldsymbol{k}} \in W_{\boldsymbol{k}}$. These upper bounds are stated in Lemma 3.4 of Bungartz and Griebel [18] for the discrete L^2-norm, L^∞-norm and energy norm. The local cost-benefit ratio $\mathrm{cbr}_*(\boldsymbol{k})$ for a general norm $\|\cdot\|_*$ is defined as

$$\mathrm{cbr}_*(\boldsymbol{k}) = \frac{b_*(\boldsymbol{k})}{c(\boldsymbol{k})}. \tag{4.12}$$

The generalized index set $\mathcal{I}$ then consists of all indices $\boldsymbol{k}$ with $\mathrm{cbr}_*(\boldsymbol{k}) \geq \sigma_*(n)$ for a given threshold $\sigma_*(n)$. The threshold $\sigma_*(n)$ of level n is in the order of the cost-benefit ratio $\mathrm{cbr}_*((n, 1, \dots, 1))$ if the cost is equally distributed in all dimensions. This means that we fix the finest increment space that is uniquely refined in the first dimension and add other indices that can also be refined in other dimension with an equal or better cost-benefit ratio. The resulting index set is

$$\mathcal{I} = \begin{cases} \{\boldsymbol{k} \in \mathbb{N}^d \mid |\boldsymbol{k}|_1 \leq n + d - 1\} & \text{for } L^2\text{- and } L^\infty\text{-norm,} \\ \{\boldsymbol{k} \in \mathbb{N}^d \mid |\boldsymbol{k}|_1 - \frac{1}{5}\log_2(\sum_{i=1}^d 4^{k_j}) \leq n + d - 1 - \frac{1}{5}\log_2(4^n + 4d - 4)\} & \text{for energy norm.} \end{cases} \tag{4.13}$$

The L^2-based sparse grid space of level n that we will use in the following is therefore given by

$$V_n^{(1)} = \bigoplus_{k_1=1}^{n+d-1} \cdots \bigoplus_{k_d=1}^{n+d-1-\sum_{i<d} k_i} W_{k_1}^{(1)} \otimes \ldots \otimes W_{k_d}^{(d)} = \bigoplus_{|\boldsymbol{k}|_1 \leq n+d-1} W_{\boldsymbol{k}}. \tag{4.14}$$

In Fig. 4.2 we illustrate the grid points which build the hierarchical increment spaces $W_{\boldsymbol{k}}$ for $d = 2$. The second to last column shows the grid points of the full grid spaces $V_n^{(\infty)}$ for $n = 1, 2, 3$ and 4. Analogously, the last column visualizes the corresponding L^2-based sparse grid spaces $V_n^{(1)}$.

Compared to the full grid space $V_n^{(\infty)}$, the number of degrees of freedom reduces from $O(N^d)$ in (4.10) to

$$|V_n^{(1)}| = \sum_{i=0}^{n-1} 2^i \cdot \binom{d-1+i}{d-1} \approx 2^n \cdot \left(\frac{n^{d-1}}{(d-1)!} + O(n^{d-2}) \right) \approx O(N \cdot \log_2(N)^{d-1}) \tag{4.15}$$

in which $N = 2^n$ and $|W_{\boldsymbol{k}}| = 2^{|\boldsymbol{k}-\boldsymbol{1}|_1}$ have been used. The basic question is if the reduced complexity of the sparse grid space in (4.15) compared to the full grid space in (4.10) also deteriorates the solution's accuracy. For this purpose, the regularity of the solution u has to be considered.

4.1.3. Assumptions on the regularity of the solution

Classically, we assume that the solution $u : \boldsymbol{x} \in \bar{\Omega} = [0,1]_{i=1}^d \subset \mathbb{R}^d \mapsto u(\boldsymbol{x}) \in \mathbb{R}$ is contained in the Hölder space $C^m(\bar{\Omega}, \mathbb{R})$. This space contains all functions $u : \Omega \to \mathbb{R}$ that are m-times continuously differentiable and for which the partial derivatives $\partial^s u$ of order $|s|_1 \leq m$ are continuously extendible on $\bar{\Omega}$. The space $C^m(\bar{\Omega}, \mathbb{R})$ is a Banach space with norm

$$||u||_{C^m} = \sum_{|s|_1 \leq m} ||\partial^s u||_{C^0}$$

in which $||u||_{C^0} = \sup_{\boldsymbol{x} \in \bar{\Omega}} |u(\boldsymbol{x})|$. Additionally, $C_0^m(\Omega, \mathbb{R})$ denotes the subspace with functions $f \in C^m$ that have a zero boundary condition on $\partial\Omega$. In a FD discretization of (4.1) Hölder regularity is required.

The regularity assumptions are weakened for the PDE problem (4.1) in its weak formulation. For this purpose, let $L^p(\bar{\Omega}, \mathbb{R})$ be the class of p-integrable functions on the measurable space $\bar{\Omega}$. Since $\bar{\Omega}$ is a subset of $\mathbb{R}^n$, the measurable space is defined using the underlying Borel σ-algebra $\mathcal{B}$ and the Lebesgue measure μ. For $1 \leq p < \infty$ the Banach space $L^p(\bar{\Omega}, \mathbb{R})$ is defined as

$$L^p(\bar{\Omega}, \mathbb{R}) = \{ f : \bar{\Omega} \to \mathbb{R} \,|\, f \text{ is } \mu\text{-measurable and } ||f||_{L^p} < \infty \} \tag{4.16}$$

with norm

$$||f||_{L^p} = \left(\int_{\bar{\Omega}} |f|^p \, d\mu \right)^{\frac{1}{p}}. \tag{4.17}$$

The space $L^p(\bar{\Omega}, \mathbb{R})$ consists of equivalence classes of measurable functions and two elements are identical if they are equal almost everywhere with respect to μ. The definition can be extended to the case $p = \infty$ if $f : \bar{\Omega} \to \mathbb{R}$ is essentially bounded with respect to μ, i.e.

$$\sup_{\boldsymbol{x} \in \bar{\Omega} \setminus N} |f(\boldsymbol{x})| < \infty$$

for a μ-measure zero set $N \in \mathcal{B}$. The corresponding norm for $L^\infty(\bar{\Omega}, \mathbb{R})$ is

$$\|f\|_{L^\infty} \;=\; \inf_{\mu\text{-zero sets } N} \left(\sup_{\boldsymbol{x}\in\bar{\Omega}\setminus N} |f(\boldsymbol{x})|\right). \tag{4.18}$$

The elements in $L^p(\bar{\Omega}, \mathbb{R})$ are unique up to a μ-zero set, i.e.

$$f = g \text{ in } L^p(\bar{\Omega}, \mathbb{R}) \;\Leftrightarrow\; f = g \text{ up to a } \mu\text{-zero set.}$$

We further note that $L^2(\bar{\Omega}, \mathbb{R})$ is also a Hilbert space with inner product

$$(f, g)_{L^2} \;=\; \int_{\bar{\Omega}} f(\boldsymbol{x}) g(\boldsymbol{x}) \, d\mu(\boldsymbol{x})$$

for any two elements $f, g \in L^2(\bar{\Omega}, \mathbb{R})$.

Next, the weak derivative $D^{(\boldsymbol{s})} f \in L^p(\bar{\Omega}, \mathbb{R})$ of order $|\boldsymbol{s}|_1 \leq m$ is defined as the solution of the integral equation

$$\int_{\bar{\Omega}} \partial^{\boldsymbol{s}} \varphi \cdot f \, d\mu \;=\; (-1)^{|\boldsymbol{s}|_1} \int_{\bar{\Omega}} \varphi \cdot D^{(\boldsymbol{s})} f \, d\mu \tag{4.19}$$

for all $\varphi \in C_0^m(\Omega, \mathbb{R})$. We then define the Sobolev space of order $m \in \mathbb{N}$ and exponent p, $1 \leq p \leq \infty$, as

$$W^{m,p}(\bar{\Omega}, \mathbb{R}) \;=\; \{f \in L^p(\bar{\Omega}, \mathbb{R}) \,|\, D^{(\boldsymbol{s})} f \in L^p(\bar{\Omega}, \mathbb{R}) \text{ for all } |\boldsymbol{s}|_1 \leq m\}. \tag{4.20}$$

This space is a Banach space with norm

$$\|f\|_{W^{m,p}} \;=\; \left(\sum_{|\boldsymbol{s}|_1 \leq m} \|D^{(\boldsymbol{s})} f\|_{L^p}\right)^{1/p}. \tag{4.21}$$

For a more compact notation, we write $W^{m,p}(\bar{\Omega})$ instead of $W^{m,p}(\bar{\Omega}, \mathbb{R})$ if the image space is $\mathbb{R}$. Analogously to $L^2(\bar{\Omega})$, the space $W^{m,2}(\bar{\Omega})$ is a Hilbert space with inner product

$$(f, g)_{W^{m,2}} \;=\; \sum_{|\boldsymbol{s}|_1 \leq m} (D^{(\boldsymbol{s})} f, D^{(\boldsymbol{s})} g)_{L^2}. \tag{4.22}$$

We use $H^m(\bar{\Omega})$ as a more compact notation for $W^{m,2}(\bar{\Omega})$. This smoothness class is relevant, for instance, when a finite element discretization on a full grid space $V_n^{(\infty)}$ of level n is used. With the required Hölder or Sobolev smoothness, a discretization scheme for the PDE problem (4.1) of order p with full grid mesh width h_n in all d dimensions achieves an accuracy with respect to the L^2- and L^∞-norm of

$$\|u - u_n\|_* = O(h_n^p). \tag{4.23}$$

Furthermore, we note that Hölder spaces and Sobolev spaces are related via the Sobolev embedding theorem:

Theorem 4.1 [Sobolev embedding theorem]
Let $\Omega \subset \mathbb{R}^d$ be open and $m, k \in \mathbb{N}$ with $m > k + \frac{d}{2}$. Then, any equivalence class $f \in H^m(\bar{\Omega})$ is represented by an element in $C^k(\bar{\Omega})$.

Proof: The proof is given as Theorem V.2.12 in the book by Werner [137] and as Theorem 8.13 in the book by Alt [1].

Roughly speaking, Theorem 4.1 states that with increasing dimensionality d the smoothness of a function for which the m-th weak derivative exists reduces compared to classical Hölder smoothness.

For sparse grid discretizations of level n on the sparse grid space $V_n^{(1)}$ defined in (4.14) the reduced computational complexity leads to stronger smoothness requirements of the solution. The inner product (4.22) is associated with Sobolev spaces $H^m(\bar{\Omega})$ of isotropic smoothness. A further inner product

$$(f, g)_{H^m_{\text{mix}}} = \sum_{|s|_\infty \leq m} (D^{(s)} f, D^{(s)} g)_{L^2} \tag{4.24}$$

motivates a different class of Hilbert spaces

$$H^m_{\text{mix}}(\bar{\Omega}) = \{f \in L^p(\bar{\Omega}) \mid \|f\|_{H^m_{\text{mix}}} < \infty\} \tag{4.25}$$

with a norm defined by $\|f\|_{H^m_{\text{mix}}} = (f, f)^{\frac{1}{2}}_{H^m_{\text{mix}}}$. These function spaces are termed as Sobolev spaces of mixed smoothness.

If the discretization scheme of order p, which has been used in (4.23) for the full grid space $V_n^{(\infty)}$, is now applied to the corresponding sparse grid space $V_n^{(1)}$, the accuracy is slightly reduced. More precisely, Bungartz et al. [19] show that the accuracy of the solution u_n^s on the sparse grid space $V_n^{(1)}$ reduces to

$$\|u - u_n^s\|_* = O(h_n^p \cdot \log(h_n)^{d-1}) \tag{4.26}$$

if u has sufficient mixed Sobolev smoothness. As can be seen by comparing (4.22) and (4.24), the smoothness requirement in H^m_{mix} is stronger than in H^m since $H^m_{\text{mix}} \subset H^m$. A better understanding of these stronger smoothness requirements can be obtained from Sobolev spaces H^s with a non-integer index $s \in \mathbb{R}$. These more general Sobolev spaces are also called *Bessel potential spaces*; see, for instance, Chapter 2.1 of Knapek [71] for a rigorous definition using a Fourier transformation. Here, the relation

$$H^{sd} \subset H^s_{\text{mix}} \subset H^s \subset H^{s/d}_{\text{mix}} \tag{4.27}$$

for $s \in \mathbb{R}^+$ can be derived. Equation (4.27) indicates that the mixed smoothness requirement for the sparse grids becomes the stronger, the higher the dimensionality d of the physical problem is.

We further note that Sobolev spaces of mixed smoothness can be represented as tensor product of Sobolev spaces of dimensionality one, i.e. we can relate $H^m_{\text{mix}}(\bar{\Omega})$ with $H^m([0,1]) \otimes \cdots \otimes H^m([0,1])$. For the PDE problem of the multi-bead-spring chain in Section 1.2 we require regularity in $H^1_{\text{mix}}(\bar{\Omega})$. Due to Theorem 4.1 this can be related to a tensor product of function spaces that have at least a representative in C^0. The continuity of the solution is essential for the sparse grid combination technique, which we describe in the following section, as the approach requires an interpolation of different solutions on a common solution space.

4.2. Sparse grid combination technique

An advantage of L^2-based sparse grids is their reduced number of degrees of freedom in the order of $O(N \cdot \log_2(N)^{d-1})$ compared to full grid spaces with a complexity of $O(N^d)$. This approach requires, however, a stronger regularity of the solution typically in $H^1_{\text{mix}}(\bar{\Omega})$ if $\mathcal{L}(\cdot)$ in (4.1) is a second-order

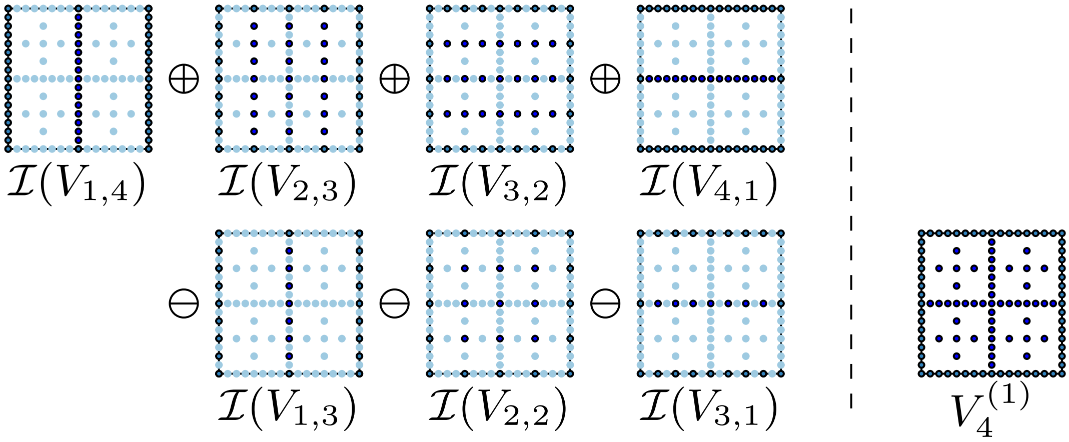


Figure 4.3.: Linear combination of bilinearly interpolated $2D$ full grid spaces $V_{i,j}$ and the corresponding discrete solutions $u_{i,j}$ according to the combination technique with $n = 4$. Interpolated points are colored in light blue.

differential operator. Furthermore, the discretization in $V_n^{(1)}$ requires a hierarchical data structure. Consequently, the solver has to be specifically adapted to hierarchical increment spaces.

In this section, we consider the *sparse grid combination technique*. This approach bases on multivariate extrapolation, cf. Bungartz, Griebel and Rüde [21]. It allows us to approximate the sparse grid solution in $V_n^{(1)}$ but has two immanent advantages:

- Solutions with a standard *nodal point basis* on full grid spaces $V_n^{(\infty)}$ as defined in (4.8) can be used in the combination technique. This is advantageous if PDE solvers such as *NaSt3DPGF* already exist for the specific application; cf. Chapter 3.

- The combination technique is intrinsically parallel. Since our application, the multiscale simulation of polymeric fluids, is extremely demanding with respect to its computational complexity, we have to make use of massively parallel computers.

The sparse grid combination technique was first introduced by Griebel, Schneider and Zenger [53]. For an illustration of the general principle, we first consider the PDE problem (4.1) in $d = 2$ dimensions. We discretize (4.1) on a full grid $V_{l,m}$ as in (4.7) and obtain a linear system of equations of the form

$$L_{l,m}(u_{l,m}) = f_{l,m} \qquad \text{in } V_{l,m} \tag{4.28}$$

with appropriate boundary conditions. The index pair $(l, m) \in \mathbb{N}^2$ uniquely identifies the discrete problem (4.28). A level n sparse grid solution in $V_n^{(1)}$ is then approximated in the combination space V_n^c by a multivariate extrapolation of several solutions according to

$$u_n^c = \sum_{l_1+l_2=n+1} u_{l_1,l_2} - \sum_{l_1+l_2=n} u_{l_1,l_2} \qquad \text{for } l_1, l_2 = 1, \ldots, n. \tag{4.29}$$

Equation (4.29) cannot be directly computed but requires an interpolation of all the different coarse and fine grid solutions u_{l_1,l_2} in V_{l_1,l_2} on a common grid. This grid is actually $V_n^{(1)}$ from (4.15). Fig. 4.3 illustrates the interpolation $\mathcal{I} : V_{l_1,l_2} \to V_4^{(1)}$ that is involved in (4.29) for the case $n = 4$. Only the grid points colored in dark blue are contained in V_{l_1,l_2}. The grid points that are contained in $V_4^{(1)}$ but not in V_{l_1,l_2} are colored in light blue.

In contrast to Fig. 4.2, the boundary conditions are now taken into account. At a first glance, this leads to a large number of grid points on the boundary compared to the inner domain. The grid points on the boundary of $V_4^{(1)}$ coincide with the boundary points of the full grid $V_4^{(\infty)}$ However, for a PDE problem the solution is prescribed on the boundary. In Chapter 6 we combine different finite difference solutions by using the combination technique. In this case, we only have to combine interior grid points since the solution on the boundary is identical on every grid V_{l_1,l_2} and only the solution on the interior grid is unknown. This is different, for instance, when the combination technique is applied to integration problems. In this case, an evaluation of the integrand on the boundary has the same complexity as in the inner domain.

Next, we consider the combination formula (4.29) for $d = 3$ and for general dimensions d. In the three-dimensional case, the combination formula is

$$u_n^c = \sum_{l_1+l_2+l_3=n+2} u_{l_1,l_2,l_3} - 2 \cdot \sum_{l_1+l_2+l_3=n+1} u_{l_1,l_2,l_3} + \sum_{l_1+l_2+l_3=n} u_{l_1,l_2,l_3} \qquad \text{for } l_1, l_2, l_3 = 1, \dots, n. \tag{4.30}$$

In the general d-dimensional case the combination approximation is given by

$$u_n^c = \sum_{i=0}^{d-1} (-1)^i \binom{d-1}{i} \sum_{|\boldsymbol{l}|_1 = n+(d-1)-i} u_{\boldsymbol{l}} \qquad \text{for } l_1, \dots, l_d = 1, \dots, n. \tag{4.31}$$

4.2.1. Error bounds for combination technique

Griebel, Schneider and Zenger [53] show that the combined solution u_n^c of (4.31) is identical to the sparse grid solution u_n^s for interpolation problems. This is due to the fact that the interpolant is identical on all common grid points independent of the mesh width. This is obviously not the case for PDE problems as the discrete solution at each grid point is associated with the specific mesh width. However, the additional PDE discretization error is of the same order as the approximation error as long as a certain error splitting exists. For this purpose, we require the following theorem:

Theorem 4.2
Let u be the solution of the PDE problem 4.1 and $u_{\boldsymbol{l}} \in V_{\boldsymbol{l}}$ a discrete solution for which pointwise an error expansion of the form

$$u(\boldsymbol{x_l}) - u_{\boldsymbol{l}}(\boldsymbol{x_l}) = \sum_{i=1}^{d} \sum_{\{j_1,\dots,j_i\} \subset \{1,\dots,d\}} c_{j_1,\dots,j_i}(\boldsymbol{x_l}, h_{j_1}, \dots, h_{j_i})\, h_{j_1}^p \cdot \dots \cdot h_{j_i}^p \tag{4.32}$$

exists for all $\boldsymbol{x_l} \in \Omega_{\boldsymbol{l}}$ with $l_i \leq n$ for $i = 1, \dots, d$. Furthermore, the coefficients $c_{j_1,\dots,j_i}(\boldsymbol{x_l})$ are pointwise bounded by K, i.e.

$$|c_{j_1,\dots,j_i}(\boldsymbol{x_l})| \leq K \qquad \forall\, 1 \leq i \leq d \text{ and } \forall\, \{j_1, \dots, j_i\} \subset \{1, \dots, d\}$$

for all $\boldsymbol{x_l} \in \Omega_{\boldsymbol{l}}$ with $l_i \leq n$ for $i = 1, \dots, d$. Then, the combination technique solution $u_n^c \in V_n^c$ from

(4.31) *fulfills the error estimate*

$$|u - u_n^c(\boldsymbol{x}_n)| \le \frac{2K}{(d-1)!}\begin{pmatrix}2^p+1\\2^{p-1}\end{pmatrix}^{d-1}(n+2(d-1))^{d-1}\,2^{-pn} \approx O\left(h_n^p \cdot \log_2(h_n)^{d-1}\right) \tag{4.33}$$

for all $\boldsymbol{x}_n \in \Omega_n$.

Proof: The proof is given as Theorem 5.4 in Reisinger [113]. Theorem 4.2 states that the accuracy of the combined solution u_n^c is of the same order as the accuracy of the classical sparse grid solution (4.26).

It remains to show that an expansion as in (4.32) exists for a given PDE problem and an appropriate discretization. Bungartz et al. [20] have shown such an expansion and have given a proof of the convergence of the combination technique for the 2D Laplace equation on the unit cube with Dirichlet boundary conditions. Furthermore, an error expansion has been shown by Reisinger [113] for finite difference discretizations of Poisson and advection-diffusion equations in d dimensions. This motivates the use of the combination technique for the multiscale bead-spring-chain system of (1.59)–(1.63) although such an expansion is not known in this case.

4.2.2. Analysis of computational complexity

In (4.10) we state that the complexity of the interior domain of $V_n^{(\infty)}$ is of order $O(N^d)$ with $N = 2^n$. Instead of that, the sparse grid estimate in $V_n^{(1)}$ has a complexity of order $O(N \cdot \log_2(N)^{d-1})$; cf. equation (4.15). We subsequently show that the combination technique has a computational complexity of the same order as the sparse grid approximation. Compared with the full grid approach in $V_n^{(\infty)}$, the combination formula (4.31) sums up $O(n^{d-1})$ problems of reduced complexity $\dim(V_{\boldsymbol{l}})$ with $|\boldsymbol{l}|_1 \ll nd$. We therefore obtain

$$|V_n^c| = \sum_{i=0}^{d-1}\sum_{|\boldsymbol{l}|_1=n+(d-1)-i}|V_{\boldsymbol{l}}| = \sum_{i=0}^{d-1}\sum_{|\boldsymbol{l}|_1=n+(d-1)-i}\sum_{\boldsymbol{k}\le\boldsymbol{l}}|W_{\boldsymbol{k}}| \tag{4.34a}$$

$$\approx \sum_{|\boldsymbol{l}|_1=n+(d-1)}|W_{\boldsymbol{l}}| + \text{ lower complexity spaces } W_{\boldsymbol{k}} \tag{4.34b}$$

$$\approx \begin{pmatrix}n+d-2\\d-1\end{pmatrix}2^{n-1} = 2^n \cdot \left(\frac{n^{d-1}}{(d-1)!} + O(n^{d-2})\right) \approx O(N \cdot \log_2(N)^{d-1}) \tag{4.34c}$$

where $N = 2^n$ and $|W_{\boldsymbol{l}}| = 2^{|\boldsymbol{l}|_1-d}$ have been used. We also note that there are $\binom{n+d-2}{d-1}$ ways to build the sum $n+d-1$ in (4.34b) with d positive integers.

As motivated in Fig. 4.3 the combination technique in (4.31) requires d-linear interpolation. Obviously, the complexity of the interpolation has to be of the same order as (4.34c) to ensure a global complexity of $O(N \cdot \log_2(N)^{d-1})$. This problem will be discussed in the following.

Interpolation on hierarchical increment space

We consider d-linear interpolation as required for the combination formula (4.31) in d-dimensional space. Theoretically, it is also possible to interpolate the different coarse grid solution to the finest full grid space $V_n^{(\infty)}$. This, however, involves additional grid points which are not contained in any coarse

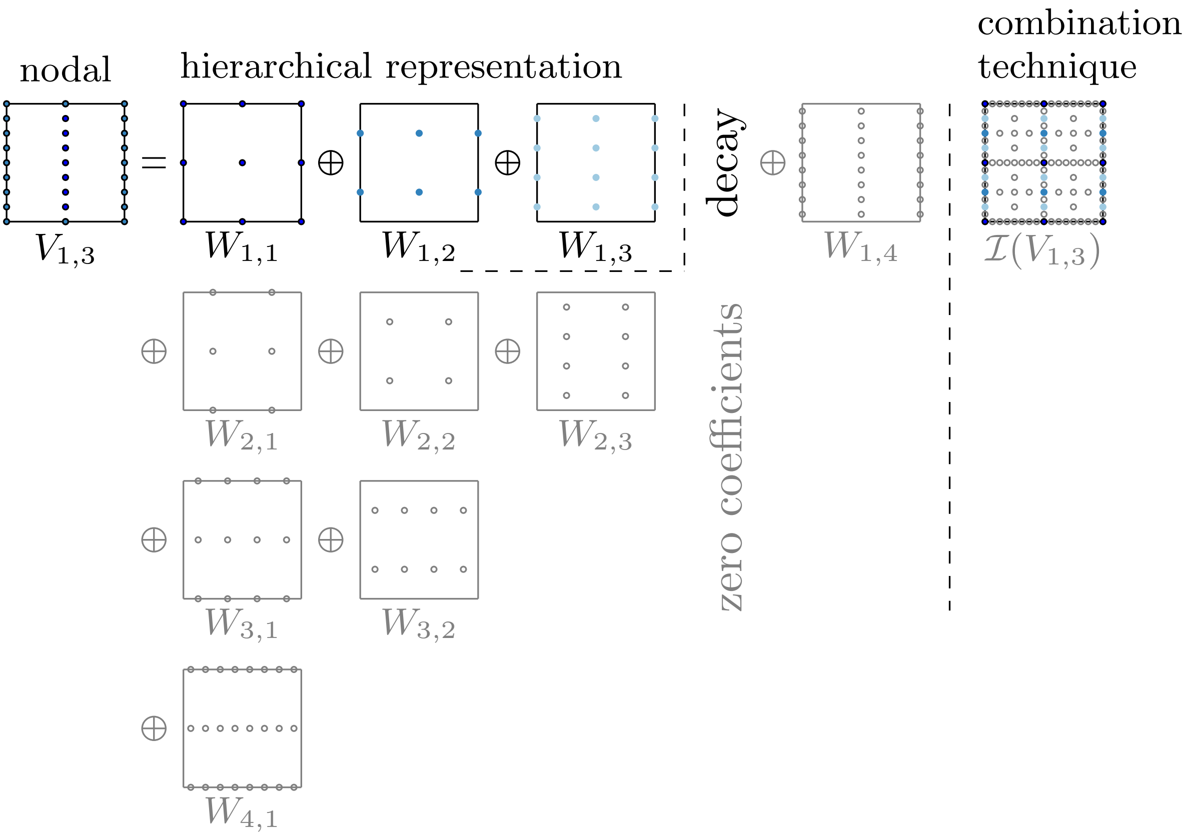


Figure 4.4.: Illustration of how a full grid solution is decomposed into a sequence of hierarchical increment spaces. For this purpose, the nodal values have to be transformed into the hierarchical surpluses. Increment spaces from $V_4^{(1)}$ without any contribution to $V_{1,3}$ are colored gray.

grid solution but on which, nevertheless, the combined solution has to be interpolated. Consequently, interpolation of $N = 2^n$ grid points in d dimensions is performed with $O(N^d)$ operations. As a result, the interpolation on the full grid space $V_n^{(\infty)}$ destroys the reduced complexity of (4.15) that is possible by using sparse grids. As a result, we have to make sure that the interpolation in (4.31) is performed with a complexity of $O(N \cdot \log_2(N)^{d-1})$ operations, i.e. the interpolation has to be performed on $V_n^{(1)}$, to ensure an overall complexity of this order.

The definition of $V_n^{(1)}$ in (4.14) bases on a sequence of hierarchical increment spaces. It is therefore most economical to first transform all the coarse full grid solutions on $V_{\boldsymbol{l}}$ with $|\boldsymbol{l}|_1 \leq n + (d-1)$ from a nodal basis to its hierarchical basis representation $\bigoplus_{\boldsymbol{k} \leq \boldsymbol{l}} W_{\boldsymbol{k}}$ as in (4.7) and then build the interpolated solution on $V_n^{(1)}$ according to $\bigoplus_{|\boldsymbol{k}|_1 \leq n+d-1} W_{\boldsymbol{k}}$ on the right-hand side of (4.14). The key idea is that the coefficients on the incremental spaces $W_{\boldsymbol{k}}$ that are contained in $V_n^{(1)}$ but not in $V_{\boldsymbol{l}}$ are zero. Therefore, no interpolation to new grid points is necessary. We illustrate this principle in Fig. 4.4 for $V_{1,3}$ from the 2-dim example in Fig. 4.3. Fig. 4.4 shows that only the hierarchical surplus for the increment spaces $W_{1,1}$, $W_{1,2}$ and $W_{1,3}$ have to be calculated. Since the coefficients of all other spaces are zero, we do not explicitly have to compute them. Since these spaces do not give any contribution in the hierarchical

Algorithm 3: Data transformation from nodal to its hierarchical representation.

```
   Data: Discrete data $u(\boldsymbol{x_{l,j}})$ on level $\boldsymbol{l} \in \mathbb{N}^d$ with mesh width $h_{\boldsymbol{l}}$ in nodal basis representation.
   Result: Data $u(\boldsymbol{x_{l,j}})$ in hierarchical basis representation as in $W_{\boldsymbol{k}}$ with $\boldsymbol{k} \le \boldsymbol{l}$.
 1 for $i \leftarrow 1$ to $d$ ;                                   /* loop over all dimensions */
 2 do
 3     for $k \leftarrow 1$ to $l_i$ ;           /* every dimension $i = 1, \dots, d$ has $l_i$ sublevels */
 4     do
 5         $h \leftarrow 2^{k-1}$ ;          /* distance of points on sublevel $k$ in dimension $i$ */
 6         forall the dimensions $\neq i$ pointwise ;        /* loop over every dimension $\neq i$ */
 7         do
 8             for $j_i \leftarrow h$ to $(2^{l_i} - h)$ step $2h$ do
 9                 $u(\dots, x_{k,j_i} \dots) \leftarrow u(\dots, x_{k,j_i} \dots) - \frac{1}{2}\left(u(\dots, x_{k,j_i-h} \dots) + u(\dots, x_{k,j_i+h} \dots)\right)$ ;
                   /* $u(\dots)$ indicates pointwise operation in all dimensions $\neq i$ */
10             end
11         end
12     end
13 end
```

basis, we have colored them in gray in Fig. 4.4 to emphasize this.

The algorithm for the transformation from a nodal basis to a hierarchical basis is given in Griebel and Thurner [54] for $d = 1$ and $d = 2$ dimensions. It can easily be generalized to d-dimensional spaces for a solution u on a grid $\boldsymbol{x_{l,j}} = (x_{l_1,j_1}, \dots, x_{l_d,j_d})$ by using Algorithm 3. The input of the algorithm is a discrete solution on a fine full grid $V_{\boldsymbol{l}}$ for which then the values in the hierarchical representation are constructed iteratively. The approach firstly determines the coefficients of the finest incremental space $W_{\boldsymbol{l}}$ and then continues with the coarser grids $W_{\boldsymbol{k}}$ with $|\boldsymbol{k}|_1 < |\boldsymbol{l}|_1$. For each index $\boldsymbol{k}$, a point that is contained in $W_{\boldsymbol{k}}$ but not in $W_{\boldsymbol{k}-\boldsymbol{e}_i}$, with $\boldsymbol{e}_i$ as unit vector in $\mathbb{R}^d$, the difference between the numerical solution at this grid point and the interpolant from the neighboring grid points is computed and stored in $W_{\boldsymbol{k}-\boldsymbol{e}_i}$, with $\boldsymbol{e}_i$; cf. line 9 of Algorithm 3.

The accuracy of d-linear interpolation is of order $O(h_{\boldsymbol{l}})$ as long as the solution is in $H^1_{\mathrm{mix}}(\bar{\Omega})$ and therefore continuous. For this reason, the interpolation error tends to decrease for increasing levels of the hierarchical increment spaces. As a consequence, the hierarchical increments in $W_{\boldsymbol{k}}$, the differences between linearly interpolated and actual grid values, also decrease. Furthermore, the decay of the coefficients in $W_{\boldsymbol{k}}$ is indicated in Fig. 4.4 by the color coding that reaches from dark blue to light blue for visualization purposes.

As a result, the transformation on the hierarchical basis has the same complexity of order $O(N \cdot \log_2(N)^{d-1})$ as the combination technique in equation (4.31). It is, however, obvious that the inverse transformation from the combination technique approximation on $V_n^{(1)}$ to the full grid solution on $V_n^{(\infty)}$ requires additional operations. The complexity then is again of order $O(N^d)$. Consequently, it depends on the application which kind of combination technique is more adequate for the current problem. For existing codes such as NaSt3DGPF it is usually better to work with the nodal basis and interpolate on $V_n^{(\infty)}$ since the post processing software such as ParaView requires a rectilinear data format.

For completeness, we note another variant of the combination technique, the *optimized combination technique* as developed by Garcke and Hegland[58, 41]. In this case, the combination coefficients $c_{\boldsymbol{l}}$ are not chosen according to equation (4.31) but instead specifically adapted to the underlying problem. For a given sequence of coarse full grid solutions, the optimized approach delivers the optimal combination formula. Moreover, the combined solution is the closest approximation to the sparse grid solution.

In the following section, we discuss the Multilevel Monte Carlo method (MLMC) and analyze its connection to sparse grid discretizations. This comparison leads to dimension-adaptive sparse grid algorithms. In Chapter 5, we present a dimension-adaptive algorithm for the combination technique to solve the coupled multiscale polymer flow problem.

4.3. Relation multilevel Monte Carlo - sparse grids

Sparse grids are one viable approach for the solution of the coupled multiscale system from (1.59)–(1.63). There exist other approaches in the literature for stochastic differential equations, for instance, the multilevel Monte Carlo (MLMC) method by Giles [47]. Interestingly, this approach can also be interpreted as a sparse grid approximation. Furthermore, the interpretation as a sparse grid approach allows stricter error bounds compared to MLMC in its classical formulation; see Gerstner and Heinz [45].

In the following section, we first state the basic principles of MLMC. Then, we give details on the relation between sparse grids discretizations on the one hand and the MLMC method on the other hand. In Chapter 5 we use the basic idea of MLMC, combining solutions on different temporal and stochastic grids, in our dimension-adaptive algorithm for the multiscale simulation of polymeric fluids.

4.3.1. Multilevel Monte Carlo method

As a problem of interest let

$$\mathrm{d}\boldsymbol{Q}(t) \;=\; \mu(\boldsymbol{Q},t)\,dt + \sigma(\boldsymbol{Q},t)\,\mathrm{d}\boldsymbol{W}(t) \tag{4.35}$$

be a general ordinary stochastic differential equation with $0 \le t \le T$, drift function $\mu : \mathbb{R}^n \times [0,T] \to \mathbb{R}^n$, volatility $\sigma : \mathbb{R}^n \times [0,T] \to \mathbb{R}^n$ and initial data $\boldsymbol{Q}(0) \in \mathbb{R}^n$; cf. equation (1.61). The quantity of interest is the expectation $\mathbb{E}[f(\boldsymbol{Q}(T))]$ with f as a scalar or vector-valued function with a Lipschitz bound. In our application, the expectation is the stress tensor from (1.62). In computational finance, the expectation is usually known as payoff.

The most simple approach to discretize (4.35) in time is the Euler-Maruyama method. This approach was used in (2.21) for the simple FENE-P and Hookean spring models. Employing the method with an equidistant timestep width h results in

$$\boldsymbol{Q}^{n+1} \;=\; \boldsymbol{Q}^n + \mu(\boldsymbol{Q}^n, t^n)\,h + \sigma(\boldsymbol{Q}^n, t^n)\,\Delta\boldsymbol{W}^n \tag{4.36}$$

with independent Wiener increments $\Delta\boldsymbol{W}^n$. Equation (4.36) is performed for $s = 1,\ldots,M$ samples $\boldsymbol{Q}^{n,(s)}$ until $t^n = T$. In the following, we denote the discrete samples at time T as $\boldsymbol{Q}^{T,(s)}$. Then, the resulting estimate for

$$\mathbb{E}[f(\boldsymbol{Q}(T))] \;=\; \int f(\boldsymbol{Q}(T,\omega))\,\mathrm{d}\boldsymbol{P}(\omega) \tag{4.37}$$

in which $\mathrm{d}\boldsymbol{P}(\omega)$ is the underlying probability measure of $\boldsymbol{Q}$ is

$$\boldsymbol{Y} \;=\; M^{-1}\sum_{s=1}^{M} f(\boldsymbol{Q}^{T,(s)}). \tag{4.38}$$

The MLMC method approximates $\mathbb{E}[f(\boldsymbol{Q}(T))]$ as a sequence

$$\hat{\boldsymbol{Y}} = \sum_{k=0}^{L} \hat{\boldsymbol{Y}}_k \tag{4.39}$$

of $k = 0, 1, \dots, L$ estimators. Equation (4.39) bases on the linearity of the expectation. The estimators differ in the timestep width h_k in (4.36) and in the number of stochastic samples M_k. More precisely, we set

$$h_k = K^{-k} T \tag{4.40}$$

with $K \in \mathbb{N}$, typically $K \in \{2, 3, 4\}$, and define the estimators as

$$\hat{\boldsymbol{Y}}_0 = M_0^{-1} \sum_{s=1}^{M_0} f(\boldsymbol{Q}_0^{T,(s)}) \qquad \text{for } k = 0, \tag{4.41}$$

$$\hat{\boldsymbol{Y}}_k = M_k^{-1} \sum_{s=1}^{M_k} \left(f(\boldsymbol{Q}_k^{T,(s)}) - f(\boldsymbol{Q}_{k-1}^{T,(s)}) \right) \qquad \text{for } k = 1, \dots, L. \tag{4.42}$$

We further note that the two function evaluations in equation (4.42) actually requires to evolve the same number of stochastic realizations by using the same Brownian path but with different timestep widths h_k and h_{k-1}. The key idea of MLMC is that the variance of (4.42) decreases with increasing level k such that less samples points are required.

Provided certain conditions on the drift μ and the diffusion σ are fulfilled, the expected mean-square error (MSE) can be decomposed as

$$\text{MSE} = \text{Var}(\hat{\boldsymbol{Y}}) + \mathbb{E}[\hat{\boldsymbol{Y}} - \mathbb{E}[f(\boldsymbol{Q}(T))]]^2. \tag{4.43}$$

The first error contribution in (4.43) is the variance in $\hat{\boldsymbol{Y}}$ that results from the Monte Carlo approximation in (4.41) and (4.42). The variance decreases linearly in the number of samples. The second contribution in (4.43) is the square of the bias due to the Euler-Maruyama discretization.

The aim of the MLMC method is to achieve a MSE for $\hat{\boldsymbol{Y}}$ less than ϵ^2 but with a lower computational complexity as for the classical estimator $\boldsymbol{Y}$ in (4.38). According to Giles [47], the optimal number of samples for given ϵ^2 is

$$M_k = \left\lceil 2\epsilon^{-2} \sqrt{\text{Var}(\hat{\boldsymbol{Y}}_k)\, h_k} \left(\sum_{k=0}^{L} \sqrt{\text{Var}(\hat{\boldsymbol{Y}}_k / h_k)} \right) \right\rceil. \tag{4.44}$$

The MLMC approach starts with one level, i.e. $L = 0$, and then proceeds as following (cf. Giles [47]):

1. Estimate the variance $\text{Var}(\hat{\boldsymbol{Y}}_L)$ on level L.
2. Compute the optimal number of samples M_k for $k = 0, \dots, L$ to ensure $\text{Var}(\hat{\boldsymbol{Y}}) < \frac{1}{2}\epsilon^2$.
3. If $\mathbb{E}[\hat{\boldsymbol{Y}} - \mathbb{E}[f(\boldsymbol{Q}(T))]] < \frac{1}{\sqrt{2}}\epsilon$, terminate the algorithm and otherwise set $L \leftarrow L + 1$ and go to 1.

After the algorithm has been terminated, the MSE is usually lower than ϵ^2 as indicated in (4.43). We further note that the restriction on the variance, i.e. $\text{Var}(\hat{\boldsymbol{Y}}) < \frac{1}{2}\epsilon^2$, in step 2 is enforced in every iteration step. However, this requires that the required number of samples M_k for $k = 1, \dots, L$ changes in every iteration step. Therefore, the required number of samples M_k might decrease with ongoing simulation. Then, since this additional but actually superfluous number of samples is not skipped, the decrease in variance might be larger than required. This guarantees a high reliability of the algorithm but reduces its efficiency.

For a MSE of $O(\epsilon^2)$, the classical Euler-Maruyama approach in (4.38) has a computational complexity of $O(\epsilon^{-3})$. This is due to a number of samples M in the order of $O(\epsilon^{-2})$ and a timestep width $h = O(\epsilon)$. In contrast to this, the MLMC complexity is of order $O(\epsilon^{-2}\log(\epsilon)^2)$ if the Euler-Maruyama scheme (4.36) is used. The complexity can be further reduced to $O(\epsilon^{-2})$ for higher-order schemes such as a Milstein discretization. A general result is given as Theorem 3.1 in Giles [47].

4.3.2. Interpretation of MLMC as sparse grid approximation

As explained in detail in the previous section, the MLMC method decomposes the expectation of a stochastic process and efficiently computes each partial sum using different timestep widths and sample numbers. We now discuss the connection of MLMC to sparse grid discretizations. For this purpose, we will interpret the different timestep widths and number of samples as different resolved discretizations of a sparse grid with two parameter dimensions. This relation was first annotated by Gerstner and Heinz [45] for problems with stochastic ordinary differential equations as in Section 4.3.1 and by Harbrecht, Peters and Siebenmorgen [56] for elliptic stochastic partial differential equations such as stochastic diffusion problems. A recent discussion on this topic is also given by Griebel, Harbrecht and Peters [49].

Equation (4.14) in Section 4.1 defines L^2-based sparse grids in d dimensions. For $d = 2$ the definition of the sparse grid space simplifies to

$$V_n^{\mathrm{sg}} = \bigoplus_{k_1+k_2\le n+1} W_{k_1}^{(1)} \otimes W_{k_2}^{(2)}. \tag{4.45}$$

Here, we write V_n^{sg} instead of $V_n^{(1)}$ as in (4.14) to avoid confusion with the one dimensional full grid spaces $V^{(1)}$ and $V^{(2)}$ that will be used in the following. Furthermore, in contrast to (4.14), we start with an index 0 in each coordinate direction since this resembles the MLMC indexing. Next, in view of equation (4.7) we can rewrite (4.45) as

$$V_n^{\mathrm{sg}} = \bigoplus_{k_1=0}^{n+1} W_{k_1}^{(1)} \otimes \left(\bigoplus_{k_2=0}^{n+1-k_1} W_{k_2}^{(2)} \right) = \bigoplus_{k=0}^{n+1} W_k^{(1)} \otimes V_{n+1-k}^{(2)}. \tag{4.46}$$

Analogously, we derive a second representation of V_n^{sg} as

$$V_n^{\mathrm{sg}} = \bigoplus_{k_2=0}^{n+1} \left(\bigoplus_{k_1=0}^{n+1-k_2} W_{k_1}^{(1)} \right) \otimes W_{k_2}^{(2)} = \bigoplus_{k=0}^{n+1} V_{n+1-k}^{(1)} \otimes W_k^{(2)}. \tag{4.47}$$

Both representations in (4.46) and in (4.47) are equivalent. We illustrate these different representations in Fig. 4.5; cf. Griebel, Harbrecht and Peters [49]. However, there is the advantage of the representation in (4.46) that the sequence of one-dimensional full grid spaces $\{V_k^{(2)}\}$ in the second dimensions does not need to be nested as the hierarchical space in (4.6). The same is valid for the sequence $\{V_k^{(1)}\}$ in (4.47).

Interestingly, the MLMC method in Section 4.3.1 uses a representation as in (4.47) with

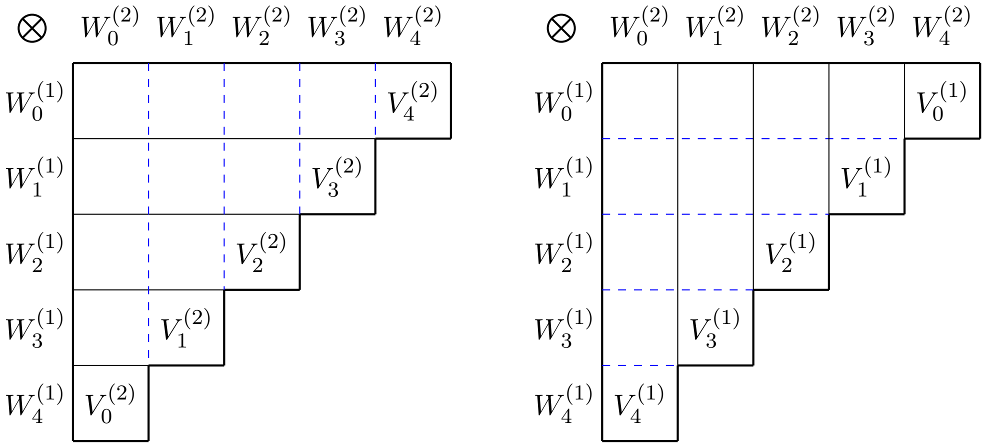

$\otimes$
$W_0^{(2)}$ $W_1^{(2)}$ $W_2^{(2)}$ $W_3^{(2)}$ $W_4^{(2)}$
$W_0^{(1)}$ $W_1^{(1)}$ $W_2^{(1)}$ $W_3^{(1)}$ $W_4^{(1)}$
$V_4^{(2)}$ $V_3^{(2)}$ $V_2^{(2)}$ $V_1^{(2)}$ $V_0^{(2)}$
$\otimes$
$W_0^{(2)}$ $W_1^{(2)}$ $W_2^{(2)}$ $W_3^{(2)}$ $W_4^{(2)}$
$W_0^{(1)}$ $W_1^{(1)}$ $W_2^{(1)}$ $W_3^{(1)}$ $W_4^{(1)}$
$V_0^{(1)}$ $V_1^{(1)}$ $V_2^{(1)}$ $V_3^{(1)}$ $V_4^{(1)}$

Figure 4.5.: Different representations of the sparse grid spaces $V^{\mathrm{sg}}_{k_1,k_2}$ with $k_1, k_2 \in \{1, \dots, 5\}$.

- $\{V_k^{(1)}\}$ as a sequence of Monte Carlo quadrature rules with increasing sampling number and
- $\{V_k^{(2)}\}$ as a sequence of temporal discretizations with decreasing timestep widths such that the complement spaces $W_{k+1}^{(2)} = V_{k+1}^{(2)} \ominus V_k^{(2)}$ with $V_{-1}^{(2)} = 0$ can be built.

More precisely, (4.39)–(4.42) is a decomposition of the expectation according to

$$\int f(\boldsymbol{Q}(T,\omega))\,\mathrm{d}\boldsymbol{P}(\omega) \approx \sum_{k=0}^{L} \mathrm{Qu}_k\left(\Delta f(\boldsymbol{Q}_k^T)\right) \tag{4.48}$$

with $\mathrm{Qu}_k : L^1(D, \mathcal{X}) \to \mathcal{X}$ as Monte Carlo quadrature rule for a suitable Bochner space $\mathcal{X}$ using M_k samples and with

$$\Delta f(\boldsymbol{Q}_k^T) = f(\boldsymbol{Q}_k^T - \boldsymbol{Q}_{k-1}^T) \quad \text{with } \boldsymbol{Q}_{-1}^T = 0. \tag{4.49}$$

However, the decomposition in (4.48) has not yet the form of (4.47) since $\{M_k\}_{k=0,\dots,L}$ as in (4.44) is not necessarily a decreasing sequence of sample numbers. In the following, we assume that $\{M_k\}_{k=0,\dots,L}$ decreases with k or otherwise perform a refinement of the coarser levels such that $M_0 \geq M_1 \geq \dots M_L$. Then, we set $\hat{\mathrm{Qu}}_{L-k} := \mathrm{Qu}_k$ for $k = 0, \dots, L$ and obtain

$$\int f(\boldsymbol{Q}(T,\omega))\,\mathrm{d}\boldsymbol{P}(\omega) \approx \sum_{k=0}^{L} \hat{\mathrm{Qu}}_{L-k}\left(\Delta f(\boldsymbol{Q}_k^T)\right) \tag{4.50}$$

which is similar to the splitting in (4.47). The sequence of quadrature rules $\{\hat{\mathrm{Qu}}_0, \dots, \hat{\mathrm{Qu}}_L\}$ is then defined on the full grid sequence $\{V_0^{(1)}, \dots, V_L^{(1)}\}$. As mentioned before, these spaces need not to be nested since the number of samples is defined by (4.44). On the other hand, the sequence of complement spaces $\{W_0^{(2)}, \dots, W_L^{(2)}\}$ for the temporal grids is nested. Here, the timestep width is $h_k = K^{-k}T$ for $k = 0, \dots, L$ with $K \in \mathbb{N}$.

Instead of estimating the full grid sequence $\{V_k^{(1)}\}$ according to (4.44), it is also possible to obtain a sequence from a dimension-adaptive algorithm. This approach will be considered for comparison in the following section.

4.3.3. Dimension-adaptive sparse grids

A dimension-adaptive sparse grid algorithm, closely related to the MLMC method, that copes with stochastic ODEs as in (4.35) is given by Gerstner and Heinz [45]. The difference to MLMC is that

- the sample point sequence $\{M_k\}_{k=0,\dots,L}$ is not computed according to (4.44) but starts with a low number of samples which is then adaptively refined and

- the contributions to the MSE in (4.43) from the variance and the bias are not both bounded by $\epsilon^2/2$. Instead of that only their sum is bounded by ϵ^2. This gives a higher degree of freedom for an optimal distribution of the error terms.

The adaptive algorithm by Gerstner and Heinz [45] starts on two levels, i.e. $L = 1$, and with a low initial number of samples on these levels M_0 and M_1, e.g. $M_0 = M_1 = 100$. The algorithm then continues as following:

1. Estimate the variances $\mathrm{Var}(\hat{\boldsymbol{Y}}_k)$ for $k = 0, \dots, L$. The MSE is then approximated as:
   - $\mathrm{Var}(\hat{\boldsymbol{Y}}) \approx \sum_{k=0}^{L} \mathrm{Var}(\hat{\boldsymbol{Y}}_k)$ (variance estimator)
   - $\mathbb{E}[\hat{\boldsymbol{Y}} - \mathbb{E}[f(\boldsymbol{Q}(T))]]^2 \approx (\hat{\boldsymbol{Y}}_L/(K^\alpha - 1))^2$ (bias estimator)

   with K as in (4.40) and with α as weak order of convergence.

2. The refinement then depends on the dominant MSE error contribution:
   - if the estimated variance is larger than the bias, we set $M_k \leftarrow 2M_k$ for $k \in \{1, \dots, L\}$ with the largest variance contribution in $\sum_{k=0}^{L} \mathrm{Var}(\hat{\boldsymbol{Y}}_k)$.
   - otherwise, if the bias is larger than the variance, set $L \leftarrow L + 1$ and start with a low number of initial samples on the new level.

3. If the estimated MSE as in (4.43) is lower than ϵ^2 the algorithm terminates; otherwise goto 1.

Gerstner and Heinz [45] compare this dimension-adaptive sparse grids algorithm with the classical MLMC algorithm from Section 4.3.1. Their conclusion for an option pricing problem is the following:

- The rate of convergence with respect to the variance error for MLMC and for their dimension-adaptive approach is $1/2$ which is expected for Monte Carlo quadrature.

- An initially given MSE ϵ is reached by both algorithms. However, MLMC usually has a lower error than requested. On the downside, this implies that MLMC is not optimal with respect to its efficiency.

- The adaptive refinement in MLMC is performed in one dimension only, i.e. for the timestep width. Instead of that, the dimension-adaptive sparse grid algorithm adaptively refines both dimensions, the timestep width and the number of samples, even though the quadrature is performed on full grid spaces. Consequently, the sparse grid algorithm requires more iteration steps than the MLMC algorithm for termination.

As a result, the decision between both algorithms depends on the underlying problem. If the variance estimation that is required in every iteration step has a high computational complexity, then MLMC is the method of choice since the number of refinement steps is comparatively low. On the other hand, since MLMC tends to overestimate the required number of samples the sparse grid algorithm tends to be more efficient with respect to the computational complexity.

Another aspect between both algorithms is the amount of knowledge about the underlying problem that has to be taken into account. MLMC depends on a reliable variance estimation to calculate the required number of samples for the quadrature grid. A dimension-adaptive sparse grid algorithm approximates the required grid accuracy in the form of (4.46) or (4.47) in several iteration steps. The refinement criterion as given by Gerstner and Heinz [45] is therefore only one possible choice. This allows a higher flexibility for more complex problems in which a refinement criterion is not directly clear. This flexibility is important for our applications in polymer physics which will be discussed in Chapter 5.

Relation to the combination technique

In the following chapter, we will apply the combination technique to the coupled multiscale system (1.59)–(1.63). For this purpose, we derive a dimension-adaptive algorithm for the combination technique to solve our multiscale polymer system. The resulting algorithm can be interpreted as a generalization of MLMC to problem dimensions $d > 2$. Furthermore, it turns out that a dimension-adaptive refinement allows for an optimal balancing of the different error contributions in the coupled multiscale system. As a result, we are able to perform multiscale simulations as in Chapter 3 with a much lower computational complexity.

5. Discretization of multiscale polymer model on sparse grids

In Chapter 4 we introduced sparse grids to cope with problems of high dimensionality. In the following, we apply sparse grids to our multiscale model in polymer physics. However, there are several problems and considerations that have to be taken into account:

- Is it possible to reuse the existing multiscale flow solver described in Chapter 2?
- Which are the important problem dimensions for the mathematical model and for the discretization?
- Can the computing time in the order of weeks that is observed in Chapter 3 be reduced?

In Section 4.2 we have already discussed the combination technique. This approach allows us to reuse our full grid multiscale algorithm described in Section 2.2 and gives a positive answer to the first question.

In our application, the combination technique can be used with anisotropic spatial, temporal and stochastic grids. Furthermore, the modeling accuracy can be interpreted as a further problem dimension. The major question is how to balance these different grids in the combination technique? The answer depends on the corresponding rates of convergence. The numerical schemes require a certain smoothness and a sufficient grid accuracy to reach their theoretical rates of convergence. Furthermore, it is not directly clear how the overall error is affected by the modeling accuracy of the polymeric molecules.

We show that a close to optimal balancing can be achieved with a dimension-adaptive refinement since it identifies the problem dimensions of major importance. Moreover, it can be used in situations in which the regularity in different problem dimensions is not clear. This gives a positive answer to the second question.

First in Section 5.1, we discuss general concepts of the dimension-adaptive combination technique. For our application, this approach can be considered as a generalization of the Multilevel Monte Carlo (MLMC) method by Giles [47] which balances the stochastic variance and the temporal error only. Then in Section 5.2, we apply the dimension-adaptive combination technique to our multiscale polymer model. The main result of this chapter is Algorithm 5 to allow for dimension-adaptive simulations in polymer physics.

The computing time of the dimension-adaptive combination technique will be analyzed in Chapter 6. It can be shown that our approach dramatically reduces the computing time which gives a positive answer to the last initial question.

5.1. Dimension-adaptive combination technique

In Section 4.1 we discuss the general concepts of sparse grid discretizations. It turns out that depending on the norm $\|\cdot\|_*$ the resulting index set $\mathcal{I}$ differs. Equation (4.13) states the index sets for the L^2 and

L^∞ estimate on the one hand and for the energy norm on the other hand.

It is intuitively clear that other norm estimates lead to different index sets $\mathcal{I}$. A first generalization of $\mathcal{I}$ can be achieved by using a weight vector $\boldsymbol{a} \in \mathbb{R}_+^d$ with $|\boldsymbol{a}|_1 = d$ that reflects the importance of different dimensions; see, for instance, Garcke and Griebel [42] and Gerstner and Griebel [43]. A further generalization results from an adaptively generated index set.

5.1.1. Generalized index sets

The dimension-adaptive combination technique iteratively builds a problem-dependent index set $\mathcal{I} \subset \mathbb{N}^d$. Then, the approximation space is built from all discrete grid spaces that are contained in $\mathcal{I}$. The building process is an optimization problem that can be interpreted as a binary knapsack problem; see Bungartz and Griebel [17]. The adaptive algorithm starts with a single coarse full grid solution and then adds further grids to $\mathcal{I}$ according to a specific refinement criterion. The choice of the criterion strongly affects the resulting index set $\mathcal{I}$. We will discuss the construction of $\mathcal{I}$ and the refinement criterion subsequently but in a first step assume that $\mathcal{I}$ is already known. Then, the dimension-adaptive combination technique for general index sets, see Hegland [57] and Gerstner and Griebel [43], is defined as

$$u_{\mathcal{I}}^c = \sum_{\boldsymbol{l}\in\mathcal{I}} \left(\sum_{\boldsymbol{z}=\boldsymbol{0}\in\mathbb{N}^d}^{(1,\ldots,1)} (-1)^{|\boldsymbol{z}|_1} \cdot \chi^{\mathcal{I}}(\boldsymbol{l}+\boldsymbol{z}) \right) u_{\boldsymbol{l}} = \sum_{\boldsymbol{l}} c_{\boldsymbol{l}} u_{\boldsymbol{l}}. \tag{5.1}$$

In this case, $\chi^{\mathcal{I}}$ is a characteristic function on the index set $\mathcal{I}$ according to

$$\chi^{\mathcal{I}}(\boldsymbol{l}) = \begin{cases} 1 & \text{if } \boldsymbol{l} \in \mathcal{I} \\ 0 & \text{else} \end{cases} \tag{5.2}$$

and $c_{\boldsymbol{l}}$ is the combination coefficient with

$$c_{\boldsymbol{l}} = \left(\sum_{\boldsymbol{z}=\boldsymbol{0}}^{\boldsymbol{1}} (-1)^{|\boldsymbol{z}|_1} \chi^{\mathcal{I}}(\boldsymbol{l}+\boldsymbol{z}) \right). \tag{5.3}$$

In Fig. 5.1 we give an example of such an index set for $d = 3$. Here, each cube represents a full grid solution in V_{l_1,l_2,l_3} and the box color specifies the corresponding combination coefficient according to (5.3). In total, 57 grids have a combination coefficient unequal to zero and are used in the combination formula (5.1).

We now give details on the construction of $\mathcal{I}$ in (5.1). Interestingly, not every index set can be used in the combination formula. The generalized index set has to ensure an admissibility condition or consistency conditions; see Griebel and Gerstner [44] and Plaskota [107]. It holds for an admissible index set $\mathcal{I}$: If $\boldsymbol{l} \in \mathcal{I}$ with $\boldsymbol{e}_j$ unit vector then

$$\boldsymbol{l} - \boldsymbol{e}_j \in \mathcal{I} \quad \text{for } 1 \le j \le d, \quad l_j > l_j^{\min} \ge 1. \tag{5.4}$$

Hegland [57] denotes such an index set as a *downset*. The corresponding generalized sparse grid space is

$$V_{\mathcal{I}}^{(s)} = \bigoplus_{\boldsymbol{l}\in\mathcal{I}} W_{\boldsymbol{l}}. \tag{5.5}$$

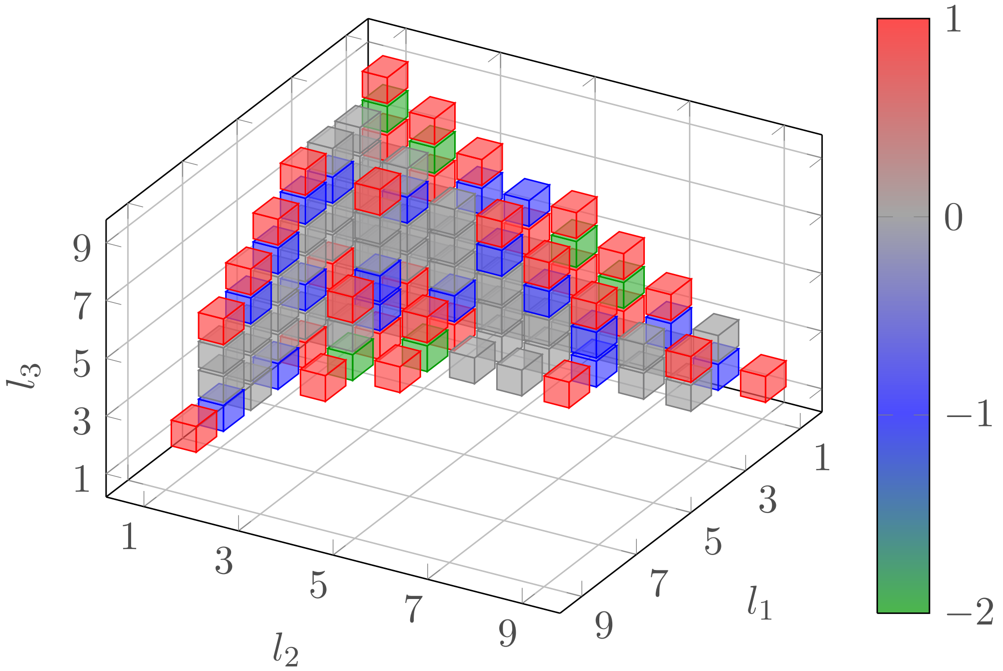


Figure 5.1.: A generalized index set $\mathcal{I}$ in three dimensions. The box color indicates the combination coefficient $c_{\boldsymbol{l}}$ according to (5.3).

We note that the admissibility condition (5.4) does not require that the coarsest grid of the approximation space $V_{(1,\ldots,1)}$ is contained in $\mathcal{I}$. In applications, however, this is usually the case. Furthermore, error estimates for general index sets are still an active area of research; see Wasilkowski and Wozniakowski [135]. For sufficient mixed Sobolev smoothness and a specific multivariate expansion of the discretization error, the error estimate for the classical combination technique is of the same order as for sparse grids defined in (4.26). Since an adaptive algorithm allows to detect the important dimensions, the accuracy of the adaptive algorithm is usually at least in the order of the classical algorithm or even better; see Gerstner and Griebel [44].

The algorithm starts with a single element in $\mathcal{I}$. We here assume, for simplicity, the usual case in which $\mathcal{I} = \{(1,\ldots,1)\} \subset \mathbb{N}^d$, i.e. the coarse grid solution contains only one interior grid point. The algorithm then adds additional indices $\boldsymbol{l}$, representing a full grid solution $u_{\boldsymbol{l}} \in V_{\boldsymbol{l}}$, to $\mathcal{I}$ with respect to the conditions that

- the extended index set remains admissible and
- the error of the combined solution decreases.

The first condition of admissibility can be ensured by an approach in which $\mathcal{I}$ is separated into two subsets $\mathcal{A}$ and $\mathcal{O}$ with $\mathcal{I} = \mathcal{O} \cup \mathcal{A}$ and $\mathcal{A} \cap \mathcal{O} = \emptyset$. The sets $\mathcal{A}$ and $\mathcal{O}$ are denoted as *active* and *old index set*, respectively.

In each adaptive step, an element $\boldsymbol{l} \in \mathcal{A}$ is selected that determines new grids $\boldsymbol{k} \notin \mathcal{I}$ whose numerical solution $u_{\boldsymbol{k}}$ might be computed and added to the combination formula. These new grids $\boldsymbol{k}$ have to fulfill two requirements:

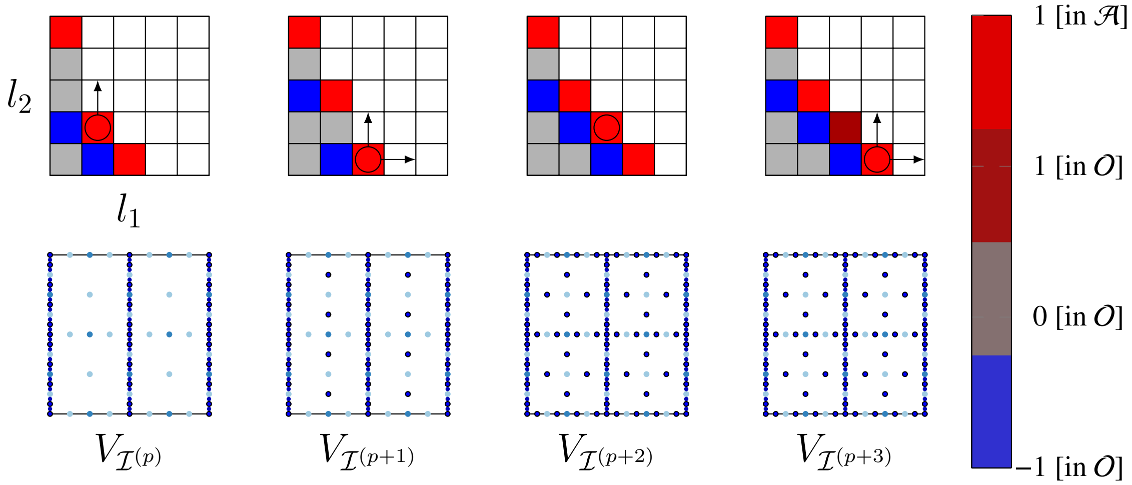


Figure 5.2.: Evolution of the dimension-adaptive algorithm (iteration step p up to step $p + 3$) for $d = 2$. The second row shows the nodes of the corresponding sparse grid space $V_{\mathcal{I}^{(p)}}$.

- $\boldsymbol{k}$ is adjacent to $\boldsymbol{l} \in \mathcal{A}$, i.e. $\exists i \in \{1, \dots, d\}$ with $\boldsymbol{k} = \boldsymbol{l} + \boldsymbol{e}_i$, and
- all backward neighbors of $\boldsymbol{k}$ are contained in O, i.e. $\boldsymbol{k} - \boldsymbol{e}_i \in O$ for all $i = 1, \dots, d$.

Then, up to d grids that fulfill both requirements are added to $\mathcal{A}$ and the selected $\boldsymbol{l} \in \mathcal{A}$ is removed from $\mathcal{A}$ and added to O. For a better illustration, the process of refinement with both subsets $\mathcal{A}$ and O is illustrated in Fig. 5.2 for the iteration steps p up to $p + 3$. Analogously to Fig. 5.1, all grids with a combination coefficient one are colored red and those with a coefficient of minus one are colored blue. Usually, the red colored cells belong to the active index set $\mathcal{A}$. However, in iteration step $p + 3$ the solution $u_{3,2}$ from the grid $\Omega_{3,2}$ is still combined with a combination coefficient one even though the index $(3, 2)$ does not belong to $\mathcal{A}$ any more. We indicate this difference with a different red tone in the colorbar on the right-hand side. Furthermore, cells which are colored blue and gray belong to the old index set O. In every iteration step, an index is selected for refinement. This index is indicated with a black circle. Moreover, two arrows indicate different possible directions of refinement in the current iteration step. In some cases, this does not lead to additional full grid solutions so that only an element from $\mathcal{A}$ is transferred to O. Furthermore, we note that the algorithm looses some of its efficiency in situations in which a refinement is performed in all d dimensions although refinement is actually only required in one dimension. Then, the remaining $d - 1$ full grid solutions are computed but only lead to small benefits.

In the following, we discuss how to select an adequate index $\boldsymbol{l} \in \mathcal{A}$ for the adaptive algorithm. This requires an error estimation for the solution on grid $\boldsymbol{l}$.

5.1.2. Error estimation in adaptive algorithm

For PDE problems in the form of (4.1) it is not guaranteed that the error decreases with each refinement step. It is therefore crucial to develop an adequate refinement criterion. For this purpose, we distinguish two different approaches for a posteriori error detection which are

- an *error estimator* that bases on a theory and allows a qualitative error description and
- an *error indicator* for which only a heuristic problem dependent approach exists.

In both cases, a local error $\epsilon_{\boldsymbol{l}}$ is associated with each $\boldsymbol{l}$ that might be added to $\mathcal{I}$. In our application, we approximate the size of the hierarchical surplus in the increment space $W_{\boldsymbol{l}}$ and use this as an error estimator. More precisely, we estimate the hierarchical surplus $w_{\boldsymbol{l}} \in W_{\boldsymbol{l}}$ in the combination technique by

$$\|w_{\boldsymbol{l}}\|_* \approx \|\otimes_{i=1}^d \Delta_i \boldsymbol{u_l}\|_* = \|\Delta_d(\otimes_{i=1}^{d-1}\Delta_i)\boldsymbol{u_l}\|_* = \|(\Delta_d \dots \Delta_1)\boldsymbol{u_l}\|_* \tag{5.6}$$

with the first-order difference operator

$$\Delta_i \boldsymbol{u_l} = \begin{cases} \boldsymbol{u_l} - \boldsymbol{u}_{\boldsymbol{l}-\boldsymbol{e}_i}, & \text{if } l_i > 1, \\ \boldsymbol{u_l}, & \text{if } l_i = 1, \end{cases} \tag{5.7}$$

and with $\boldsymbol{e}_i$ as canonical unit vector and $\boldsymbol{u_l} \in V_{\boldsymbol{l}}$ as full grid solution. Here, $\|\cdot\|_*$ is a suitable error norm depending on the application and on whether $\boldsymbol{u_l}$ is a scalar function, a vector or a tensor. Furthermore, we note that the approximation in (5.6) is exact for interpolation problems since the interpolant on two different meshes is identical on common grid points.

Next, we make two assumptions on the decay of expectation and variance of the hierarchical surplus approximation. A comparable decay is also assumed in Haji-Ali, Nobile and Tempone [55] for a problem related to groundwater flow. Note that the variance of the hierarchical surplus is caused by the stochastic nature of the polymeric structure. Since the stochastic microscopic scale couples with the macroscopic scale, all macroscopic unknowns also become random fields. However, as reported in Chapter 1, the stochastic variance of the velocity field is very small in practice. Depending on the rate of convergence, the expectation $\left\|\mathbb{E}\left(\otimes_{i=1}^d \Delta_i \boldsymbol{u_l}\right)\right\|_*$ decays according to

$$\left\|\mathbb{E}\left(\otimes_{i=1}^d \Delta_i \boldsymbol{u_l}\right)\right\|_* \leq C \prod_{i=1}^d \beta_i^{-l_i w_i} \tag{5.8}$$

with constants $C, \beta_i, w_i \in \mathbb{R}, \beta_i > 1$ and $w_i > 0$ for $= 1, \dots, d$. The constants β_i correspond to the rate of grid refinement and the constants w_i represent the order of convergence of the discretization scheme in dimension i. A similar assumption can be made for the variance of $\otimes_{i=1}^d \Delta_i \boldsymbol{u_l}$. Here, we assume a decay of $\operatorname{Var}\left(\otimes_{i=1}^d \Delta_i \boldsymbol{u_l}\right)$ according to

$$\operatorname{Var}\left(\otimes_{i=1}^d \Delta_i \boldsymbol{u_l}\right) \leq C_2 \prod_{i=1}^d \beta_i^{-l_i s_i} \tag{5.9}$$

with constants C_2 and $s_i > 0$ or $= 1, \dots, d$; cf. Haji-Ali, Nobile and Tempone [55].

Another aspect that is relevant for optimization is the computational effort, the work or cost that is required to compute each solution in $V_{\boldsymbol{l}}$. If $\epsilon_{\boldsymbol{l}}$ is an accurate estimate of the local error $\boldsymbol{l}$, we expect the largest error reduction in each extension step by adding this solution. However, with respect to the computational complexity this refinement criteria is not necessarily optimal. Other grids $\boldsymbol{k}$ might lead to a similar error reduction but have a much lower computational effort to obtain the numerical solution, especially if $\boldsymbol{k} \ll \boldsymbol{l}$. The complexity in elliptic PDE problems directly relates to the number of unknowns, for instance, to the number of interior grid points in an FD discretization. Time-dependent problems or problems with a stochastic component require a more general definition of the involved work $n_{\boldsymbol{l}}$.

The work $n_{\boldsymbol{l}}$ and the benefit $\epsilon_{\boldsymbol{l}}$ can be used to define a generalized local refinement indicator $g_{\boldsymbol{l}}$ as

$$g_{\boldsymbol{l}} = f(\epsilon_{\boldsymbol{l}}, n_{\boldsymbol{l}}) \tag{5.10}$$

with an unspecified function f that gives a weight between work $n_{\boldsymbol{l}}$ and benefit $\epsilon_{\boldsymbol{l}}$; see Griebel and Gerstner [44]. In this thesis, we discuss two possible choices for f that both will be applied to our specific PDE problem in Section 5.2. These choices are

- A *κ-weighted cost-benefit indicator*

$$g_{\boldsymbol{l}} = \max\left\{\kappa \cdot \frac{\epsilon_{\boldsymbol{l}}}{\epsilon_{(1,\ldots,1)}}, (1-\kappa) \cdot \frac{n_{(1,\ldots,1)}}{n_{\boldsymbol{l}}}\right\} \tag{5.11}$$

  with $\kappa \in [0,1]$. We both normalize the cost and the benefit with the corresponding values on the coarsest grid with index $\mathbf{1} = (1,\ldots,1)$ to make both values comparable. This approach balances the ratio of error reduction with the ratio of complexity increase with an additional parameter κ. Depending on the choice of κ, the indicator (5.11) becomes

$$\kappa = \begin{cases} 1, & \text{greedy approach,} \\ 0, & \text{classical combination technique} \\ \text{else,} & \kappa\text{-weighted combination technique.} \end{cases}$$

- A *profit indicator*

$$g_{\boldsymbol{l}} = \frac{\epsilon_{\boldsymbol{l}} \cdot n_{\mathbf{1}}}{\epsilon_{\mathbf{1}} \cdot n_{\boldsymbol{l}}} = C \cdot \frac{\epsilon_{\boldsymbol{l}}}{n_{\boldsymbol{l}}} \tag{5.12}$$

  with $C = \frac{n_{\mathbf{1}}}{\epsilon_{\boldsymbol{l}}} \in \mathbb{R}$. Since the cost and benefit on the coarsest grid lead to a scaling factor C that applies to every local indicator $g_{\boldsymbol{l}}$, it is assumed that $C = 1$ in the following without affecting the refinement procedure.

In addition to the local error indicators $\epsilon_{\boldsymbol{l}}$, we also require a global error indicator E. Apart from some simple benchmark experiments, an analytic solution or even the existence of a solution for the multiscale bead-spring chain is not known; see Section 1.2.4. Consequently, the exact error of the combined solution in (5.1) is also unknown. One approach to overcome this limitation is to measure the difference $\|u^c_{\mathcal{I}^{(p+1)}} - u^c_{\mathcal{I}^{(p)}}\|_*$ between two iterations in step p and $p+1$ with the corresponding index sets $\mathcal{I}^{(p+1)}$ and $\mathcal{I}^{(p)}$. If this difference is smaller than a given tolerance TOL more than r times in a row, the algorithm terminates. The r-times repetition of this criterion ensures that a short stagnation in the error reduction does not lead to a termination of the algorithm.

As a result, Algorithm 4 sums up the previous considerations. In the following section, we adapt Algorithm 4 specifically to the multiscale polymer flow model from Chapter 1.

Algorithm 4: Dimension-adaptive combination technique with generalized index sets.

```
Result: Solution $\boldsymbol{u}^c$ with error < TOL.
$\boldsymbol{l} := (1, \dots, 1)$;
$A := \{\boldsymbol{l}\}$ ;                                   /* active index set */
$O := \emptyset$ ;                                            /* old index set */
$g_{\boldsymbol{l}}$ ;                  /* local error indicator/ local profit indicator */
Determine global error indicator $E$ and error tolerance TOL;
while $E > TOL$ do
    select $\boldsymbol{l} \in A$ with largest $g_{\boldsymbol{l}}$ according to (5.11) or (5.12);
    $O = O \cup \{\boldsymbol{l}\}, \quad A = A \backslash \{\boldsymbol{l}\}$;
    for $t \leftarrow 1$ to $d$ do
        $\boldsymbol{j} = \boldsymbol{l} + \boldsymbol{e}_t$;
        if $\boldsymbol{j} - \boldsymbol{e}_l \in O \quad \forall\, l = 1, \dots, d$ then
            $A = A \cup \{\boldsymbol{j}\}$;
            Solve PDE problem (4.1) on full grid space $V_{\boldsymbol{j}}$;
            Compute local benefit/ profit indicator $g_{\boldsymbol{j}}$;
        end
    end
    Compute new global error indicator $E$;
end
Compute $\boldsymbol{u}^c$ on index set $\mathcal{I} = O \cup A$ according to (5.1) ;
```

5.2. Dimension-adaptivity for multiscale polymeric fluids

In this section, we develop a dimension-adaptive algorithm for the coupled multiscale system in (1.59)–(1.63) or its discrete equivalent in (2.21)–(2.28).

First, we identify the different problem dimensions of the discrete system. For this purpose, we use a very general interpretation of the term *problem dimension*. More precisely, we balance not only the discretization accuracy as in Haji-Ali et al. [55] but also take the modeling accuracy of the spring-chain system into account. As a result, the problem dimensions for our specific application are

1. *spatial grid*: the spatial discretization of the velocity field $\boldsymbol{u}(\boldsymbol{x}, t)$, the pressure field $p(\boldsymbol{x}, t)$, the stress tensor field $\boldsymbol{\tau}_p(\boldsymbol{x}, t)$ and the Brownian configuration field $\boldsymbol{Q}(\boldsymbol{x}, t)$ with respect to the flow space variable $\boldsymbol{x} \in \mathcal{O} \subset \mathbb{R}^3$ as described in Section 2.2.1. Here, the maximum numbers $i_{\max}$, $j_{\max}$ and $k_{\max}$ of finite difference grid cells in the coordinate directions x, y and z lead to an anisotropic three-dimensional spatial grid.

2. *temporal grid*: the temporal discretization of the coupled system up to the final time T is described in Section 2.2.2. Here, the same timestep width Δt is applied to the discrete flow equations for $\boldsymbol{u}$ and p and to the stochastic partial differential equation for $\boldsymbol{Q}$. Therefore, we obtain a one-dimensional grid with $T/\Delta t$ steps in time.

3. *stochastic grid*: the stochastic partial differential equation (2.21) or (2.22) for the Brownian configuration field and equation (2.23) for the equilibrium control variate are solved for M_s stochastic samples. The number of samples M_s is considered as a one-dimensional problem dimension.

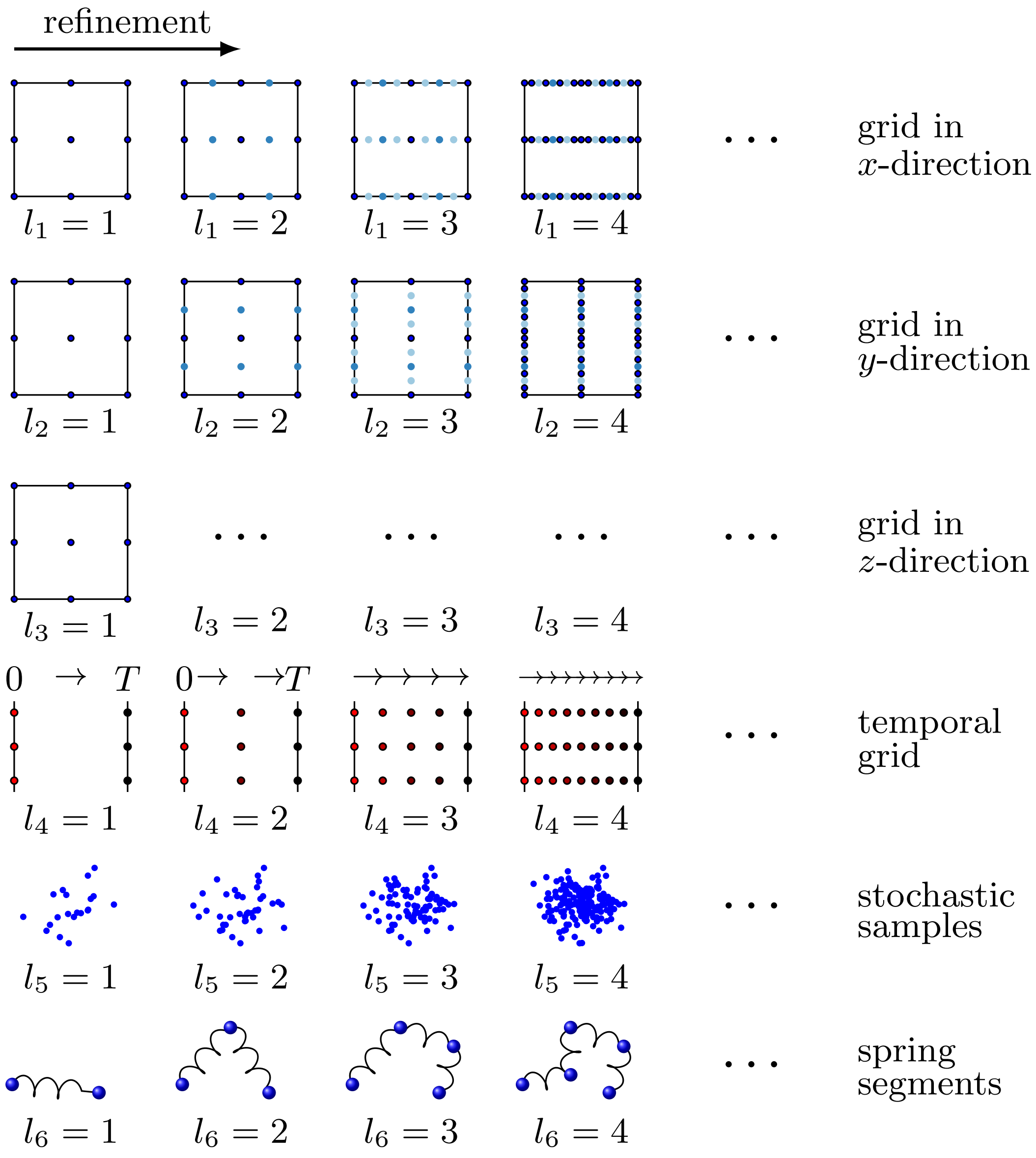


Figure 5.3.: Refinement of the index set $\boldsymbol{l} = (1, 1, 1, 1, 1, 1) \in \mathcal{I}$ in a six-dimensional parameter space.

4. *modeling grid*: the number of spring segments $N \in \mathbb{N}$ is also considered as a modeling dimension. As mentioned before, N determines the dimensionality of the configuration space $D \subset \mathbb{R}^{3N}$ and consequently the dimensionality of the stochastic samples. Note that although the configuration space is of high dimensionality it is a one-dimensional problem for the adaptive algorithm.

For a better understanding, we illustrate the different parameter dimensions of the six-dimensional index set $\mathcal{I}$ in Fig. 5.3. This parameter space is used for the applications in Chapter 6.

Furthermore, we note that the problem dimensions in the combination technique are not equivalent to the dimensions of physical problem in the coupled system (1.59)–(1.63) As described in Section 1.2, the coupled multiscale polymer system is of dimensionality $3N + 4$ since the physical space O is three-dimensional, the configuration space D is $3N$-dimensional and the system depends on time.

Due to the particle-based Monte Carlo approach for the configuration space described in Section 2.1, the configuration space D is sampled with realizations of dimension $3N$ which gives equal importance to every dimension of D. If the configuration space had been discretized with a grid-based approach, an anisotropic grid for the discretization of D in our dimension-adaptive algorithm would have been possible. In this case, a dimension-adaptive combination technique with even more parameter dimensions would have been possible which further reduces the computational complexity of the physical problem.

5.2.1. Effect of coupled system on adaptive refinement

In Section 4.1, sparse grid discretizations for a general PDE problem in the form of (4.1) have been introduced. The coupled multiscale polymer system (1.59)–(1.63) can be written in this form. The solution of the system is a tuple $(\boldsymbol{u}, p, \boldsymbol{Q}, \boldsymbol{\tau_p})$ but, depending on the application, we normally focus on a subset of $(\boldsymbol{u}, p, \boldsymbol{Q}, \boldsymbol{\tau_p})$:

- If the interest is on an exact description of the fluid flow, an accurate approximate of the velocity field $\boldsymbol{u}$ is the primary quantity of interest.

- If we are interested in elasticity effects due to the non-Newtonian behavior of the fluid, an exact description of the stress tensor $\boldsymbol{\tau}_p$ is searched for.

It is obviously clear that approximations $\boldsymbol{u}_{\mathcal{I}}^C$ and $\boldsymbol{\tau}_{p,\mathcal{I}}^C$ of $\boldsymbol{u}$ and $\boldsymbol{\tau}_p$ using formula (5.1) with index set $\mathcal{I}$ have to fulfill

$$\begin{cases} \boldsymbol{u}_{\mathcal{I}}^C & \to \boldsymbol{u} \\ \boldsymbol{\tau}_{p,\mathcal{I}}^C & \to \boldsymbol{\tau}_p \end{cases} \quad \text{for cardinality}(\mathcal{I}) \to \infty, \tag{5.13}$$

i.e. with increasing index set $\mathcal{I}$ the combined solution converges to the real solution. Furthermore, the dimension-adaptive algorithm tries to minimize the cardinality of $\mathcal{I}$ for a given error tolerance ϵ. Depending on the definition of ϵ, e.g. $\epsilon = \|\boldsymbol{u}_{\mathcal{I}}^C - \boldsymbol{u}\|_*$ or $\epsilon = \|\boldsymbol{\tau}_{p,\mathcal{I}}^C - \boldsymbol{\tau}_p\|_*$, the index set $\mathcal{I}$ differs. A dimension-adaptive approach primarily refines

- the spatial and temporal grid if the error in $\boldsymbol{u}_{\mathcal{I}}^C$ shall be reduced and

- the temporal, stochastic and spring-chain grid if an error reduction in $\boldsymbol{\tau_p}_{\mathcal{I}}^C$ is aimed for.

For the general case in which both fields are of interest, we use a weighted approximation of the hierarchical surpluses $\|W_{\boldsymbol{l}}(\boldsymbol{u})\|_*$ and $\|W_{\boldsymbol{l}}(\boldsymbol{\tau_p})\|_*$ as error indicator in the adaptive refinement strategy. In this case, $\|W_{\boldsymbol{l}}(\boldsymbol{u})\|_*$ approximates the hierarchical surplus of the velocity field $\boldsymbol{u}$ by using formula (5.6) and a suitable vector norm $\|\cdot\|_*$. The same definition applies to $\|W_{\boldsymbol{l}}(\boldsymbol{\tau_p})\|_*$ but in this case $\|\cdot\|_*$ refers to a suitable tensor norm for $\boldsymbol{\tau_p}$. Then, the local indicators in (5.11) and (5.12) modify to

$$g_{\boldsymbol{l}} = \max\left\{\kappa_1 \cdot \frac{\|W_{\boldsymbol{l}}(\boldsymbol{u})\|_*}{\|W_{(1,\dots,1)}(\boldsymbol{u})\|_*}, \kappa_2 \cdot \frac{\|W_{\boldsymbol{l}}(\boldsymbol{\tau_p})\|_*}{\|W_{(1,\dots,1)}(\boldsymbol{\tau_p})\|_*}, (1 - \max(\kappa_1, \kappa_2)) \cdot \frac{n_{(1,\dots,1)}}{n_{\boldsymbol{l}}}\right\}, \tag{5.14}$$

$$g_{\boldsymbol{l}} = \max\left\{\frac{\lambda \cdot \|W_{\boldsymbol{l}}(\boldsymbol{u})\|_*}{n_{\boldsymbol{l}}\,\|W_{(1,\dots,1)}(\boldsymbol{u})\|_*}, \frac{(1-\lambda)\cdot\|W_{\boldsymbol{l}}(\boldsymbol{\tau_p})\|_*)}{n_{\boldsymbol{l}}\,\|W_{(1,\dots,1)}(\boldsymbol{\tau_p})\|_*}\right\} \tag{5.15}$$

with $\kappa_1, \kappa_2, \lambda \in [0,1]$. Both refinement indicators balance the benefit with the involved work $n_{\boldsymbol{l}}$ that is associated with $\boldsymbol{l}$. Furthermore, λ in (5.15) balances the magnitude of the velocity field surplus with the magnitude of the stress tensor field surplus.

Adequate choice of the error norm

Up to now, the vector norm $||W_{\boldsymbol{l}}(\boldsymbol{u})||_*$ and the tensor norm $||W_{\boldsymbol{l}}(\boldsymbol{\tau_p})||_*$ in (5.14) and (5.15) have not been precisely defined. Next, we give details on an adequate choice of these norms in our application.

The accuracy of the velocity field $\boldsymbol{u}(\boldsymbol{x_i}, t_j) \in \mathbb{R}^3$ at fixed position in space-time $(\boldsymbol{x}_i, t_j)$ is measured according to one of the following two vector norms:

$$||\boldsymbol{u}(\boldsymbol{x}_i, t_j)||_2 = \left(\boldsymbol{u}(\boldsymbol{x}_i, t_j)^T \cdot \boldsymbol{u}(\boldsymbol{x}_i, t_j)\right)^{1/2} \qquad \text{(Euclidean norm)} \qquad (5.16)$$

$$||\boldsymbol{u}(\boldsymbol{x}_i, t_j)||_\infty = \max\left(|u(\boldsymbol{x}_i, t_j)|, |v(\boldsymbol{x}_i, t_j)|, |w(\boldsymbol{x}_i, t_j)|\right) \qquad \text{(comp.-wise maximum norm).} \qquad (5.17)$$

Usually, the velocity field has one dominant direction of flow. Then, the maximum norm in (5.17) measures the velocity component in flow direction. Moreover, since the solution is computed on a discrete space-time grid $(\boldsymbol{x}_i, t_j)$ with $i = 1, \ldots, I$ and $j = 1, \ldots, J$ we require a weighted average of the point-wise error terms in (5.16) and in (5.17). Consequently, for a discrete solution $\boldsymbol{u} \in V_{\boldsymbol{l}}$ we define the two error norms

$$||\boldsymbol{u}||_2 = \left(\sum_{(\boldsymbol{x}_i, t_j)} ||\boldsymbol{u}(\boldsymbol{x}_i, t_j)||_2^2 \, \Delta t_j \, \Delta \boldsymbol{x}_i\right)^{1/2} \qquad \text{(discrete } L^2\text{-norm of } \boldsymbol{u}) \qquad (5.18)$$

$$||\boldsymbol{u}||_{2,\infty} = \left(\sum_{(\boldsymbol{x}_i, t_j)} ||\boldsymbol{u}(\boldsymbol{x}_i, t_j)||_\infty^2 \, \Delta t_j \, \Delta \boldsymbol{x}_i\right)^{1/2} \qquad \text{(discrete } L^2\text{-norm of main flow direction)} \qquad (5.19)$$

that are used in the refinement procedure. Here, Δt_j denotes the timestep width and $\Delta \boldsymbol{x}_i$ the spatial resolution in $V_{\boldsymbol{l}}$.

The L^2-norms for the stress tensor $\boldsymbol{\tau_p} \in \mathbb{R}^{3\times 3}$ on a space-time grid $(\boldsymbol{x}_i, t_j)$ are defined analogously but instead of the vector norms (5.16) and (5.17) we now employ one of the matrix norms

$$||\boldsymbol{\tau_p}(\boldsymbol{x}_i, t_j)||_2 = \left(\boldsymbol{\tau_p}(\boldsymbol{x}_i, t_j) : \boldsymbol{\tau_p}(\boldsymbol{x}_i, t_j)\right)^{1/2} \qquad \text{(Frobenius norm)} \qquad (5.20)$$

$$||\boldsymbol{\tau_p}(\boldsymbol{x}_i, t_j)||_\infty = \max\left(|\tau_{xx}(\boldsymbol{x}_i, t_j)|, |\tau_{xy}(\boldsymbol{x}_i, t_j)|, \ldots, |\tau_{zz}(\boldsymbol{x}_i, t_j)|\right) \qquad \text{(maximum norm)} \qquad (5.21)$$

at each grid point $(\boldsymbol{x}_i, t_j)$. In (5.20) the term $\boldsymbol{\tau_p}(\boldsymbol{x}_i, t_j) : \boldsymbol{\tau_p}(\boldsymbol{x}_i, t_j)$ denotes the *double dot product* of $\boldsymbol{\tau_p}$, i.e. the squared sum of all entries of $\boldsymbol{\tau_p}$. Finally, we obtain as error norms for optimizing the stress tensor

$$||\boldsymbol{\tau_p}||_2 = \left(\sum_{(\boldsymbol{x}_i, t_j)} ||\boldsymbol{\tau_p}(\boldsymbol{x}_i, t_j)||_2^2 \, \Delta t_j \, \Delta \boldsymbol{x}_i\right)^{1/2} \qquad \text{(discrete } L^2\text{-norm of } \boldsymbol{\tau_p}) \qquad (5.22)$$

$$||\boldsymbol{\tau_p}||_{2,\infty} = \left(\sum_{(\boldsymbol{x}_i, t_j)} ||\boldsymbol{\tau_p}(\boldsymbol{x}_i, t_j)||_\infty^2 \, \Delta t_j \, \Delta \boldsymbol{x}_i\right)^{1/2} \qquad \text{(discrete } L^2\text{-norm of main stress component).} \qquad (5.23)$$

As mentioned before, an adequate choice of $\|\cdot\|_*$ is problem specific:

- If the problem has a dominant direction of flow, e.g. in a channel flow, we usually employ the maximum norms (5.19) for $\boldsymbol{u}$ or (5.23) for $\boldsymbol{\tau_p}$.
- In flow fields without a dominant flow direction, e.g. in rotational type flows, we employ the weighted error norms (5.18) or (5.22), respectively.

Estimation of involved work

Next, we consider the work or cost that is associated with each grid level $\boldsymbol{l}$. According to (4.10), the computational cost $m_{\boldsymbol{l}}$ of our spatial grid (l_1, l_2, l_3), see Fig. 5.3, is

$$m_{\boldsymbol{l}} = \prod_{i=1}^{3} (2^{l_i} - 1). \tag{5.24}$$

In this case, $m_{\boldsymbol{l}}$ denotes the number of inner grid cells for a PDE approximation.

In our setting, we employ the work required for all stochastic realizations $n_{\boldsymbol{l}}$ as a measure for the involved work. Loosely speaking, $n_{\boldsymbol{l}}$ is the product of the

- inner grid complexity $m_{\boldsymbol{l}}$ due to the spatial grid,
- time integration complexity due to the temporal grid,
- the number of samples M_s due to the stochastic grid and
- the dimensionality of the samples due to the modeling accuracy N.

Let $\mathcal{W}_s$ be the average work that is required to compute a single realization. The work $\mathcal{W}_s$ can be approximated as

$$\mathcal{W}_s = C_1 \lceil T/\Delta t \rceil \, 3N \tag{5.25}$$

floating point operations with $C_1 \in \mathbb{R}$. Equation (5.25) sums up that a sample is of dimensionality $3N$ and that the temporal scheme employs $\lceil T/\Delta t \rceil$ steps. The total number of floating point operations is then

$$n_{\boldsymbol{l}} = C_2 \, m_{\boldsymbol{l}} \, M_s \, \mathcal{W}_s \tag{5.26}$$

with $C_2 \in \mathbb{R}$ and $m_{\boldsymbol{l}}$ as in (5.24) with $d = 3$. Theoretically, the cost is $n_{(1,\ldots,1)} = 1$ on the initial level. In practice, however, there are some restrictions on the coarsest grids due to stability reasons. We discuss these restrictions in the following section on *Shifted index sets*. Some typical restrictions are

- a CFL-type stability restriction for the temporal grid l_4 and
- a minimum number of initial samples for the stochastic grid l_5.

The second restriction is, for instance, used in the dimension-adaptive sparse grid approach for stochastic ODEs by Gerstner and Heinz [45]; see Section 4.3.3. A sufficient number of initial samples on the coarsest level ensures that the variance of the solution does not lead to a stability breakdown of the numerical approach. As a consequence of these stability considerations, the number of time steps $N_T = \lceil T/\Delta t \rceil$ is of the form

$$N_T = C_T(2^{l_4} - 1) \tag{5.27}$$

with $C_T \in \mathbb{N}$. The parameter C_T is chosen such that stability issues are avoided in all discrete approximation spaces. We note that C_T often depends on the spatial grid accuracy such that $C_T = C_T(l_1, l_2, l_3)$.

Furthermore, the number of samples M_s is

$$M_s = C_s(2^{l_5} - 1) \tag{5.28}$$

with $C_S \in \mathbb{N}$ and we usually set $C_S = 100$. This choice corresponds with the minimum number of samples used by Gerstner and Heinz [45]. Moreover, the last index $l_6 = 1, 2, \ldots$ represents the number of spring segments N.

As a result, we can rewrite the complexity estimate (5.26) as

$$n_l = O\left(l_6 \prod_{i=1}^{5} (2^{l_i} - 1)\right). \tag{5.29}$$

Consequently, the cost increases linearly in the modeling dimension but exponentially in all numerical dimensions. The constants C, C_s, C_T in (5.29) are of no practical interest for the adaptive algorithm as they cancel out in the local indicator (5.14) and do not affect the maximum in (5.15).

Effect of nested sampling

Our dimension-adaptive approach refines different problem dimensions which are illustrated in Fig. 5.3 on page 120. In the figure, the index l_5 is used for the number of stochastic samples / Brownian configuration fields. Typically, this number doubles with every level increase. For a fixed level l_5, the corresponding samples are independent and identical distributed according to the probability density function ψ from Chapter 1. Nevertheless, the question arises if a correlation between the samples on different grid levels l_5 should be applied or not. In the following, we show that a correlation of the different stochastic levels leads to several advantages in our application.

First, we recapitulate that an average decay of the hierarchical surplus according to (5.8) is assumed. This condition, however, is not necessarily fulfilled for a single realization of $\otimes_{i=1}^{d} \Delta_i \boldsymbol{u_l}$ on a specific grid $\boldsymbol{l}$. For this reason, Giles [47] repeats his MLMC algorithm, described in Section 4.3.1, several times to demonstrate that the expected root mean square error can be achieved. As explained in Chapter 3, each simulation of a full multiscale flow problem requires months of computing time so that we cannot average our simulations that way. Therefore in our application, the quantities of interest are only averaged over time similar to the simulations by Vargas, Manero and Phillips [128].

We observe, however, that the variance of a single realization of $\otimes_{i=1}^{d} \Delta_i \boldsymbol{u_l}$ is reduced if the samples on different sampling levels l_5 are correlated. For this purpose, we use *nested samples* in the grid sequence. If, for instance, the coarsest grid $l_5 = 1$ consists of 100 sample particles, we reuse those 100 samples on the next finer grid $l_5 = 2$ with 200 realizations. More precisely, we reuse the initial configuration and apply the same Brownian motion to evolve the samples in time. In practice, this reduces the computation time since we just have to compute the samples on the finest grid, store them, and then use subsets on the coarser grid levels.

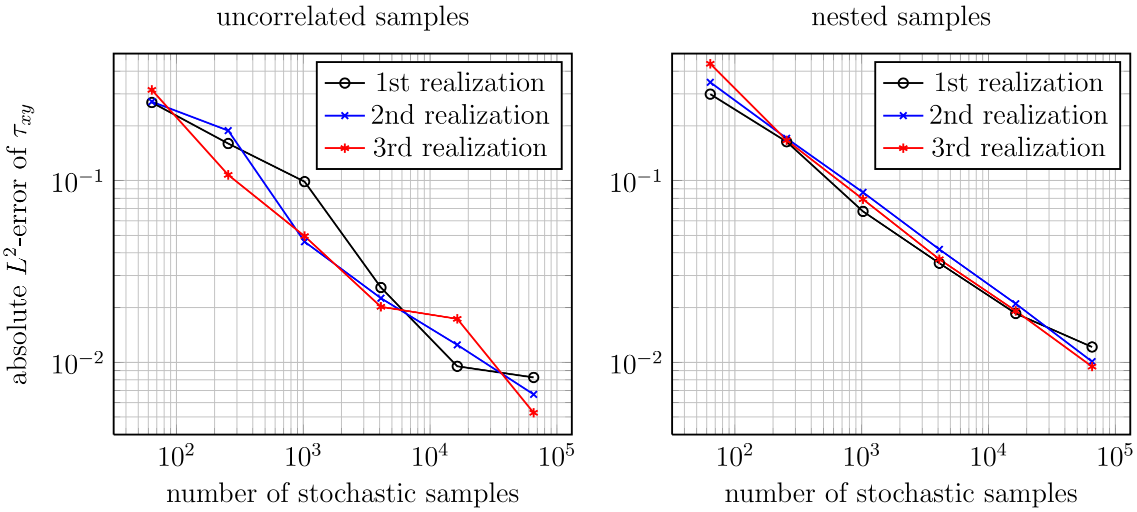


Figure 5.4.: Comparison of the τ_{xy} error decrease for uncorrelated and correlated samples in a steady Couette flow, cf. Section 6.1.1.

A further advantage of nested sampling is shown in Fig. 5.4. Here, we compare the absolute L^2-error of the τ_{xy} stress component depending on the number of stochastic samples in a steady Couette flow; see Section 6.1.1. On the left-hand side, we compare three independent runs of the algorithm for different sample numbers $M_{l_5} = 2^6, 2^8, \ldots, 2^{16}$. Note that the outcome on the left-hand side of Fig. 5.4 is interchangeable since all samples are uncorrelated. On the right-hand side, we perform three runs of the multiscale algorithm but use nested sampling points. In this case, all samples from the first realization (black solid line), from the second realization (blue solid line) and from the third realization (red solid line) are correlated. In both figures, the expected order of convergence is $p_{l_5} = 0.5$ for all levels of refinement l_5. But, we obtain a more monotonic error decrease for correlated samples. Obviously, this gives an eminent advantage in our dimension-adaptive approach since the variance according to Assumption 2 in (5.9) is reduced.

Furthermore, we note that the sample correlation between different levels depends on the rate of refinement. If we double the number of samples for each level l_5, consecutive levels which are required to build $\otimes_{i=1}^{d} \Delta_i \boldsymbol{u}_{\boldsymbol{l}}$ share 50% of the Brownian paths in the stochastic problem dimension.

Note that this kind of correlation is closely related to the correlation in the Brownian configuration field (BCF) method itself. As explained in detail for the BCF method in Section 2.2, the configuration fields evolve according to a Brownian motion that only depends on time but not on space such that their discrete Brownian motion is correlated. Our approach now extends this correlation to different grid resolutions $\boldsymbol{l}$ in the combination formula. As a result, this decreases the variance of the solution analogously to the BCF approach.

Shifting index set in the combination formula

As mentioned before, we apply the combination technique to multiscale polymer flows. It is known from previous results in the literature, see e.g. Griebel, Huber and Zenger [50] or Kranz [78], that very coarse grids in the spatial dimensions are unable to resolve vortex structures that occur, for instance, in contraction flow simulations as considered in Section 3.2.2. For this reason, several approaches exist

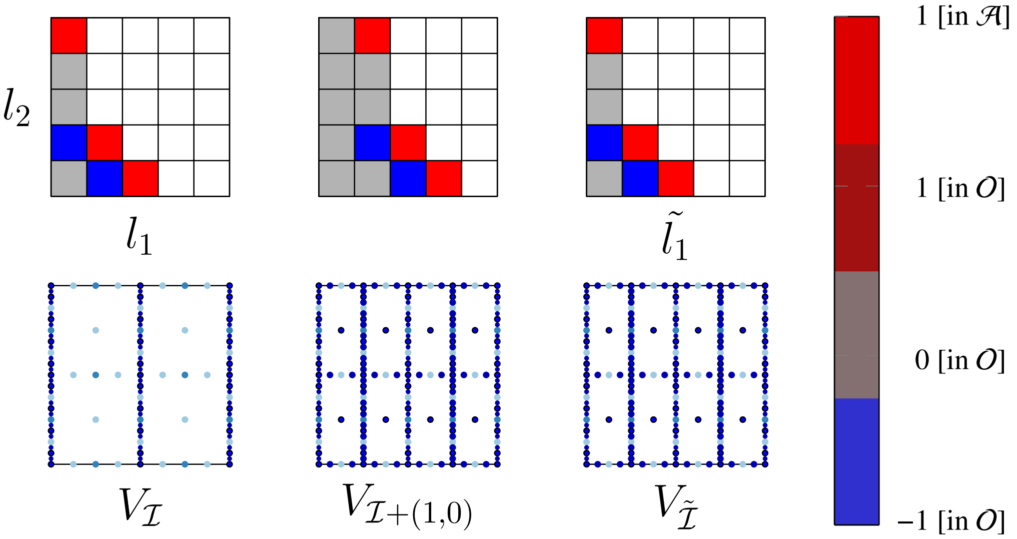


Figure 5.5.: Illustration of a constant index shift $\Delta \boldsymbol{l} = (1, 0)$ in the dimension-adaptive combination formula (top row) and the resulting sparse grid (bottom row).

to achieve a more suitable combination set which further modify the index set $\mathcal{I}$ in (5.1); see Kowitz et al. [77].

The main idea for optimizing the combination grid is to exclude anisotropic grid solutions with an insufficient accuracy in certain dimensions from our index set by applying a constant shift $\Delta \boldsymbol{l}$. In Fig 5.5 we illustrate the index set $\mathcal{I} + \Delta \boldsymbol{l}$ for a constant shift $\Delta \boldsymbol{l} = (1, 0, \dots, 0)$ in the first problem dimension. This index shift does not change the number of full grid solutions but increases their resolution. As a result, the adaptive combination formula (5.1) modifies to

$$u^c_{(\mathcal{I}+\Delta \boldsymbol{l})} = \sum_{\boldsymbol{l} \in (\mathcal{I}+\Delta \boldsymbol{l})} \left(\sum_{\boldsymbol{z}=\boldsymbol{0} \in \mathbb{N}^d}^{(1,\dots,1)} (-1)^{|\boldsymbol{z}|_1} \cdot \chi^{(\mathcal{I}+\Delta \boldsymbol{l})}(\boldsymbol{l} + \boldsymbol{z}) \right) u_{\boldsymbol{l}} \tag{5.30}$$

with a constant shift $\Delta \boldsymbol{l} \in \mathbb{N}^d$ and $\chi^{(\mathcal{I}+\Delta \boldsymbol{l})}$ as characteristic function on the shifted index set. For simplicity in practical applications, see Chapter 6, we often denote the coarsest full grid solution with the index $(1, \dots, 1)$ even if that solution is already refined. In that case, we actually consider a renamed index set $\tilde{\mathcal{I}}$ that differs from $\mathcal{I}$ just by the index shift $\Delta \boldsymbol{l}$. Due to reasons of simplicity in Chapter 6, we still denote the index sets with $\mathcal{I}$ even if they have been shifted in certain dimensions.

As expected, the size of a level shift depends on the specific problem. We note that our multiscale flow solver *NaSt3DGPF*, for instance, requires a minimum number number of grid cells in space for the discretization of the convective velocity terms. Furthermore, *NaSt3DGPF* is semi-implicit in time such that the maximum timestep width is restricted by a CFL-condition. Therefore, the simulations in Chapter 6 require more than one interior grid point.

5.2.2. Sparse dimension-adaptive algorithm

Algorithm 5: Dimension-adaptive algorithm for multiscale simulation of polymeric fluids.

```
   Result: Solution $(\boldsymbol{u}_{\mathcal{I}}^C, \boldsymbol{\tau_p}_{\mathcal{I}}^C)$ with $L^2$-error < TOL.
1  Select important problem dimensions $d = 1, \dots, 6$ ;          /* compare with Fig. 5.3 */
2  Shift grid indices with $\Delta\boldsymbol{l} \in \mathbb{N}^d$ as in (5.30) ;    /* avoid coarse full grid solutions */
3  Determine maximum resolution $\boldsymbol{l}_{\max}$ ;          /* e.g. restriction on modeling grid */
4  $\boldsymbol{l} := (1, \dots, 1)$ ;  /* coarsest full grid solution that resolves flow structure */
5  $A := \{\boldsymbol{l}\}$ ;                                        /* active index set */
6  $O := \emptyset$ ;                                                 /* old index set */
7  Select $g_{\boldsymbol{l}}$ according to (5.14) or (5.15) ;          /* local benefit/ profit indicator */
8  Determine global error indicator $E$ and error tolerance TOL;
9  while $E > TOL$ do
10     select $\boldsymbol{l} \in A$ with largest $g_{\boldsymbol{l}}$ according to (5.14) or (5.15);
11     $O = O \cup \{\boldsymbol{l}\}, \quad A = A \backslash \{\boldsymbol{l}\}$;
12     for $t \leftarrow 1$ to $d$ do
13         $\boldsymbol{j} = \boldsymbol{l} + \boldsymbol{e}_t$;
14         if $\boldsymbol{j} - \boldsymbol{e}_l \in O \quad \forall\, l = 1, \dots, d$ and if;
15         $\boldsymbol{j} \leq \boldsymbol{l}_{max}$ then
16             $A = A \cup \{\boldsymbol{j}\}$;
17             Apply multiscale flow solver from Algorithm 2 on full grid space $V_{\boldsymbol{j}}$;
18             Compute local benefit/ profit indicator $g_{\boldsymbol{j}}$;
19         end
20     end
21     Compute new global error indicator $E$ ;     /* terminate algorithm if $r$-times below
       TOL */
22 end
23 Compute $(\boldsymbol{u}_{\mathcal{I}}^C, \boldsymbol{\tau_p}_{\mathcal{I}}^C)$ on shifted index set $\mathcal{I} = O \cup A$ according to (5.30);
```

Finally, we present Algorithm 5 which modifies Algorithm 4 such that it allows for multiscale polymeric flow simulations. Using this algorithm, we obtain the numerical results in Chapter 6.

We note that the refinement procedure does not necessarily converge in every case. A typical problem in an dimension-adaptive algorithm is the following: If $\|u_{\boldsymbol{l}+\boldsymbol{e}_i} - u_{\boldsymbol{l}}\|_* \approx 0$ for any $i \in \{1, \dots, d\}$, then the algorithm terminates even if the global error is still large. This situation can occur due to a non-monotonic error decrease as illustrated in Fig. 5.4.

In these situations, the robustness of the algorithm can be increased by extending the index shift $\Delta\boldsymbol{l}$ in the combination formula (5.30), by using the L^∞-norm in (5.19) instead of the L^2-norm in (5.18) in the refinement procedure, or by requiring that the global error is below the threshold TOL r-times in a row as indicated in line 21 of Algorithm 5.

In Fig. 5.6, we illustrate the effect of an index shift on the cardinality of a two-dimensional index set $\mathcal{I}$. For this example, we assume an isotropic increase of the computational complexity in both dimensions and that we are able to compute full grid solutions up to level $l_1 + l_2 \geq 9$. After that, a further refinement becomes too expensive. First, the index set $\mathcal{I}$ contains 36 index pairs in the combination

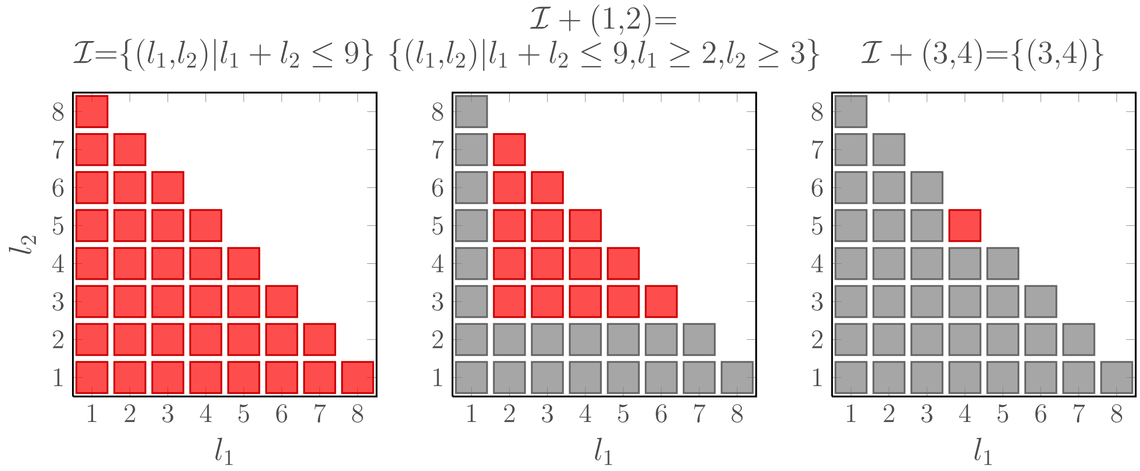

$\mathcal{I}=\{(l_1,l_2)|l_1+l_2\leq 9\}$
$\mathcal{I}+(1,2)=\{(l_1,l_2)|l_1+l_2\leq 9, l_1\geq 2, l_2\geq 3\}$
$\mathcal{I}+(3,4)=\{(3,4)\}$
l_2
l_1
1 2 3 4 5 6 7 8

Figure 5.6.: Comparison of the number of index pairs in a 2D index set (colored in red) for different index shifts. An extreme case is a single full grid solution in which the combination formula has maximum stability but looses all of its efficiency. However, this extreme situation does not occur in our simulations in the following chapter.

formula (5.30). Next, a constant shift $\mathcal{I}+(1,2)$ reduces the cardinality of $\mathcal{I}$ to 15 grids, cf. central figure in Fig. 5.6. On the one hand, this shift increases the stability of the approach since anisotropic index pairs can be avoided that otherwise that might lead to instabilities in the combined solution. On the other hand, this decreases the efficiency of the approach since the combination formula is applied to a smaller index set. An extreme example is shown on the right-hand side of Fig. 5.6. Here, a further index shift has reduced the elements in $\mathcal{I}$ such that only one full grid solution $V_{4,5}$ remains. Provided the numerical scheme converges, a single approximation space is always stable but does not obtain any benefit from the combination formula (5.30).

As a result, the example in Fig. 5.6 shows that numerical stability of Algorithm 5 can always be achieved provided its efficiency is sacrificed. This extreme example, however, does not occur in our simulations in the following Chapter 6. There, we apply Algorithm 5 to polymeric fluid flows. Furthermore, we show that our adaptive algorithm always reduces the computational complexity and allows for much more refined simulations.

6. Numerical results on sparse grids

This chapter shows the advantages of our dimension-adaptive sparse grid approach. In the following, we concentrate on two important applications which are

- shear-type flows/ Couette flows in Section 6.1 and
- extensional-type flows in Section 6.2.

Both types of flow are often considered in rheology since they allow for a polymeric fluid characterization. Furthermore, all problem dimensions that have been identified in Fig. 5.3 for optimization are used in the following applications. These problem dimensions include anisotropic

- spatial grids,
- temporal grids,
- stochastic grids and
- modeling grids.

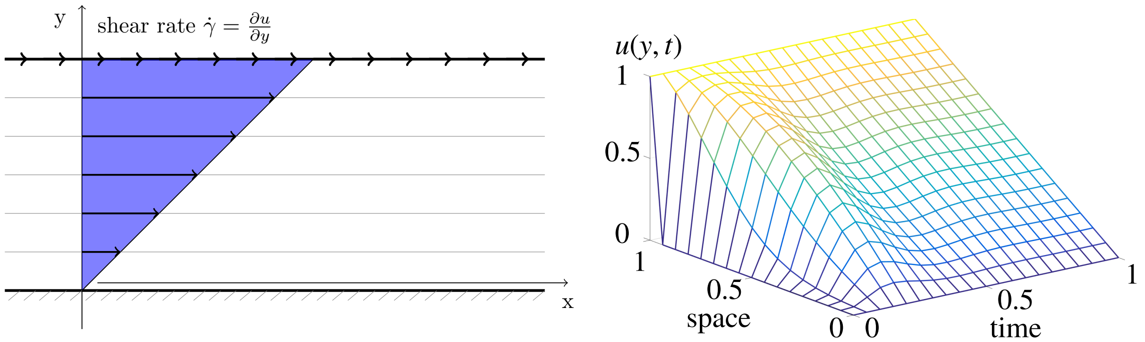
We investigate the interaction of these different dimensions and show that dimension-adaptive solvers have a lower computational complexity than classical full grid solvers. Using our approach, it is possible to reduce the curse of dimensionality that often occurs in this type of application. For all applications in Section 6.1 and Section 6.2 we explicitly state the reduction of complexity that can be achieved.

Another interesting aspect which will be analyzed in the following is that the process of dimension-adaptive refinement only has to be applied once to a specific class of fluid problems. The resulting combined grids / index set for such a class can be used for further simulations with modified flow conditions provided that the solution continuously depends on the modified parameters. We will investigate this effect in Section 6.1.3. In such situations, the adaptive refinement does not lead to any overhead compared to the classical combination technique in d-dimensional problem space.

6.1. Couette flows

A Couette flow occurs in the fluid volume between two parallel plates that move relative to each other; see Fig. 6.1 (a). In the non-Newtonian case, viscous and elastic forces act on the fluid and lead to a fluid flow in the direction of the moving plate. As the flow is assumed to be laminar, we only have to consider one velocity component of the two dimensional velocity field $\boldsymbol{u} \in \mathcal{O} \subset \mathbb{R}^2$, i.e. $\boldsymbol{u} = u(x, y, t)\,\boldsymbol{e}_x = u(y, t)\,\boldsymbol{e}_x$ (continuity equation) with $\boldsymbol{e}_x$ as canonical unit vector in flow direction.

In the beginning for $t = 0$, the fluid is at rest. Then, for $t > 0$ the upper plate at $y = L$ moves with a constant velocity V. In the Newtonian case, the fluid velocity u monotonically increases such that a constant velocity gradient $\frac{\partial u}{\partial y} = V/L := \dot{\gamma}$ occurs at steady state. In the non-Newtonian case, the



(a) shear flow at steady state

(b) $u(y,t)$ in non-Newtonian case

Figure 6.1.: Illustration of a shear flow at steady state (left) and visualization of the velocity component u over space and time in the non-Newtonian case (right).

steady state result is identical to the Newtonian case in Fig. 6.1 (a) but the temporal evolution differs. Fig. 6.1 (b) visualizes the temporal evolution of $u(y,t)$ for $t \in [0,1]$ and $V = L = 1.0$.

For a Couette flow, the complex multiscale system (1.59)–(1.62) simplifies to

$$\frac{\partial u(y,t)}{\partial t} = \frac{1}{Re}\beta\frac{\partial^2 u(y,t)}{\partial y^2} + \frac{1}{Re}\frac{\partial \tau_{xy}(y,t)}{\partial y} \tag{6.1}$$

$$\mathrm{d}\boldsymbol{Q}(y,t) = \left[(\nabla \boldsymbol{u}(y,t))^T \boldsymbol{Q}(y,t) - \frac{1}{2\,De}\boldsymbol{Q}(y,t)\right]\mathrm{d}t + \frac{1}{\sqrt{De}}\mathrm{d}\boldsymbol{W}(t) \tag{6.2}$$

$$\tau_{xy}(y,t) = \frac{1-\beta}{De}\,\mathbb{E}[Q_x(y,t)\,Q_y(y,t)] \tag{6.3}$$

with $\boldsymbol{Q}(y,t) = (Q_x, Q_y, Q_z)(y,t) \in D \subset \mathbb{R}^3$ and with De as Deborah number, Re as Reynolds number and β as viscosity ratio as defined in (1.10). Furthermore, the initial and boundary conditions are

$$\boldsymbol{u}(y,0) = 0, \quad \boldsymbol{u}(0,t) = 0, \quad \boldsymbol{u}(1,t) = V \tag{6.4a}$$

$$\tau_{xy}(y,0) = 0, \quad \partial\tau_{xy}(0,t)/\partial y = \partial\tau_{xy}(L,t)/\partial y = 0 \tag{6.4b}$$

$$\boldsymbol{Q}(y,0) \sim \psi_{eq}(\boldsymbol{q}) = \frac{1}{(2\pi)^{3/2}}\exp\left(-\frac{\|\boldsymbol{q}\|^2}{2}\right) \tag{6.4c}$$

for all $t \in [0,1]$ and for all $y \in]0, L[$.

We further note that the simplified system (6.1)–(6.3) describes a non-Newtonian fluid using a Hookean dumbbell model; cf. (1.43a). The system (6.2)–(6.3) can be rewritten into a constitutive equation for τ_{xy} which leads to the Oldroyd-B model defined in (1.4) and (1.5). Here, we use the Oldroyd-B model for comparison purposes to obtain a reference solution. Fig. 6.2 visualizes $u(y = 0.5)$ and $\tau_{xy}(y = 0.5)$ over time in the center of a channel with height $L = 1.0$. The corresponding fluid parameters are $Re = 0.1$, $De = 0.5$ and $\beta = 0.1$. Unless stated otherwise, all simulations in this section employ these parameters. In Fig. 6.2, we observe two velocity overshoots for $u(y = 0.5)$ at about $t \approx 0.2$ and $t \approx 0.65$. These overshoots and the undershoot at $t \approx 0.45$ are a result of the elastic fluid behavior that leads to a viscosity change. Due to the delayed increase of τ_{xy}, the effective viscosity in

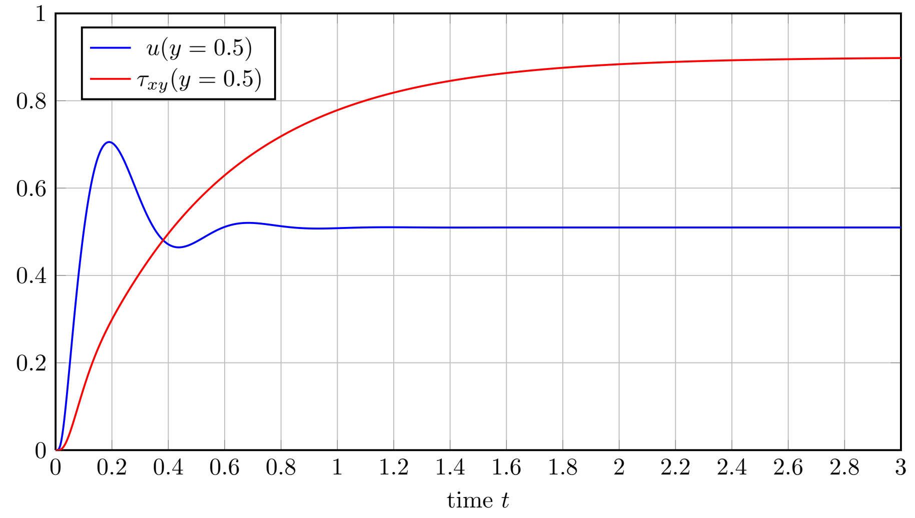


Figure 6.2.: Velocity component u and stress component τ_{xy} over time in a non-Newtonian Couette flow.

the beginning is lower than at steady state. This explains why the maximum velocity occurs at $t = 0.2$ and not at steady state.

6.1.1. Homogeneous Couette flows as a stochastic ODE problem

In this section, we consider a further simplification of (6.1)–(6.3) that leads to a stochastic ODE problem. For this purpose, we note that the velocity field at steady state for $t \to \infty$ has a constant velocity gradient $\dot{\gamma} = \frac{\partial u(y)}{\partial y}$ for all $y \in [0, L]$; cf. Fig. 6.1 (a). Since the stochastic PDE (6.2) only depends on the velocity gradient, the fluid equations (6.1) do not have to be solved in that case. Furthermore, the random field $\boldsymbol{Q}(y, t)$ in (6.2) and in (6.3) then simplifies to a stochastic process $\boldsymbol{Q}(t)$. As a result, we only have to solve the stochastic ODE (6.2) and to perform the integration in (6.3) to compute the stress component τ_{xy} over time. In this simplified setting, we decide for $\beta = 0$ as viscosity ratio since we only compute the elastic stress component and do not consider a Newtonian stress contribution. The time-integration of $\boldsymbol{Q}(t)$ is performed for the Hookean spring force with an Euler-Maruyama method; see Section 2.2.2. The implementation has been programmed in Matlab. The routines provided by Matlab are used to generate the initial equilibrium samples and to perform the interpolation of the solution on a common grid.

In the following, we compare the accuracy and the computational effort for the simplified ODE problem that is discretized on different full grids $V_n^{(\infty)}$ as in Chapter 3, on approximations to the sparse grid space $V_n^{(1)}$ with the classical combination formula (4.31) described in Chapter 4.2 and on approximations with the dimension-adaptive algorithm described in Chapter 5.2; see Algorithm 5.

In Table 6.1 we list a sequence $\boldsymbol{l} = (l_1, l_2)$ of anisotropic full grid spaces $V_{\boldsymbol{l}}$ that is derived from the different dimensions in Fig. 5.3. Similar to the Multilevel Monte Carlo (MLMC) method by Giles [47], we then combine full grid solutions with a different number of timesteps and with different numbers of Brownian configuration fields / sample points. For this purpose, we use the combination formula for shifted index sets (5.30) described in Section 5.2.1. Moreover, the grid sequence listed in Table 6.1 has

Table 6.1.: Sequence of different resolutions (l_1, l_2) for a steady Couette flow.

level $\boldsymbol{l}$	1	2	3	4	5	6	7	8
samples l_1	256	1024	4096	16384	65536	262,144	1,048,576	4,194,304
temporal grid l_2	1/8	1/16	1/32	1/64	1/128	1/256	1/512	1/1024

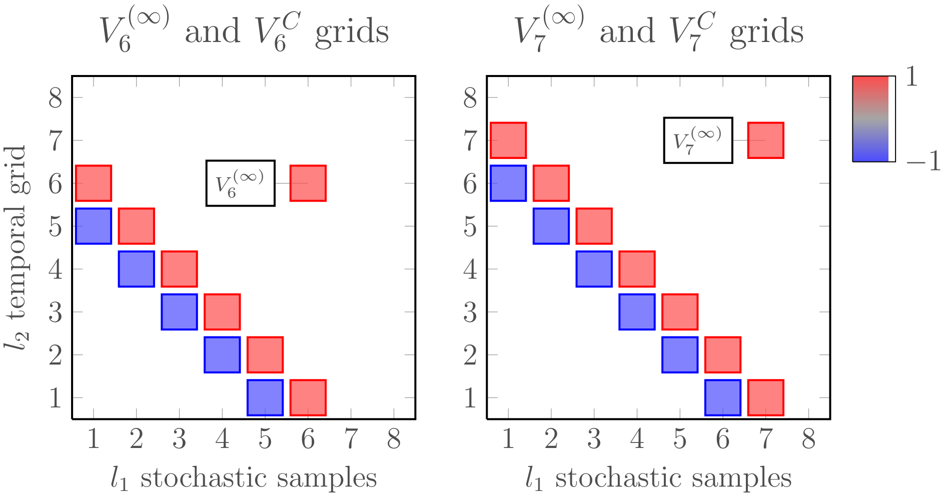


Figure 6.3.: Comparison of the full grid spaces $V_6^{(\infty)}$ and $V_7^{(\infty)}$ with the grids in the classical combination spaces $V_6^{(C)}$ and $V_7^{(C)}$. The resolution of the different grids is listed in Table 6.1.

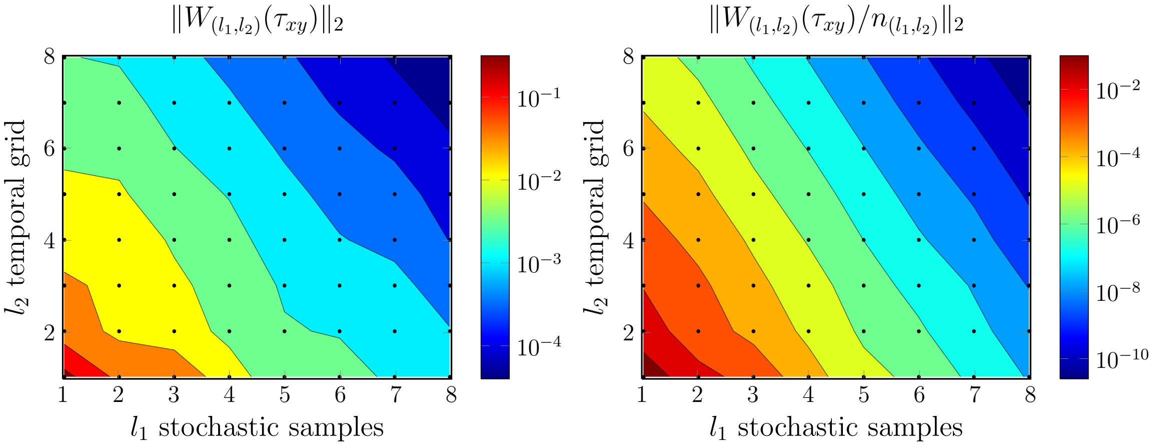
already been refined such that extremely anisotropic full grid solutions are avoided in the classical and in the dimension-adaptive combination technique.

On the left-hand side of Fig. 6.3, the temporal and spatial resolution of the single full grid that defines $V_6^{(\infty)}$ is compared with the resolution of the grids that build $V_6^{(C)}$. Here, grids colored in red have a positive weight +1 and grids colored in blue have a negative weight -1 in the two-dimensional combination formula (4.29). Analogously, the corresponding grids in $V_7^{(\infty)}$ and in $V_7^{(C)}$ are shown on the right-hand side of Fig. 6.3.

The grids in $V_6^{(C)}$ and $V_7^{(C)}$ equally weight both dimensions. This approach is only justified if the refinement factors in Table 6.1 coincide with the corresponding rates of convergence of the numerical schemes. In Chapter 2, we describe the numerical schemes in detail. For the simplified stochastic ODE in (6.2), we employ a first-order Euler-Maruyama scheme in time and evaluate the integral in (6.3) with a Monte Carlo approach. For this reason, we expect a first-order convergence in time and a decay of the sampling error with a rate of $1/2$. Consequently, the rate of grid refinement in Table 6.1 is $\beta_1 = 4$ for the first parameter dimension and $\beta_2 = 2$ for the second dimension; cf. Section 5.1.2.

Potential of dimension-adaptivity In Table 6.1 the factor of refinement was chosen such that the classical combination formula (4.29) is applicable. Next, we apply the dimension-adaptive combination technique. Provided the grid sequence in Table 6.1 was chosen properly, the dimension adaptive refinement should lead to the identical result.

To better comprehend the dimension-adaptive refinement, we first analyze the benefit and the profit



(a) benefit indicator

(b) profit indicator

Figure 6.4.: Comparison of the benefit and the profit indicator for the grid index pair (l_1, l_2) in a steady Couette flow. Note that the rate of refinement is $\beta_1 = 4$ in the first and $\beta_2 = 2$ in the second parameter dimension.

indicator of each index pair (l_1, l_2) in Fig. 6.4. As a benefit indicator for the contribution of V_{l_1,l_2} we employ the size of the hierarchical surplus approximation $\|W_{l_1,l_2}(\tau_{xy})\|_2$ defined as the tensor product of first-order difference operators; see equation (5.6) and (5.7). Up to a normalization constant, this choice coincides with the indicator g_l in (5.14) with $\kappa_1 = 0$ and $\kappa_2 = 1$. As expected, $\|W_{l_1,l_2}(\tau_{xy})\|_2$ decreases with increasing $(l_1, l_2) \in \mathcal{I}$. Apart from some slight oscillations due to the stochastic nature of τ_{xy}, its contour lines in Fig. 6.4 (a) run diagonally. A further explanation for the oscillations could be that the temporal grid with a time-step width of $1/8$ for $l_2 = 1$ is still in the pre-asymptotic regime and a further index shift in that direction might be necessary to obtain the asymptotic first-order rate of convergence; cf. Section 5.2.1. Nevertheless, using the benefit indicator in Fig. 6.4 (a), Algorithm 5 produces an index set which is nearly identical to the index set shown in Fig. 6.3 apart from a slight tendency to perform a further refinement step in the temporal dimension.

Next, we consider the profit indicator in Fig. 6.4 (b). Up to a normalization constant, this choice coincides with the indicator g_l from (5.15) in Section 5.2.1 for $\lambda = 0$. Consequently, $\|W_{l_1,l_2}(\tau_{xy})/n_{l_1,l_2}\|_2$ relates the benefit associated with the hierarchical surplus approximation on the one hand with the cost n_{l_1,l_2} of the approximation space V_{l_1,l_2} on the other hand. Since the cost n_{l_1,l_2} associated with level (l_1, l_2) is of order $O(\text{nr. of samples} \cdot \Delta t_{l_2}^{-1})$, we just employ the product of samples points and the inverse timestep width as an approximation to the cost. This definition can also be applied to the case that the final simulation time is $T_{\text{final}} \neq 1$ since the shape of the contour lines of $\|W_{l_1,l_2}(\tau_{xy})/n_{l_1,l_2}\|_2$ is not affected by this scaling. Theoretically, we could also relate the cost on level (l_1, l_2) to the cost of the coarsest grid $V_{1,1}$ and consider $n_{l_1,l_2}/n_{1,1}$ instead of n_{l_1,l_2}. However, the resulting index set $\mathcal{I}$ is not affected by this scaling.

Fig 6.5 shows the resulting index set $\mathcal{I}^{(p)}$ for the iterations $p = 1, 2, 4, 7, 12, 19$ of Algorithm 5. In agreement with the color coding in Fig. 6.3, grids with a positive weight +1 are colored in red, grids with a negative weight -1 are colored in blue and the remaining gray colored grids are not contained in the combination formula since they have a zero weight contribution. We note that the early iteration

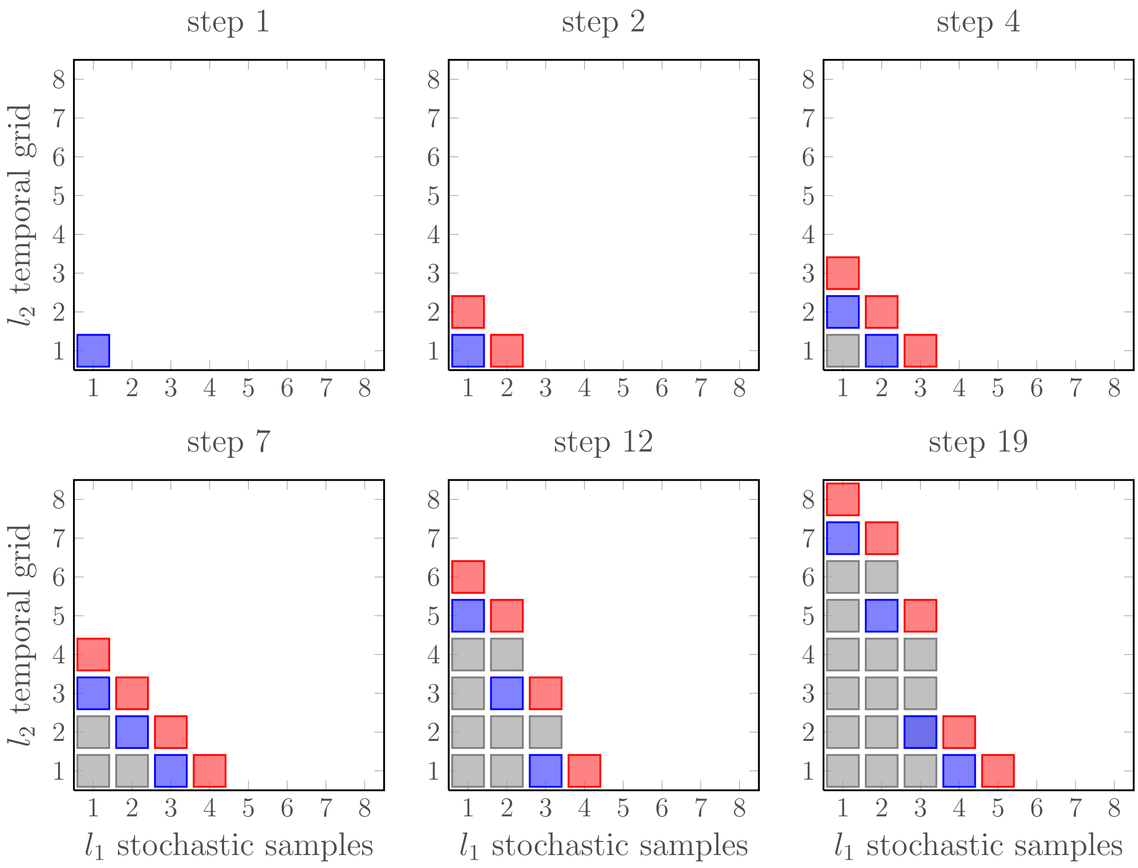


Figure 6.5.: Index set $\mathcal{I}^{(p)}$ for different iteration steps p of the dimension-adaptive algorithm. The color of the grids indicates their weight in the combination formula which is +1 (red), -1 (blue) and 0 (gray).

steps $p = 1, 2, 4, 7$ reproduce the classical combination technique space $V_n^{(C)}$ with $n = 1, 2, 3, 4$. This corresponds with the contour lines of the profit indicator shown in Fig. 6.4 (b). Then, the profit indicator leads to a stronger refinement of the temporal grid. The resulting grid is shown in Fig 6.5 for the iteration steps $p = 12$ and $p = 19$. The primary reason for the stronger refinement of the temporal grid is the anisotropic increase of the cost n_{l_1,l_2} in this direction. This is a consequence of the rate of refinement in the different dimensions in Table 6.1. As a result, a profit indicator has a tendency to include grids with a low computational complexity that would not have been chosen by a refinement with respect to the benefit; cf. Fig. 6.4 (a).

In Table 6.2 we compare the accuracy of all three approaches with respect to the computational effort. For this purpose, a reference solution of τ_{xy} has been computed on a fine temporal grid by solving the corresponding macroscopic constitutive equation (1.8) in the special case $\beta = \epsilon = \zeta = 0$. This parameter choice leads to a simplification of the Oldroyd-B model which is known as upper-convected Maxwell (UCM) model. Furthermore, Table 6.2 specifies the relative costs of the solutions. The relative cost of a full grid solution V_{l_1,l_2} is defined as the ratio $n_{l_1,l_2}/n_{1,1}$. Then, the coarsest grid $V_{1,1}$ has a relative cost of 1.

As a measure for the accuracy of the approaches, Table 6.2 lists the relative L^2-error with respect

Table 6.2.: Relative L^2-error of τ_{xy} with respect to first iteration.

n	cost (full)	$V_n^{(\infty)}$ (full)	cost (sparse)	$V_n^{(C)}$	step p	cost (adapt.)	$V_{\mathcal{I}^{(p)}}$
1	1	1.0	1	1.0	1	1	1.0
2	8	6.7_{-1}	7	7.3_{-1}	2	7	7.3_{-1}
3	64	1.9_{-1}	34	2.2_{-1}	4	34	2.2_{-1}
4	512	7.2_{-2}	148	1.7_{-1}	7	148	1.7_{-1}
5	4,096	3.9_{-2}	616	9.0_{-2}	12	272	1.1_{-1}
6	32,769	2.2_{-2}	2,512	4.7_{-2}	19	1,248	5.3_{-2}
7	262,144	1.2_{-2}	10,144	2.6_{-2}			

to the first iteration, i.e. $\|\tau_{xy,n} - \tau_{xy,\mathrm{ref}}\|_2 / \|\tau_{xy,1} - \tau_{xy,\mathrm{ref}}\|_2$ with $\tau_{xy,n} \in V_n^{(\infty)}$ and analogously for the errors in the sparse grid approximation spaces. We consider relative errors since the absolute errors in $\boldsymbol{u}$ and in $\boldsymbol{\tau}_p$ usually differ in size. Therefore, investigating relative errors allow a better balancing of the different quantities of interest in the following sections. In accordance with the expected numerical rates of convergence, the error of the full grid solutions in $V_n^{(\infty)}$ decreases by a factor of 2 if the level index n is increased by 1; cf. Table 6.1. Next, the central column in Table 6.2 lists the relative error of the classical combination technique. As expected, on the same level index n the solution in $V_n^{(C)}$ has a larger error as the solution in $V_n^{(\infty)}$. On the other hand, the solution in $V_n^{(C)}$ has a much lower complexity compared to $V_n^{(\infty)}$. For the same computational effort, the sparse grid approach achieves a lower relative error as the full grid solution.

The right column in Table 6.2 lists the error of the dimension-adaptive refinement. Since the iteration steps $p = 1, 2, 4, 7$ reproduces the classical combination space the error is the same as in $V_n^{(C)}$. The iteration steps $p = 12$ and $p = 19$ seem to achieve a slight improvement compared to classical combination space. However, the results are not conclusive here. One reason for the lower complexity of $V_{\mathcal{I}^{(p)}}$ for $p = 12$ and $p = 19$ is that the corresponding index sets contain less non-zero full grid solutions than than the combination spaces $V_5^{(C)}$ and $V_6^{(C)}$. We note, however, that the dimension-adaptive approach would perform worse if the full grid solutions with zero weight (gray) would also be considered in the cost calculation. Since we show in Section 6.1.3 that the adaptive index set $\mathcal{I}$ can be used for modified flow conditions this justifies our cost consideration in Table 6.2 where solutions spaces with zero weight are ignored.

6.1.2. General Couette flows as a stochastic PDE problem

In this section we consider the coupled system specified in (6.1)–(6.3). In contrast to Section 6.1.1, the velocity field is now time-dependent and a two-way coupling exists between the velocity and the stress tensor field. Furthermore, the velocity and the stress tensor become time-dependent random fields and the spatial resolution can be varied as a third parameter dimension in the dimension-adaptive algorithm; cf. Fig. 5.3. We apply Algorithm 5 to the velocity field component u since this unknown quantity is of higher practical relevance than the stress tensor field. A further reason to optimize for the velocity field instead of for the stress component τ_{xy} is that the SPDE (6.2) is only moderately influenced by the spatial grid. This is due to the fact that a high spatial resolution only affects the velocity gradient as a coefficient in the SPDE (6.2) for $\boldsymbol{Q}$ or τ_{xy}, respectively. Consequently, a dimension-adaptive procedure in three parameter dimensions for τ_{xy} roughly reproduces a grid sequence as in Fig. 6.4 and therefore effectively

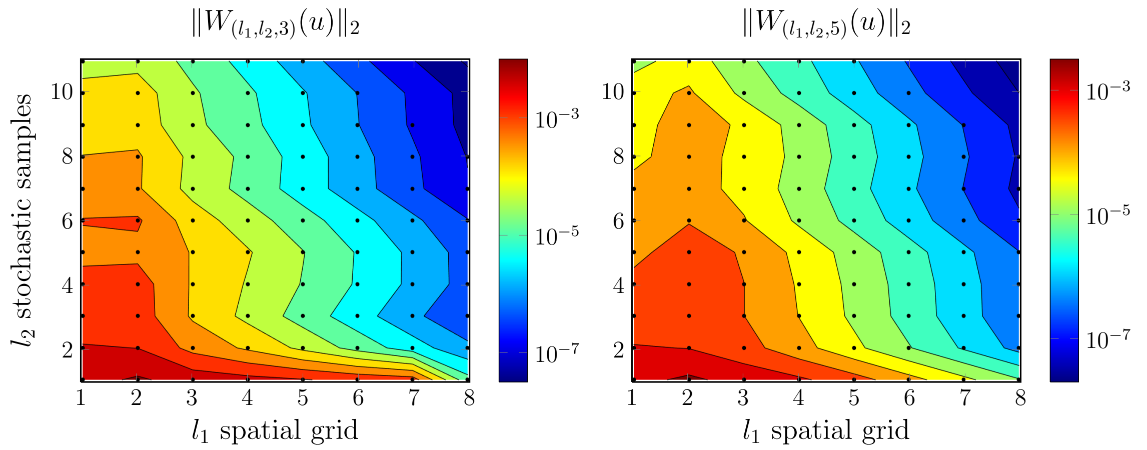
Table 6.3.: Uniformly refined grid sequence (l_1, l_2, l_3) for a time-dependent Couette flow.

level $\boldsymbol{l}$	1	2	3	4	5	6	7	8	...	11
space $1/\Delta x_{l_1}$	2	4	8	16	32	64	128	256		
samples l_2	256	512	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	...	2^{18}
$1/\Delta t_{l_3}$	16	32	64	128	256	512	1024	2048		

reduces to the two parameter dimensions *time* (bias error) and *stochastic realizations* (variance error). The situation differs for the velocity component u. Here, since the absolute error in u is usually one order of magnitude smaller than the absolute error in τ_{xy}, a three-dimensional refinement is beneficial. For a high accuracy in u an exact approximation of $\frac{\partial \tau_{xy}(y,t)}{\partial y}$ in (6.1) is of major importance which requires a fine temporal and stochastic grid to approximate τ_{xy}.

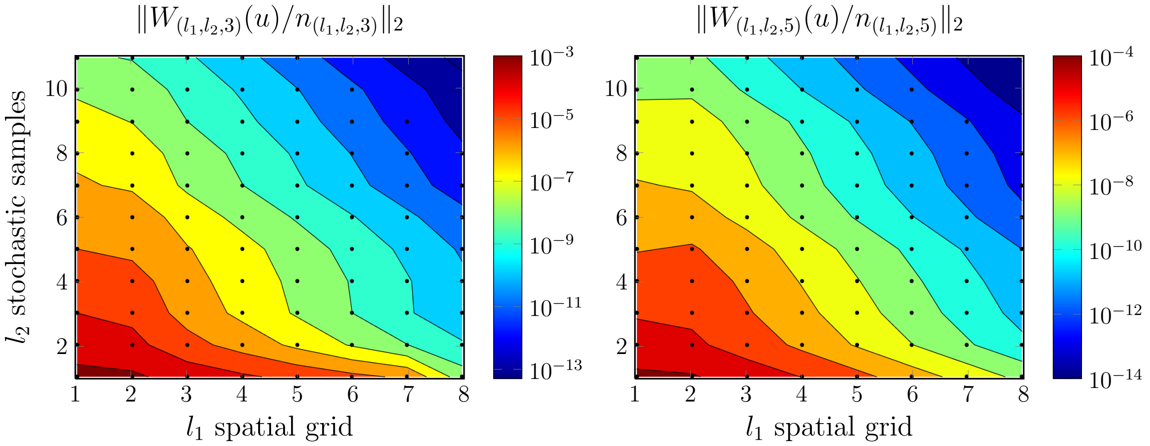
In Table 6.3 we list a sequence of different spatial, temporal and stochastic grid resolutions. For the discretization in space we use central differences with a constant mesh width $\Delta x_{l_1} = 1/l_1$. The central difference stencil simplifies to one-sided differences at the boundary. Moreover, time integration of the macroscopic velocity field is performed with an implicit Euler scheme that is unconditionally stable. Here, the constant timestep width is $\Delta t_{l_3} = 1/l_3$. As a result, the numerical scheme is of first-oder accuracy in space and in time. The corresponding linear system of equations is directly solved with the routines provided by Matlab. Analogously to Section 6.1.1, the SPDE in (6.2) is solved in time with an Euler-Maruyama scheme with constant timestep width $\Delta t_{l_3} = 1/l_3$ for l_2 samples. Then, the integral in (6.3) is evaluated with Monte Carlo quadrature using l_2 sampling points in the center of the finite difference cells as described in Section 2.1. In each dimension, the rate of grid refinement is $\beta_i = 2$ for $i = 1, 2, 3$. Therefore, the cost uniformly increases in each dimension. This differs to Table 6.1 in Section 6.1.1 in which the corresponding order of convergence was considered. Now, we directly use the dimension-adaptive refinement algorithm to decide for the directions and the rates of refinement. The advantage of this approach is that the grid sequence contains more levels and a refinement step only moderately increases the cost of the resulting index set. Consequently, it is easier to avoid a too strong refinement in one problem dimension.

Fig. 6.6 and Fig. 6.7 visualize the benefit and the profit indicator for two-dimensional cuts with $l_3 = 3$ and $l_3 = 5$ in the three-dimensional grid sequence. The benefit indicator $||W_{(l_1,l_2,l_3)}(u)||_2$ for the velocity component is computed according to (5.6) and indicates grid spaces with the largest benefit contribution in the dimension-adaptive procedure. This approach of refinement is a special case of the local refinement indicator $g_{\boldsymbol{l}}$ in (5.14) with $\kappa_1 = 1$ and $\kappa_2 = 0$. In Fig. 6.7, we visualize the ratio of the benefit to the involved work $||W_{(l_1,l_2,l_3)}(u)/n_{l_1,l_2,l_3}||_2$. In the adaptive refinement procedure described in Section 5.2.1, this corresponds to a refinement indicator as in (5.15) for the special choice $\lambda = 1$, i.e. an optimization of the velocity field. A comparison of Fig. 6.6 and Fig. 6.7 reveals some differences in the isoline's shape. While the benefit indicator performs a strong refinement in the stochastic dimension, the profit indicator recommends are more uniform refinement. This is due to the low cost contribution of mixed full grid spaces V_{l_1,l_2,l_3} with $l_1 \approx l_2$ and moderate values for l_1 and l_2. Furthermore, the profit indicator in Fig. 6.7 performs a similar refinement strategy also in the third parameter dimension. Apart from that, we note in Fig. 6.6 some minor oscillations of the isolines for grid spaces V_{l_1,l_2,l_3} with either $l_1 = 1$ or with $l_2 = 1$. A shift of the index set from $\mathcal{I}$ to $\mathcal{I} + (1, 1, 0)$ as described in Section 5.2.1 would remove these oscillations. However, since Algorithm 5 does not show any stability issues for the grid sequence as in Table 6.3, we keep these coarse full grid solutions in our index set $\mathcal{I}$.



(a) benefit indicator $l_3 = 3$ (b) benefit indicator $l_3 = 5$

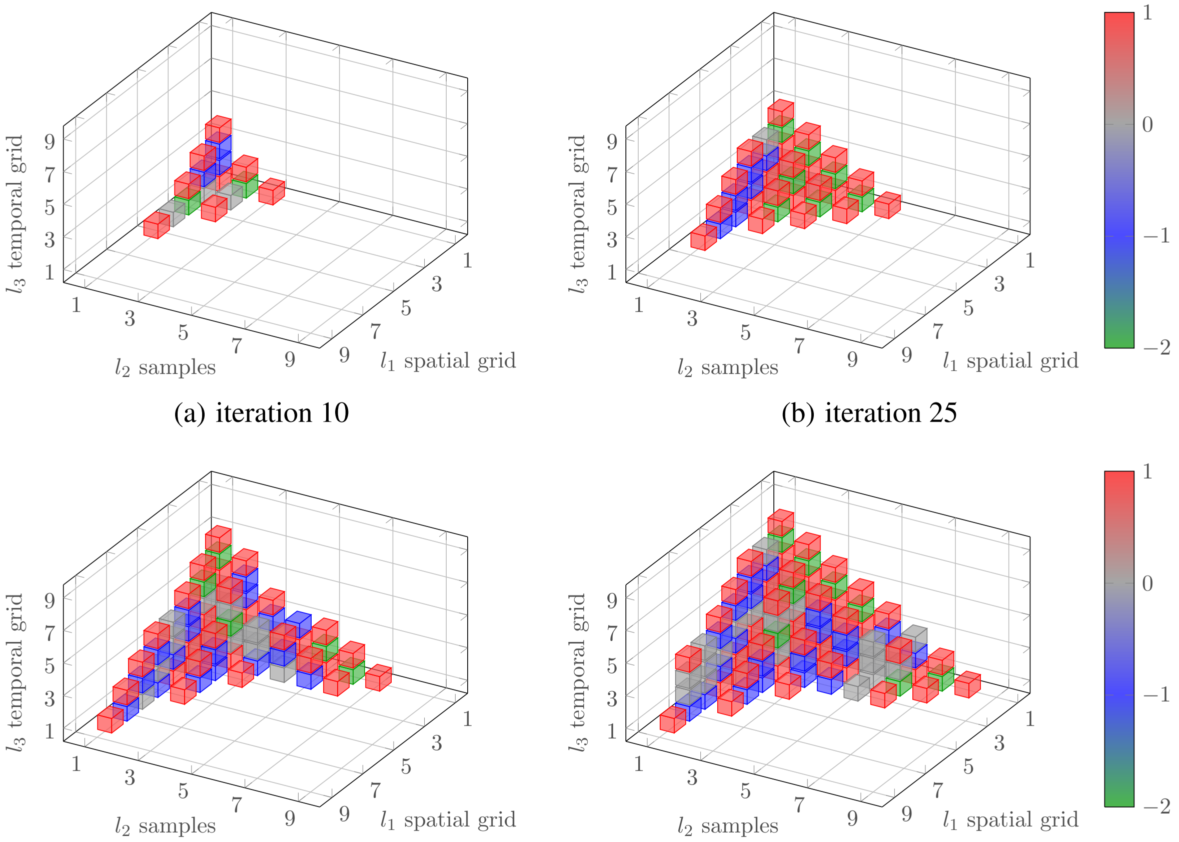
Figure 6.6.: 2D-cut of the benefit indicator with the index pairs $(l_1, l_2, 3)$ and $(l_1, l_2, 5)$ in a time-dependent Couette flow.



(a) profit indicator $l_3 = 3$ (b) profit indicator $l_3 = 5$

Figure 6.7.: Visualization of a 2D cut of the profit indicator with the index pairs $(l_1, l_2, 3)$ and $(l_1, l_2, 5)$ in a time-dependent Couette flow.

Fig. 6.8 visualizes the index sets $\mathcal{I}^{(p)}$ for the iteration steps $p = 10, 25, 55, 90$ of the dimension adaptive algorithm. The color of the grids in Fig. 6.8 represents the weight contribution in the combination formula (5.30). Grids with a positive weight $+1$ are colored red, grids with a negative weight -1 are colored blue and grids that are not required as they have a zero weight contribution are colored gray. Furthermore, the combination technique in 3 dimensions also possesses grids with a weight factor -2. These grids are colored green. The optimization in Fig. 6.8 bases on the size of the profit indicator $\|W_{(l_1,l_2,l_3)}(u)/n_{l_1,l_2,l_3}\|_2$ as shown in Fig. 6.7. The profit indicator leads to a more or less equal refinement in each problem dimension. Later on, we will perform a refinement that solely bases on the benefit



(a) iteration 10

(b) iteration 25

(c) iteration 55

(d) iteration 90

Figure 6.8.: Resulting index set $\mathcal{I}^{(p)}$ for the iteration steps $p = 10, 25, 55, 90$ of the dimension adaptive algorithm. The direction of refinement bases on the profit indicator.

indicator $\|W_{(l_1,l_2,l_3)}(u)\|_2$ and compare the different index sets. It will turn out that the index $\mathcal{I}^{(90)}$ seems to be close to an optimal index set with respect to the cost for a given relative L^2-accuracy of 0.02 for u.

Fig 6.9 shows the solution $u(y, t)$ at two different positions in the channel with height $y = 0.25$ and $y = 0.5$ for the index sets $\mathcal{I}^{(p)}$ with $p = 10, 25, 55, 90$ in comparison with the reference solution. Again, the reference solution was computed with the macroscopic Oldroyd-B model (1.1), (1.2) and (1.4) on a fine spatial and temporal grid. The fluid velocity is largest in the channel center at $y = 0.5$ (top row of Fig. 6.9) and decreases towards the channel walls. At both positions in the flow space, we observe a sequence of velocity over- and undershoots which decrease with ongoing time. The velocity overshoot at $t \approx 2$ is shown in a zoomed extract on the right-hand side of Fig 6.9. As expected, the worst approximation of the velocity overshoot is obtained on the approximation space $V_{\mathcal{I}^{(10)}}$. Interestingly, the coarse approximation space $V_{\mathcal{I}^{(10)}}$ does not only overpredict the magnitude of the velocity overshoot but also shifts its position in time. More precisely, the velocity overshoot in the approximation space $V_{\mathcal{I}^{(10)}}$ appears at $t \approx 0.22$ instead of $t \approx 0.19$. This difference, however, disappears completely with further refinement.

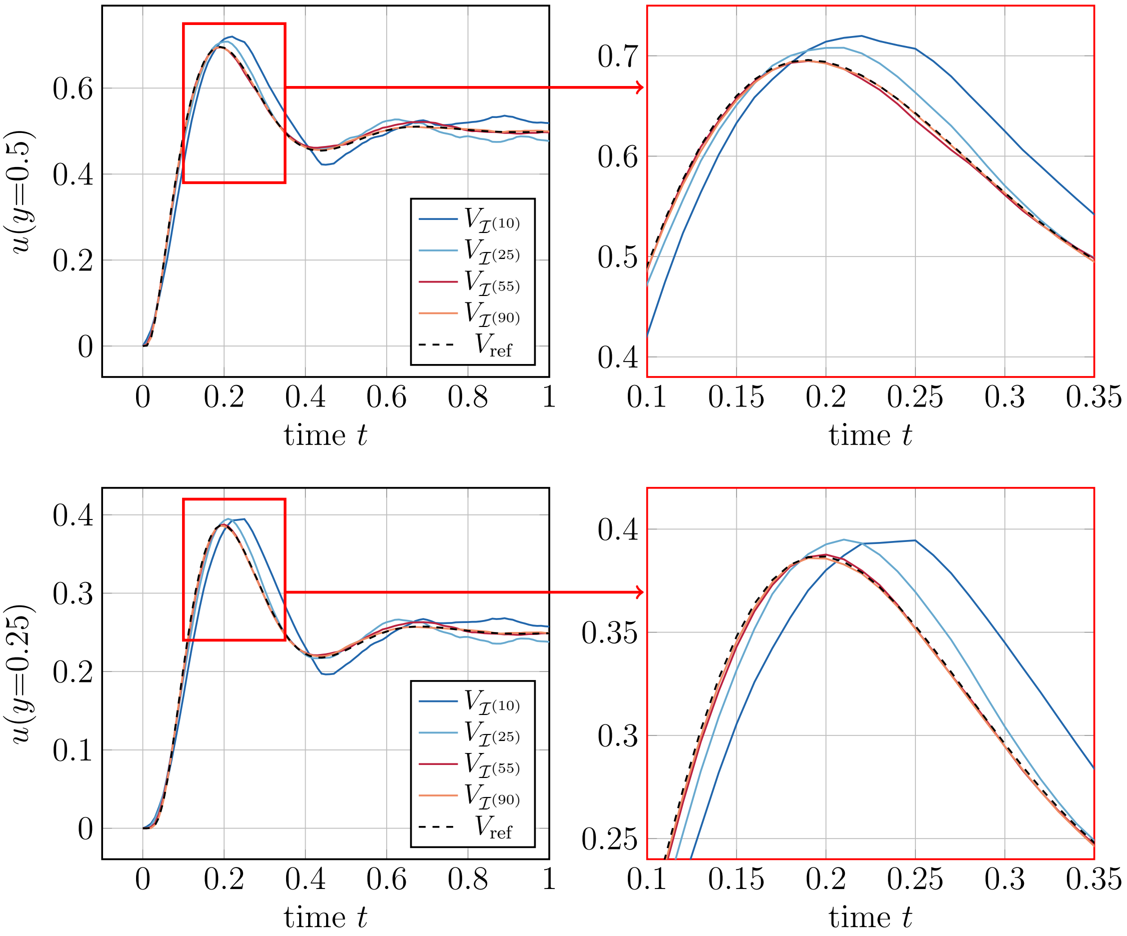


Figure 6.9.: Visualization of the velocity field approximation at two different positions in the flow domain for the spaces $V_{\mathcal{I}^{(p)}}$ and iteration steps $p = 10, 25, 55, 90$ of the dimension-adaptive refinement.

In Fig. 6.10, a quantitative measure of the error decrease with ongoing refinement is given. Here, we measure the relative error $E(u^C) = \frac{\|u^C - u_{\text{ref}}\|_2}{\|u^{(1,1,1)} - u_{\text{ref}}\|_2}$ with respect to the first iteration and with $u^C \in V_{\mathcal{I}^{(p)}}$ for $p = 1, \dots, 90$. As mentioned before, the error does not decrease monotonically. Therefore, there is no guarantee that a further refinement step necessarily decreases the error in u although this is generally the case. Furthermore, the refinement procedure can lead to situations in which $V_{\mathcal{I}^{(p)}} = V_{\mathcal{I}^{(p+1)}}$ for a certain number of refinement steps. This situation appears, for instance, when line 11 in Algorithm 5 returns *false*, i.e. if $\exists l = 1, \dots, d$ with $\boldsymbol{j} - \boldsymbol{e}_l \notin O$. As the index set is not modified in this case, the approximation space is unchanged and therefore the error stagnates. In this application, however, the algorithm achieves optimal results in accordance with the results in Fig 6.9 and in Fig. 6.10.

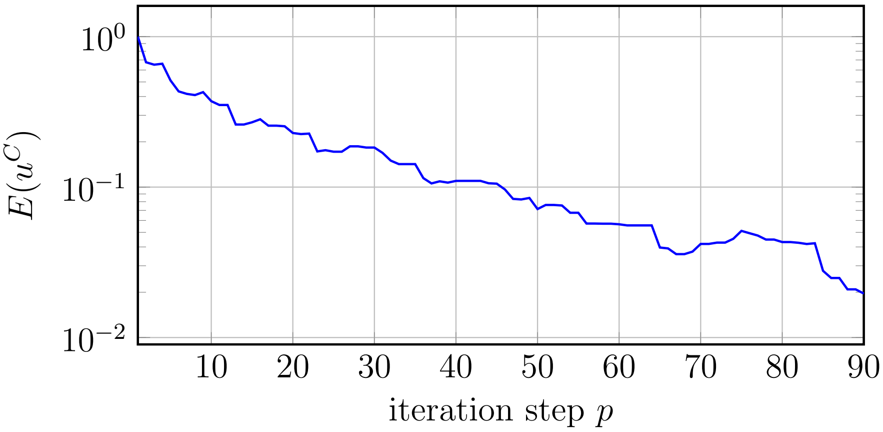


Figure 6.10.: Relative L^2-error of u with respect to $u_{1,1,1} \in V_{1,1,1}$ for each iteration step.

Comparison of profit and benefit grid Next, we compare different refinement criteria for the dimension-adaptive algorithm. Moreover, a special emphasis is on the differences between a refinement with a benefit indicator as shown in Fig. 6.6 and with a profit indicator as shown in Fig. 6.7. For this purpose, Fig. 6.11 compares two index sets that have the same relative error $E(u^C) \approx 0.02$. The index set in the first row of Fig. 6.11 was created in $p = 90$ iteration steps using the profit indicator $||W_{(l_1,l_2,l_3)}(u)/n_{l_1,l_2,l_3}||_2$. This is the index set from Fig. 6.8 (d). On the other hand, the second row of Fig. 6.11 shows an index set which results from a benefit-oriented refinement using $||W_{(l_1,l_2,l_3)}(u)||_2$. Since it is difficult to compare the full three-dimensional index sets, we concentrate on two 2D l_1/l_2-subspaces with $l_3 = 3$ and $l_3 = 5$. These choices correspond with the 2D cuts in Fig. 6.6 and Fig. 6.7. It can be seen that the index sets in Fig. 6.11 are created according to the isolines of the benefit and profit indicator, respectively. While the benefit indicator leads to index sets with very anisotropic grids, i.e. $l_1 \ll l_2$ or vice versa, the profit indicator also contains isotropic grids with $l_1 \approx l_2$. As mentioned before, the isolines of the benefit indicator oscillate for $l_1 = 1$ or $l_2 = 1$. If the index set $\mathcal{I} + (1, 1, 0)$ instead of $\mathcal{I}$ would be considered, a much stronger refinement in the stochastic than in the spatial dimension could be observed. For instance, the index set contains the solution spaces $V_{2,6,3}$ (4-times refined in l_2-dimension compared to $V_{2,2,3}$) and $V_{4,2,3}$ (2-times refined in l_1-dimension compared to $V_{2,2,3}$). This reflects the lower order of convergence of $1/2$ in the stochastic dimension compared to the first-order convergence in the spatial dimension.

It is not directly clear which of the two index sets shown in Fig. 6.11 is better. On the one hand, the profit grid is created with $p = 90$ iterations of Algorithm 5 while the benefit grid requires $p = 105$ iterations. From this perspective, the profit-oriented refinement is the method of choice since the algorithm requires fewer iterations. However, the process of refining the index set $\mathcal{I}$ has to be applied only once since the resulting index set can be reused in comparable flow situations as described in Section 6.1.3.

A further approach to evaluate the quality of the index sets in Fig. 6.11 focuses on the cost of all grids in $\mathcal{I}$ with non-zero weight contribution. Again, we normalize this cost with respect to the cost of the coarsest full grid solution $V_{1,1,1}$ in $\mathcal{I}$, i.e. the cost is defined as $\sum_{(l_1,l_2,l_3)\in\mathcal{I}} n_{l_1,l_2,l_3}/n_{1,1,1}$ for those spaces V_{l_1,l_2,l_3} with a combination coefficient $\neq 0$ in (5.30). Then for given accuracy $E(u^C) = 0.02$, the cost associated with the profit-oriented refinement is 5608. The most expensive full grids solution spaces in

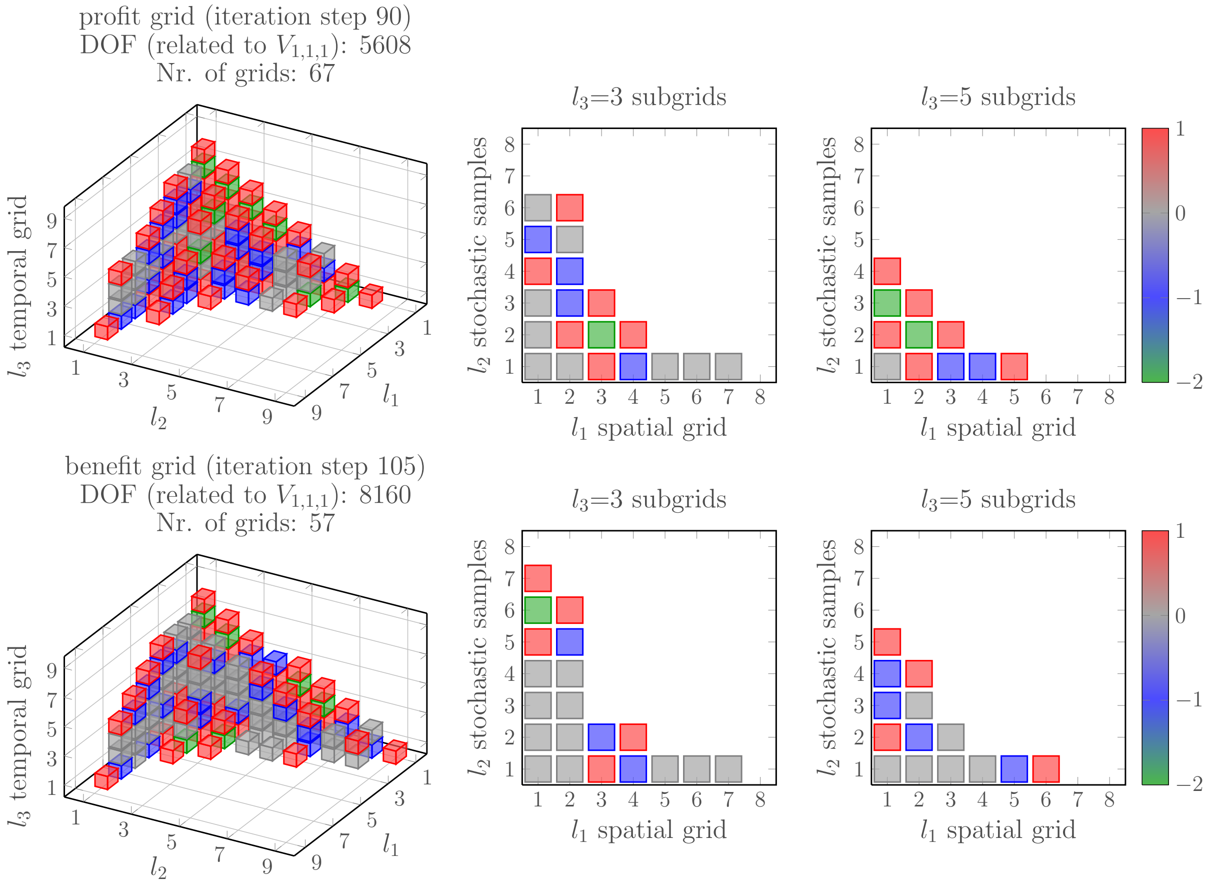


Figure 6.11.: Comparison of an index set created with a profit-oriented refinement (top row) with an index set according to a benefit-oriented refinement (bottom row). Both index sets have the same L^2-error in the velocity component u.

this index set are $V_{7,1,4}$ with relative cost $n_{7,1,4}/n_{1,1,1} = 512 = 2^9$ (9-times refined), $V_{3,1,7}$ with relative cost $n_{3,1,7}/n_{1,1,1} = 256 = 2^8$ (8-times refined) and $V_{2,6,3}$ with relative cost $n_{3,1,7}/n_{1,1,1} = 256$ (8-times refined). Analogously, the cost of a benefit-oriented grid is 8160-times the cost of the discrete space $V_{1,1,1}$ which has one inner grid point, uses 16 steps in time and employs 256 samples in the SPDE. Again, the profit indicator performs better than the benefit indicator.

Moreover, Fig. 6.11 specifies the number of grids with non-zero weight that have been used to compute the relative cost. Here, the situation differs. While the benefit indicator uses 57 grids in the combination formula, the profit indicator requires 67 grids. As mentioned before, these 67 profit grids are still cheaper than the 57 benefit grids due to their relatively coarse refinement. An explanation for this effect is that the benefit indicator always adds the best full grid solutions, ignoring the cost, and therefore requires less grids while the profit indicator gives a better balance of cost and accuracy and therefore minimizes the global cost.

Next, we analyze the influence of the refinement indicators on the resulting index set. For this purpose, we do not only employ a pure benefit indicator ($g_\mathbf{l}$ as in (5.14) with $\kappa_1 = 1$ and $\kappa_2 = 0$) and a pure profit indicator ($g_\mathbf{l}$ as in (5.15) with $\lambda = 1$) but furthermore also consider κ_1-weighted cost to

Table 6.4.: Comparison of the cost of different refinement indicators with the same relative error in the combination formula. For comparison, the last two columns list the cost and the accuracy of the two full grid approximation spaces. For a given accuracy, a profit-oriented optimization requires the lowest cost.

approach	total cost / $n_{1,1,1}$	nr. of grids	rel. error $E(u)$
$\kappa_1 = 0.7$	21,708	154	0.02
$\kappa_1 = 0.85$	7808	83	0.02
$\kappa_1 = 0.9$	7808	83	0.02
$\kappa_1 = 0.925$	8416	85	0.02
$\kappa_1 = 0.95$	8816	69	0.02
$\kappa_1 = 0.975$	8496	68	0.02
benefit grid ($\kappa_1 = 1$)	8160	57	0.02
profit grid ($\lambda = 1$)	5608	67	0.02
$V_{6,6,6}$	32,768	1	≈ 0.04
$V_{7,7,7}$	262,144	1	≈ 0.01

benefit refinement indicators for the velocity field such that $\kappa_2 = 0$. Table 6.4 compares the cost for different choices of κ_1. While $\kappa_1 = 1$ only considers the benefit of a full grid solution, $\kappa_1 = 0$ only takes its computational cost into account. We note that $\kappa_1 = 0$ results in the classical combination formula. In this specific application, $\kappa_1 = 0$ is not an optimal choice since the rate of refinement in Table 6.3 is $\beta_i = 2$ for $i = 1, 2, 3$ independent of the order of convergence in that dimension. Therefore, we cannot expect for a cost-optimized grid to perform best. Indeed, Table 6.4 shows that a large number of 154 full grid solutions is required for $\kappa_1 = 0.7$ to obtain a similar relative L^2-error as the two index sets shown in Fig. 6.11. We do not list smaller values than $\kappa_1 = 0.7$ since then even more full grid solutions would be required. In this application, the interesting range for an optimal choice of κ_1 is the interval [0.7, 1.0]. According to Table 6.4, the cost of the refinement indicator g_l as in (5.14) with $\kappa_2 = 0$ is minimized for $\kappa_1 \in [0.85, 0.9]$. Furthermore, if the number of non-zero grids in the combination formula shall be minimized it is beneficial to use larger values for κ_1 close to 1.0. Interestingly, none of the choices of κ_1 that is listed in Table 6.4 leads to the same index set as the benefit indicator (g_l as in (5.15) with $\lambda = 1$). Consequently, the investigation shows that the resulting index set $\mathcal{I}$ strongly depends on the refinement strategy.

Finally, the last two columns in Table 6.4 lists the cost and the accuracy of the two full grid solution spaces $V_{6,6,6}$ and $V_{7,7,7}$. The accuracy of all dimension-adaptive approaches is between these two full grid spaces but their computational cost is about one order of magnitude smaller. The optimal choice in our setting can be achieved with the profit indicator. Here, the relative cost is 5608 using the optimal index set. The corresponding sparse solution space has a higher accuracy than $V_{6,6,6}$ but requires roughly $1/6$ of its cost. This result clearly shows the large benefit that is possible with a sparse grid discretization. Using an optimal refinement strategy, it is possible to strongly reduce the limitations due to the curse of dimensionality that were observed in Section 2.3.

6.1.3. Effect of modified flow conditions on index sets

The process of creating an optimal index set $\mathcal{I}^{(p)}$ with a dimension-adaptive approach leads to some overhead since some elements in the final index set $\mathcal{I}$ have a zero weight in the combination formula (5.30). But, a positive aspect is that the process of creating $\mathcal{I}$ only has to be applied once for a specific class of problems. The index set can be reused for modified flow conditions. This property is important, for instance, in the context of *Uncertainty Quantification* where the same type of problem is considered various times with slightly modified flow conditions. This section considers for a fixed index set the error increase in the case of modified flow conditions.

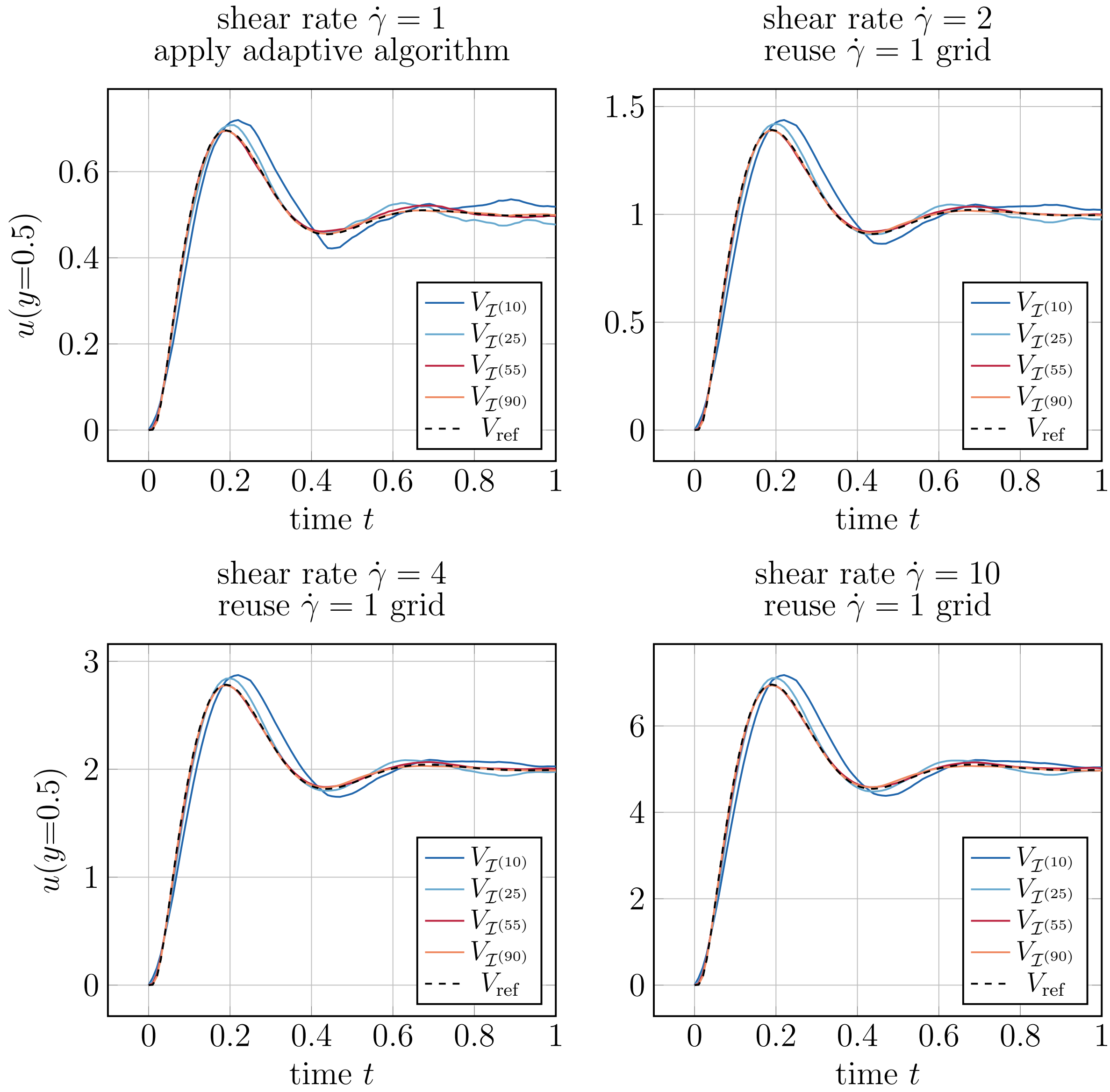


Figure 6.12.: Comparison of the velocity field over time for different shear rates $\dot{\gamma} = 1, 2, 4, 10$. For $\dot{\gamma} > 1$ we reuse the index set from the profit-oriented refinement as shown in Fig. 6.8.

As a first modification of the flow conditions, we vary the shear rate $\dot{\gamma}$ in the range [1, 10]. In every new simulation, we reuse the index sets $\mathcal{I}^{(p)}$ from the profit-oriented refinement strategy for $\dot{\gamma} = 1$ since this approach performed best in the previous section; see Table 6.4. Next, Fig. 6.12 shows the numerical solution in the channel center at $y = 0.5$ over time for the original shear rate $\dot{\gamma} = 1.0$ as in Fig. 6.9 and for the shear rates $\dot{\gamma} = 2.0$, $\dot{\gamma} = 4.0$ and $\dot{\gamma} = 10.0$. As mentioned before, we always employ the same index set $\mathcal{I}^{(p)}$ for the iteration steps $p = 10, 25, 55, 90$ which are visualized in Fig. 6.8. Of course, we have to compute the corresponding full grid solutions for the new shear rates. In this case, however, the advantage is that we only have to compute the solutions with a non-zero weight in the combination formula, for instance, 67 solutions in $\mathcal{I}^{(90)}$ with $\text{card}(\mathcal{I}^{(90)}) = 111$. Since all of these grids are comparative cheap, the numerical result can be obtained much faster than on a fine full grid.

As can be seen in Fig. 6.12 the fluid velocity linearly scales with the shear rate $\dot{\gamma}$. Apart from that, the temporal evolution of u is the same for all flow rates. In this case, the index set $\mathcal{I}^{(p)}$ can be reused for modified shear rates $\dot{\gamma} \neq 1$. For a better analysis of this assumption, we list the relative and absolute L^2-error (in space and in time) of u in Table 6.5. Interestingly, the relative error is almost identical independent of the shear rate. This is due to the linear scaling of the flow rate with $\dot{\gamma}$ which apart from the scaling does not affect the velocity field profile. This coincides with the results in Fig. 6.12 since the deviations, for instance, of the solution in $\mathcal{I}^{(10)}$ to the reference solution are very similar for all shear rates. Furthermore, Table 6.5 specifies the absolute error of the velocity field. As expected, this error scales linearly with $\dot{\gamma}$ analogously to the flow rate increase.

A further modification of the flow conditions can be applied to the Deborah number De which represents the amount of elasticity in the fluid. A modification of De is of practical relevance since the Deborah number affects the flow profile and, therefore, is more difficult to approximate with a reused index set. For this reason, Fig. 6.13 shows the velocity profile for $De = 0.5$ as in the previous section and for the three modifications $De = 1.0, 2.0, 4.0$. The comparison shows that the velocity over- and undershoots for the flow with $De = 0.5$ dampen out if De is increased and only one velocity overshoot

Table 6.5.: Comparison of absolute and relative L^2-errors on dimension-adaptive sparse grids (obtained for $\dot{\gamma} = 1$) for flows with different shear rates.

	$\dot{\gamma} = 1$	$\dot{\gamma} = 2$	$\dot{\gamma} = 4$	$\dot{\gamma} = 10$
relative error	$\|\|u^{C,p} - u_{\text{ref}}\|\|_2 / \|\|u^{C,1} - u_{\text{ref}}\|\|_2$			
iteration $p = 1$	1	1	1	1
iteration $p = 10$	0.373	0.366	0.363	0.362
iteration $p = 25$	0.172	0.165	0.163	0.162
iteration $p = 55$	0.068	0.065	0.064	0.064
iteration $p = 90$	0.02	0.019	0.02	0.02
absolute error	$\|\|u^{C,p} - u_{\text{ref}}\|\|_2$			
iteration $p = 1$	1.07_{-1}	2.16_{-1}	4.33_{-1}	1.09_{0}
iteration $p = 10$	3.98_{-2}	7.88_{-2}	1.57_{-1}	3.93_{-1}
iteration $p = 25$	1.83_{-2}	3.55_{-2}	7.05_{-2}	1.76_{-1}
iteration $p = 55$	7.21_{-3}	1.39_{-2}	2.76_{-2}	6.90_{-2}
iteration $p = 90$	2.09_{-3}	4.17_{-3}	8.59_{-3}	2.20_{-2}

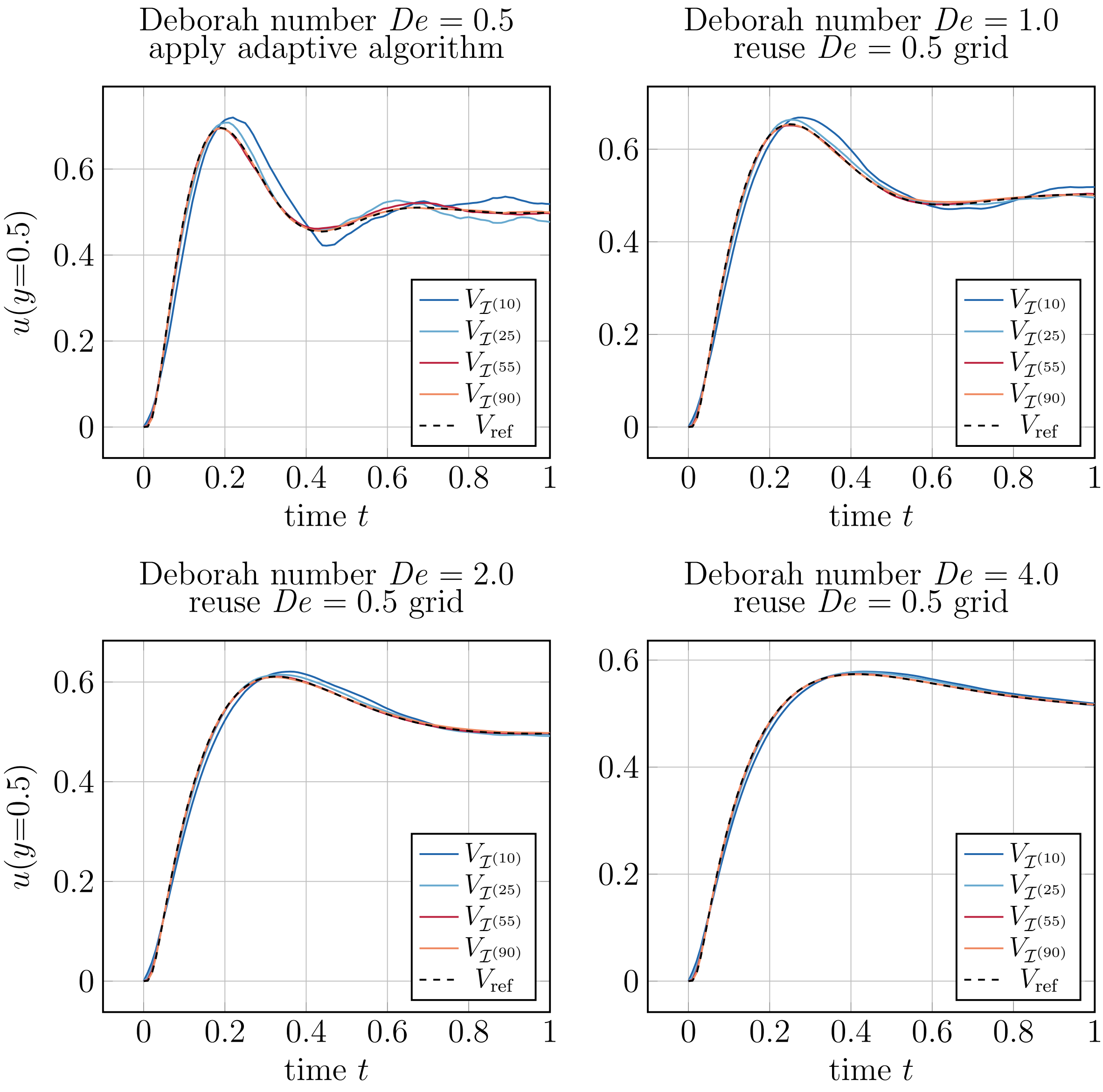


Figure 6.13.: Couette flow simulation with four different Deborah numbers. All four simulations employ the index set $\mathcal{I}$ that was adaptively built for $De = 0.5$. The solution u is visualized at position $y = 0.5$ for the dimension-adaptive iterations steps $p = 10, 25, 55, 90$.

remains. From a numerical point of view, it is easier to approximate a single large overshoot. In contrast to our expectations, the numerical approximation on $V_{\mathcal{I}(p)}$ even becomes better with increasing De.

The finding that the numerical solution in $V_{\mathcal{I}(p)}$ becomes better for larger Deborah numbers, corresponds with the results in Table 6.6 that state the quantitative error. Again, we note that the relative error is only slightly affected by the Deborah number. This is due to the fact that the relative error is measured with respect to the error in $V_{1,1,1}$. Since the absolute error in $V_{1,1,1}$ also becomes smaller for larger Deborah numbers (cf. bottom rows in Table 6.6), the relative error (top rows) is calculated with

Table 6.6.: Absolute and relative L^2-error for flows with modified Deborah numbers. The dimension-adaptive approach always uses the index set $\mathcal{I}^{(p)}$ that has been obtained for $De = 0.5$.

	$De = 0.5$	$De = 1.0$	$De = 2.0$	$De = 4.0$
relative error	$\|\|u^{C,p} - u_{\text{ref}}\|\|_2 / \|\|u^{C,1} - u_{\text{ref}}\|\|_2$			
iteration $p = 1$	1	1	1	1
iteration $p = 10$	0.373	0.344	0.349	0.379
iteration $p = 25$	0.172	0.162	0.174	0.195
iteration $p = 55$	0.068	0.063	0.067	0.075
iteration $p = 90$	0.02	0.023	0.018	0.012
absolute error	$\|\|u^{C,p} - u_{\text{ref}}\|\|_2$			
iteration $p = 1$	1.07_{-1}	9.73_{-2}	8.60_{-2}	7.49_{-2}
iteration $p = 10$	3.98_{-2}	3.35_{-2}	3.01_{-2}	2.84_{-2}
iteration $p = 25$	1.83_{-2}	1.58_{-2}	1.50_{-2}	1.46_{-2}
iteration $p = 55$	7.21_{-3}	6.08_{-3}	5.77_{-3}	5.62_{-3}
iteration $p = 90$	2.09_{-3}	2.23_{-3}	1.52_{-3}	9.25_{-4}

respect to a more accurate initial solution. Nevertheless, the smaller absolute error corresponds with the qualitative results in Fig. 6.13.

Finally, we note that further variations of the flow conditions are possible, e.g. a variation of the channel height. Even for these modifications, we expect that our index sets achieves accurate results. This is due to the fact that the flow field is laminar and continuously depends on the flow parameters. Therefore, as long as the parameters are chosen within this laminar regime, the index set as in Fig. 6.8 is optimal with respect to the profit / cost-benefit ratio for a given set of discretization schemes and a given refinement of the parameter dimensions; cf. Table 6.3. The restriction on laminar flow fields is not a severe limitation in our application since all complex simulations in Chapter 3 are performed with a Reynolds number below 1 in a laminar flow regime. This is due to the fact that the interesting flow phenomena for non-Newtonian fluids result from the fluid's elasticity and not from inertial forces. As a result, the 3D dimension-adaptive approach as shown in this section is actually applicable to a wide range of complex non-Newtonian flow phenomena. In the next section, we will consider a different example which includes the modeling accuracy as a further dimension for optimization.

6.2. Extensional flows

In the following section, we apply the dimension-adaptive combination technique to homogeneous extensional flows. Here, we will not only combine full grid spaces with anisotropic temporal and stochastic solution as for the shear flow simulations in Section 6.1 but will also combine full grid spaces with different modeling resolutions; cf. Fig. 5.3 for an illustration of the different dimensions for optimization. Therefore, the following dimension-adaptive approach is more elaborated as in Section 6.1 due to the combination of anisotropic numerical and modeling grids.

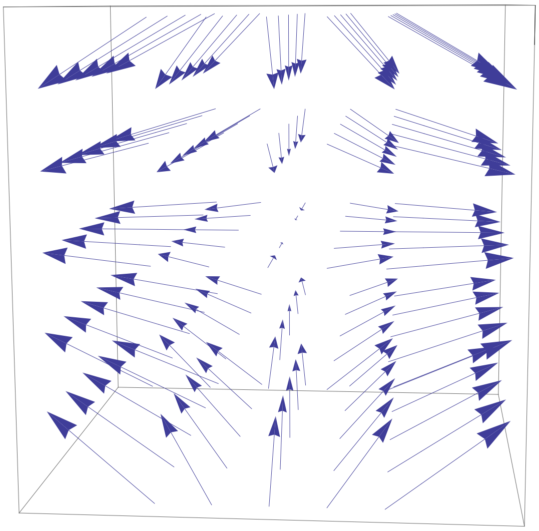


Figure 6.14.: Visualization of a three-dimensional uniaxial extensional flow.

Description of the flow

Homogeneous extensional flows have already been described in Section 3.1. Therefore, we only summarize the important characteristics of 3D homogeneous extensional flows that will be considered in the following and refer to Section 3.1 for a more general description of this flow type.

The velocity field in a homogeneous three-dimensional extensional flow is defined as

$$\boldsymbol{u} \;=\; (\dot{\epsilon}x, -\frac{\dot{\epsilon}}{2}y, -\frac{\dot{\epsilon}}{2}z) \tag{6.5}$$

in which the dimensionless scalar elongation rate $\dot{\epsilon} > 0$ characterizes the strength of the flow. The velocity field is visualized in Fig. 6.14. Since the velocity gradient that results from (6.5) does not depend on space and time, the coupled system in (1.59)–(1.63) reduces to

$$\mathrm{d}\boldsymbol{Q}(t) = \Big[\nabla\boldsymbol{u}^T\boldsymbol{Q}(t) - \frac{1}{4\,De(N)}\boldsymbol{A}\cdot\boldsymbol{F}\left(\boldsymbol{Q}(t)\right)\Big]\mathrm{d}t + \sigma\mathrm{d}\boldsymbol{W}(t) \tag{6.6}$$

$$\boldsymbol{\tau_p}(t) = \frac{3(1-\beta)\,(b(N)+5)}{b(N)\,De(N)\,((N+1)^2-1)}\sum_{i=1}^{N}\left(\mathbb{E}[\boldsymbol{Q}_i(t)\otimes\boldsymbol{F}_i(\boldsymbol{Q}_i(t))]-\mathbf{Id}\right). \tag{6.7}$$

Consequently, we solve (6.6) and (6.7) for the $3N$-valued stochastic process $\boldsymbol{Q}(t) = (\boldsymbol{Q}_1(t),\ldots,\boldsymbol{Q}_N(t))$ and for the stress tensor $\boldsymbol{\tau_p}(t)$ but finally concentrate on the stress tensor as the primary quantity of interest. In an extensional flow, all shear components of $\boldsymbol{\tau_p}(t)$ are zero and we have $\boldsymbol{\tau_p}(t) = \mathrm{diag}(\tau_{xx}(t), \tau_{yy}(t), \tau_{zz}(t))$.

The common parameters of all following simulations are listed in Table 6.7. Most of these parameters coincide with the simulation parameters in Table 3.2 for the full grid spring-chain simulations in Section 3.1.2. Nevertheless, in this section we balance the number of stochastic realizations, the number of time steps and the number of spring segments for the different full grid approximations. In contrast to the temporal scheme for the Hooke model in Section 6.1, we now employ the semi-implicit

Euler-Maruyama scheme from (2.22) for the temporal evolution of the stochastic realizations. This scheme is implicit in the FENE spring force contribution in (6.6) but explicit in the remaining terms.

3D extensional flow with spring-chain models		
Deborah number	De	1.0
viscosity ratio	β	0.0
extensional rate	$\dot{\epsilon}$	2.0
spring model	FENE	
maximum spring extension	b	120.0
spring extension gauge	$b(N)$	de Gennes (1.48)
spring relaxation gauge	$\lambda(N)$	rel. time scale (1.51)

Table 6.7.: Common simulation parameters for all full grid solutions in the index set $\mathcal{I}$.

Adaptivity with anisotropic numerical and modeling grids

As a first motivation for the following approach, we consider Fig. 6.15 which again visualizes the results from Fig. 3.3 in Section 3.1.2. Fig. 6.15 displays the temporal evolution of the extensional viscosity $\eta(\dot{\epsilon}) = (\tau_{xx} - \tau_{yy})/\dot{\epsilon}$ for different spring-chain models with $N = 1, 2, \ldots, 5$ segments on a full grid space with a high resolution in the temporal and the stochastic dimension. We first observe that the dumbbell model $N = 1$ strongly differs from the multi-segment systems. But, the larger the number of spring segments N becomes, the more reduces the difference in the predicted extensional viscosity. Moreover, the temporal evolution for $N = 4$ and $N = 5$ can only be distinguished on a very fine grid. Usually, an index set $\mathcal{I}$ in the dimension-adaptive approach contains coarse full grid solutions with relatively large bias and variance errors. For these coarse full grid solutions, the corresponding discretization errors can be larger than the difference between, for instance, the spring model with $N = 4$ or with $N = 5$ chains. Consequently, it might be beneficial to employ coarse spring chain approximations with a certain modeling error to reduce the computational cost in situations with numerical errors of comparable size. In the following, we will perform an exact analysis of this idea.

First, we adaptively built an index set $\mathcal{I}$ from a set of full grid solutions that are listed in Table 6.8. We note that this grid sequence is strongly shifted in the stochastic dimensions l_1; see Section 5.2.1. Consequently, we employ a relatively fine initial grid in the stochastic dimension to increase the stability of the numerical approach.

In the end, we aim for a cost effective approximation of a five-segment spring-chain system which is indicated in Fig. 6.15 with a dashed black line. For this reason, we artificially restrict the refinement in the modeling dimensions to $l_3 = 5$ spring segments; cf. Table 6.8. This restriction corresponds with the maximum resolution $\boldsymbol{l}_{\max}$ in line 3 of Algorithm 5. The restriction is necessary since the hierarchical surplus for grids with $l_3 = 5$ is usually not zero so that the algorithm might decide to further refine the modeling dimension. Of course, a restriction such as $l_3 \leq 5$ is not necessary if a solution for the asymptotic limit $N \to \infty$ is searched for. However, we do know an analytic formula for the limit $N \to \infty$. Consequently, it is not possible to evaluate the quality of a solution with an unlimited refinement in dimension l_3.

Analogously to the previous section, the refinement is performed according to the profit indicator $||W_{(1,l_2,l_3)}(\tau_{xx})/n_{l_1,l_2,l_3}||_2$ (cf. refinement indicator (5.15) with $\lambda = 0$) for the dominant stress tensor com-

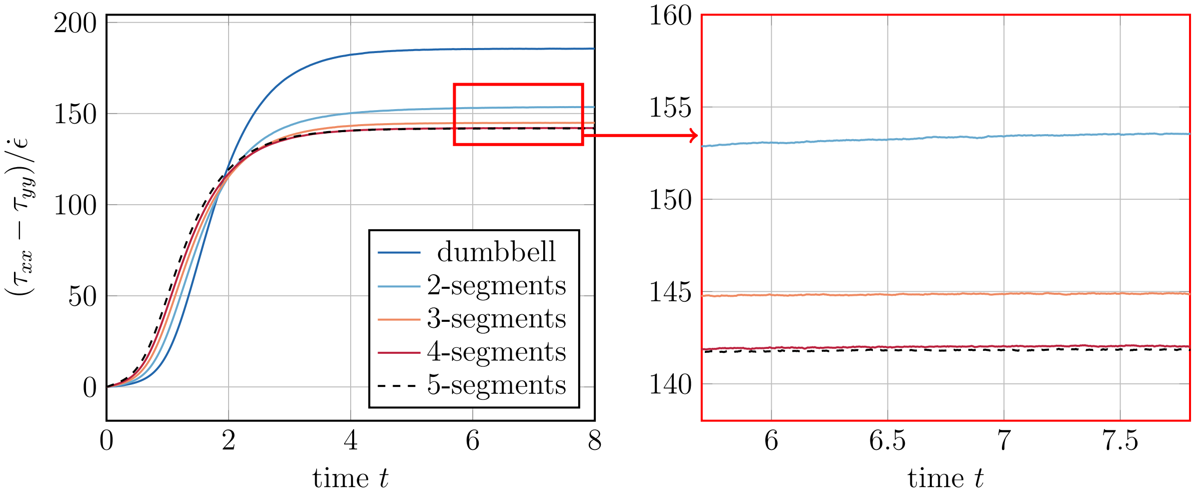


Figure 6.15.: Illustration of the basic idea for using different modeling accuracies. The figure displays the extensional viscosity for different spring-chain systems of length N. Note that the figure coincides with Fig. 3.3 in Section 3.1.2.

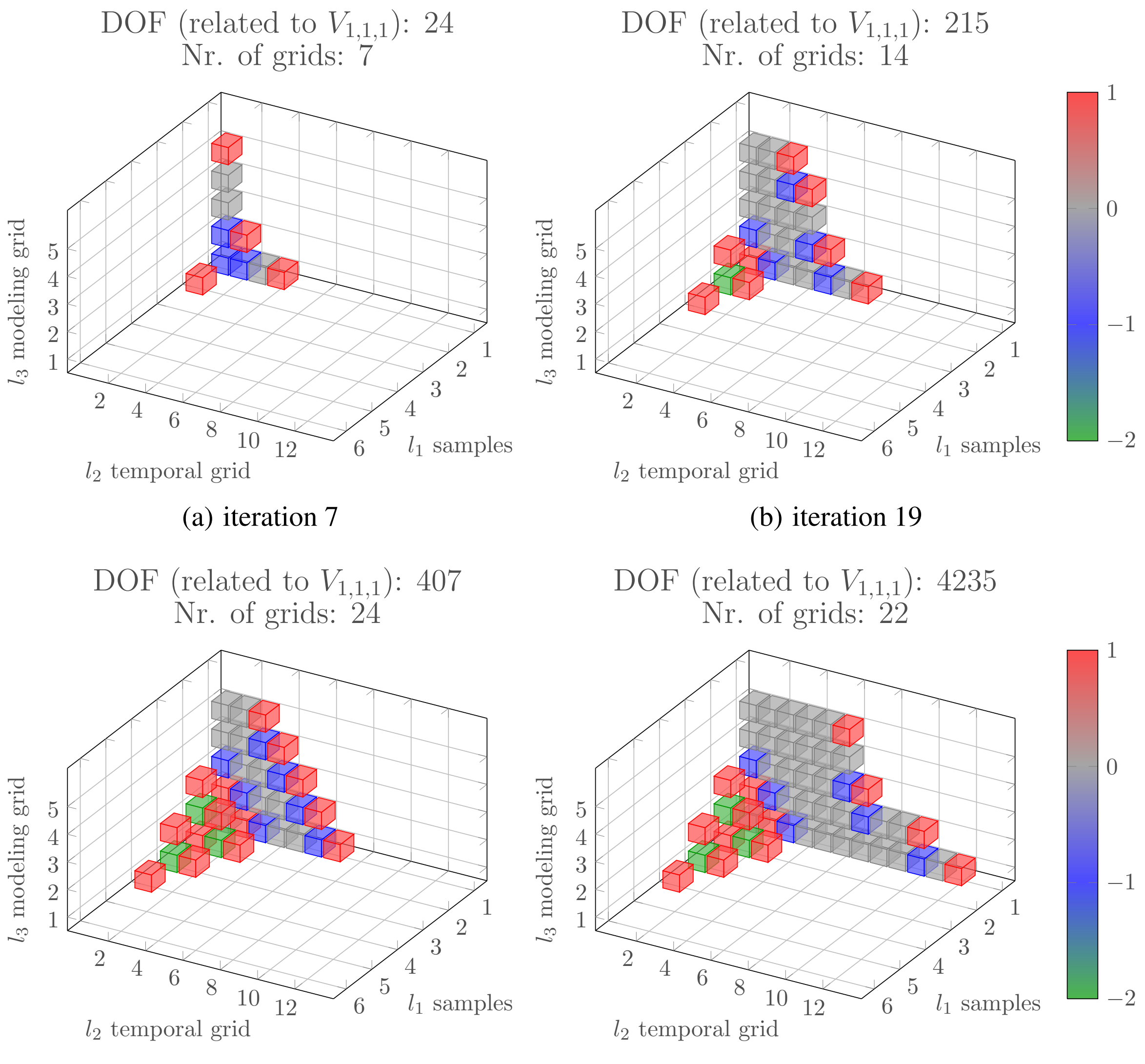
Table 6.8.: Grid sequence of the dimension-adaptive refinement of a homogeneous extensional flow.

level $\boldsymbol{l}$	1	2	3	4	5	6	7	...	12
samples l_1	1024	2048	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}	...	$2 \cdot 10^6$
$1/\Delta t_{l_2}$	2	4	8	16	32	64	128	...	4096
springs l_3	1	2	3	4	5				

ponent τ_{xx} of $\boldsymbol{\tau_p}$. Since the other normal stress components of $\boldsymbol{\tau_p}$ are close to zero, this choice of error measurement corresponds with the tensor maximum norm (5.21) and $||\cdot||_2$ as in (5.23) in Section 5.2.1.

Next, Fig. 6.16 shows the resulting index set $\mathcal{I}^{(p)}$ for the iterations $p = 7, 19, 24, 38$ of Algorithm 5. In accordance with Section 6.1.2, we measure the relative error of τ_{xx} with respect to the first iteration in each iteration step p as $E(\tau_{xx}^C) = \frac{||\tau_{xx}^C - \tau_{xx,\text{ref}}||_2}{||\tau_{xx}^{(1,1,1)} - \tau_{xx,\text{ref}}||_2}$. Here, the reference solution $\tau_{xx,\text{ref}}$ is the full grid solution in $V_{12,12,5}$, i.e. a fine numerical discretization of a five-segment spring chain system. As mentioned before, due to this choice of the reference solution a refinement with $l_3 > 5$ has to be excluded in Algorithm 5.

The adaptive refinement shown in Fig. 6.16 terminates if the error tolerance reaches $E(\tau_{xx}^C) < 0.02$. In total, 38 iterations are required for this threshold as shown in Fig. 6.16 (d). The relative error for each iteration step is shown on the left hand side of Fig. 6.17. We note that the error does not decrease monotonically. A further refinement can always lead to completely different weights in the combination formula. Furthermore, an iteration step does not necessarily lead to new grids, i.e. $\mathcal{I}^{(p)} = \mathcal{I}^{(p+1)}$, if the if-loop in line 14 of Algorithm 5 returns *false*; cf. Fig. 5.2. This effect can be seen in the iteration steps $p = 20$ and $p = 21$. Here, the error stagnates as the number of elements in $\mathcal{I}$ does not change. Therefore, the computational effort also stagnates. This is shown on the right-hand side of Fig. 6.17. Here, we do not state the cost of all grids in $\mathcal{I}$ but only take the cost of full grid spaces with non-zero



(a) iteration 7

(b) iteration 19

(c) iteration 24

(d) iteration 38

Figure 6.16.: Visualization of the index set $\mathcal{I}$ in a homogeneous extensional flow. The color of the grids specifies the cost in the combination formula. Consequently, grids colored in gray are not required in the end.

weight into account. Generally, if the cardinality of $\mathcal{I}$ increases, the number of non-zero grids grows. But, as can be seen in Fig. 6.16 (c) with 24 grids (iteration $p = 24$) and in Fig. 6.16 (d) with 22 (iteration $p = 38$) non-zero grids, the opposite situation is possible as well. Nevertheless, we generally observe a decrease of the relative error with ongoing refinement. Furthermore, the increase in the computational cost in Fig. 6.17 is moderate in comparison to a factor of $n_{l_1+1,l_2+1,l_3+1}/n_{l_1,l_2,l_3} = 4(l_3+1)/l_3$ if a full grid solution space is uniformly refined in each dimension according to $V_{l_1,l_2,l_3} \to V_{l_1+1,l_2+1,l_3+1}$.

The adaptive refinement as shown in Fig. 6.16 primarily takes place in the modeling and in the temporal dimension. This is due to the fact that the initial full grid solution is relatively fine in the

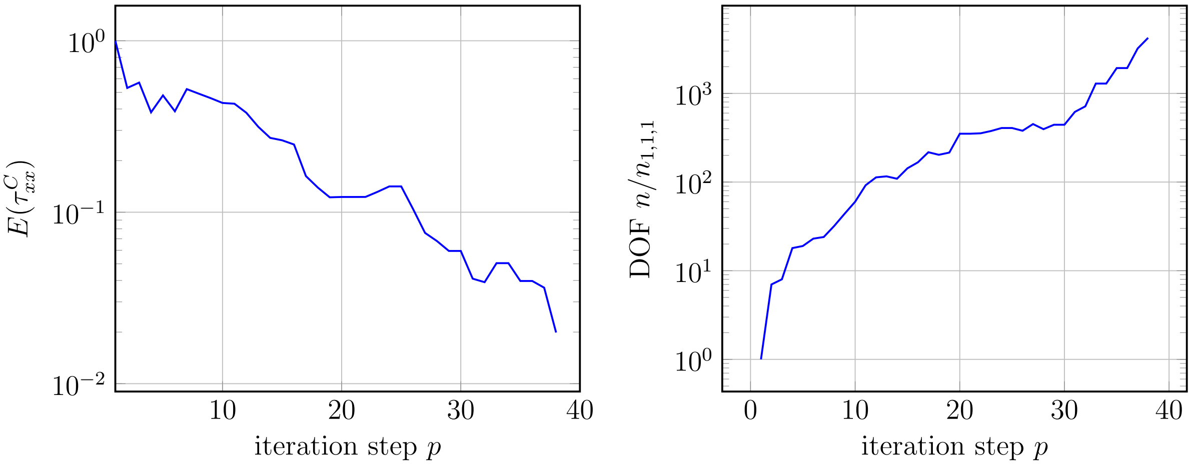


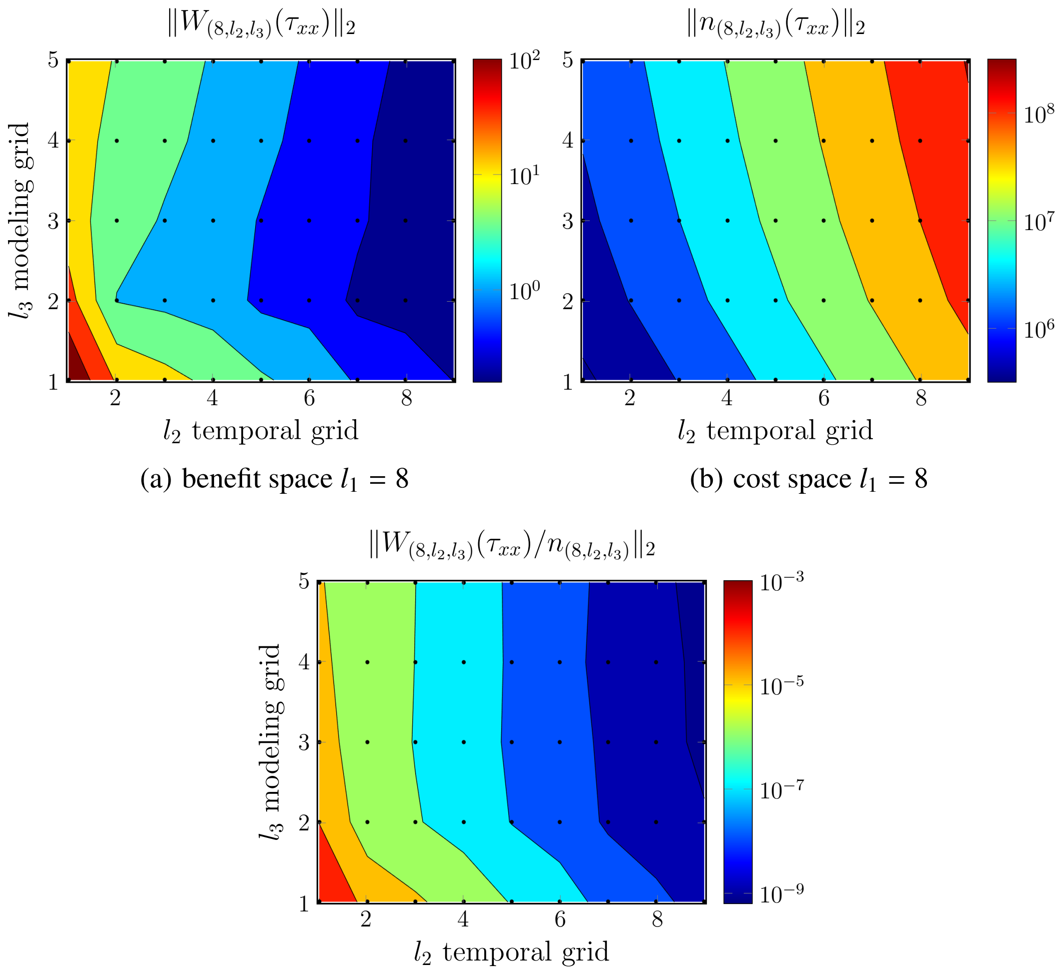
Figure 6.17.: Plot of the relative error in τ_{xx} and the cost of the corresponding non-zero full grid solutions in the combination formula for different refinement steps p.

stochastic dimension but very coarse in the temporal dimension; cf. Table 6.8. More precisely, the stochastic grid on level $l_1 = 1$ with $2^{10} = 1024$ samples has already been refined 10-times compared to the coarsest possible stochastic grid with $2^0 = 1$ samples. This initial refinement can also be interpreted as an index shift $\mathcal{I} + (10, 0, 0)$ as explained in detail in Section 5.2.1. We note that this initial refinement in the stochastic dimension was required to increase the numerical stability of the algorithm. In the end, the adaptive algorithm recognizes that only a few further refinement steps in the stochastic dimension are necessary. The dominant error terms are the bias and the modeling error.

The refinement in the temporal and in the modeling dimension also has a certain structure that will be discussed in the following; cf. Fig. 6.16. As shown in Fig. 6.16 (a), the algorithm tends to fully refine the modeling dimension up to level $l_3 = 5$ for fixed l_1 and l_2. In the end, this complete refinement leads to a full grid in the modeling dimension. An explanation for this refinement behavior is given in Fig. 6.18. Here, we show a 2D cut of the hierarchical surplus (benefit space), the computational costs and the profit index for the time and modeling dimension. This cut in Fig. 6.18 visualizes a fine stochastic grid with $l_1 = 8$. This is due to the fact that the hierarchical surplus for τ_{xx} as defined in (5.6) uses first-order differences in every dimension. If the first dimension is fully refined as for $l_1 = 8$, the first-order difference in the stochastic dimension is comparatively small and Fig. 6.18 actually shows a reduced difference stencil in the 2nd / 3rd dimension.

Fig. 6.18 (a) displays the isolines of the benefit space for a 2D-cut in the temporal-modeling dimension. For $l_3 = 1$ or $l_3 = 2$, the isolines run diagonally as expected for an isotropic error decrease in both dimensions. This underlines that the dynamic behavior of a dumbbell model with $l_3 = N = 1$ spring segments strongly differs from a multi-spring chain with $l_3 > 1$. Then, from $l_3 = 2$ to $l_3 = 5$ the isolines run parallel to the l_3-axis. Interestingly, a similar shape of the isolines is visible in Fig. 6.18 (b) for the cost space. This is a consequence of the anisotropic increase of the cost in dimension 2 and 3; see Section 5.2.1. While the cost increases exponentially in the temporal dimension l_2 since $\beta_2 = 2$, the cost increases linearly in the number of springs segments N. As a result, the contour lines in Fig. 6.18 (c) also run parallel to the l_3-axis. Therefore, the algorithm fully resolves the modeling dimension since this dimension can be inexpensively refined and the refinement leads to a large error reduction.

As shown in Fig. 6.18 (a), the hierarchical surplus does not seem to decrease in the modeling dimen-



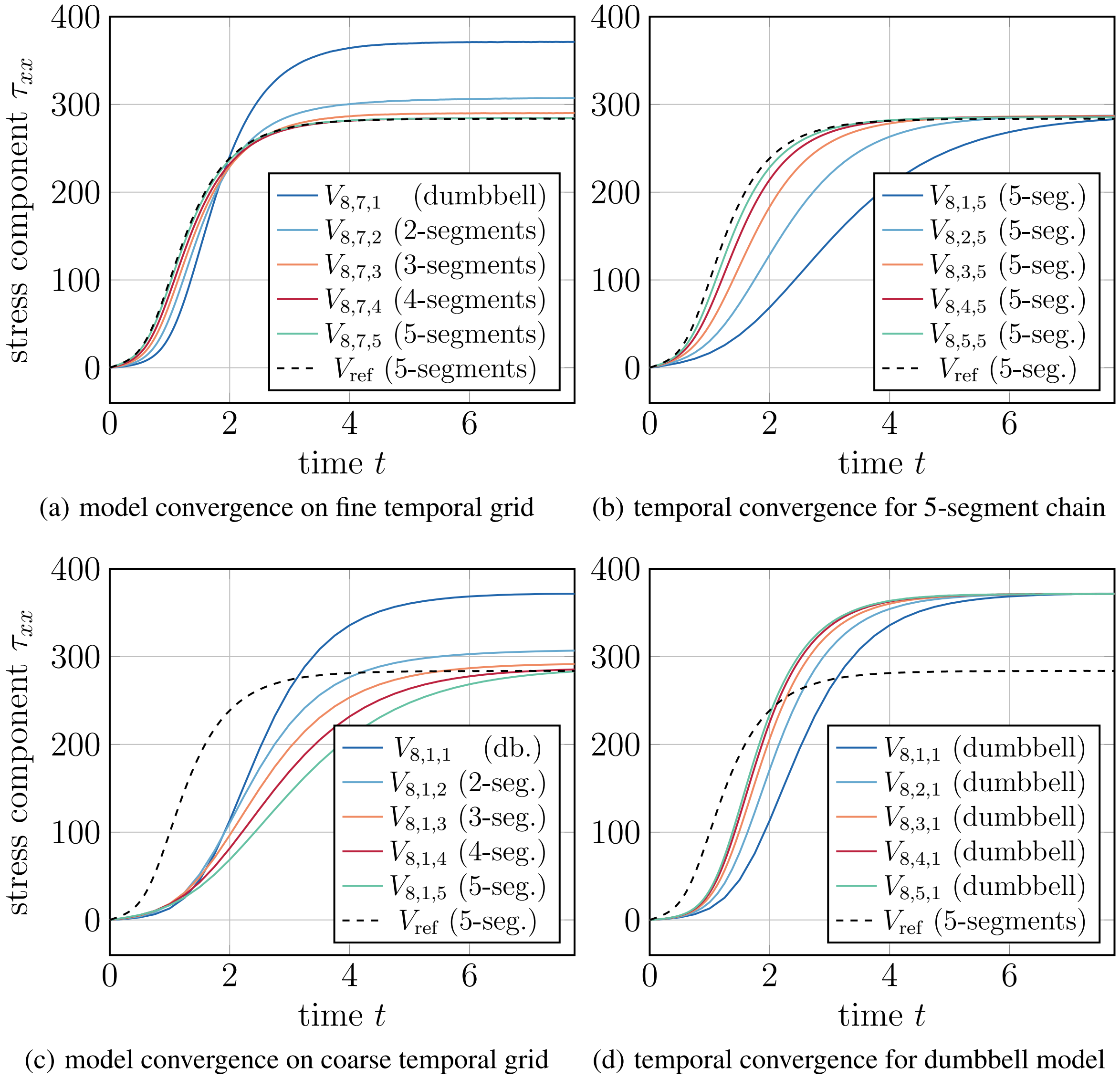
(a) benefit space $l_1 = 8$

(b) cost space $l_1 = 8$

(c) profit space $l_1 = 8$

Figure 6.18.: Visualization of the benefit space (hierarchical surplus approximation), the cost space and the profit space of τ_{xx} in a 2D-cut.

sion which is, beside the anisotropic cost, a further reason for a full refinement of that dimension. This finding seems to disagree with our initial examples in Fig. 6.15 where the four and the five-segment spring-chain predict similar stress tensor values. An explanation of this apparent contradiction is given in Fig. 6.19. All four subfigures Fig. 6.19 (a)–(d) visualize τ_{xx} on a sequence of differently refined full grid spaces. The refinements on the left-hand side in Fig. 6.19 (a) and Fig. 6.19 (c) are performed with respect to the modeling grid in both cases but differ in their temporal error. On the right-hand side in Fig. 6.19 (b) and Fig. 6.19 (d), the grids are refined in the temporal dimension and the figures show the convergence for $\Delta t_{l_2} \to 0$. We note that Fig. 6.19 (a) closely resembles Fig. 6.16. Since $\tau_{yy} \approx 0$ and $\dot{\epsilon} = 2$, the normal stress component τ_{xx} is roughly two times the size of the extensional viscosity. In Fig. 6.19 (a) the solution spaces $V_{8,7,4}$ (4-segment chain) and $V_{8,7,5}$ (5-segment chain) closely resemble each other. Interestingly, the situation differs for Fig. 6.19 (c). Here, the evolution of τ_{xx} on $V_{8,1,4}$ (4-segment chain with coarse temporal resolution) and on $V_{8,1,5}$ (5-segment chain with coarse temporal



(a) model convergence on fine temporal grid

(b) temporal convergence for 5-segment chain

(c) model convergence on coarse temporal grid

(d) temporal convergence for dumbbell model

Figure 6.19.: Comparison of the convergence behavior of different refined full grid solution. The partial figure (c) gives an explanation for a high surplus on certain grids in the modeling dimension. This leads to a full refinement of that dimension; see Fig. 6.18.

resolution) still differ strongly. This deviation leads to a high hierarchical surplus. Consequently, the algorithm directly refines this dimension; cf. Fig. 6.16. As expected, we do not observe these large differences for the temporal grids in Fig. 6.19 (b) and Fig. 6.19 (d). Here, the solutions on $V_{8,4,5}$ and $V_{8,5,5}$ and, moreover, on $V_{8,4,1}$ and $V_{8,5,1}$ are in high agreement with each other.

In the end, Fig. 6.19 (c) shows that the spring relaxation gauge from de Gennes (1.48) is not able to perfectly match the dynamic behavior of the spring-chain systems on all grids in $\mathcal{I}$. However, the steady state stress components in Fig. 6.19 (c) for $t > 6.0$ are in high agreement. This result is expected since it is much easier to equilibrate the steady state of a system instead of the full dynamics. Nevertheless,

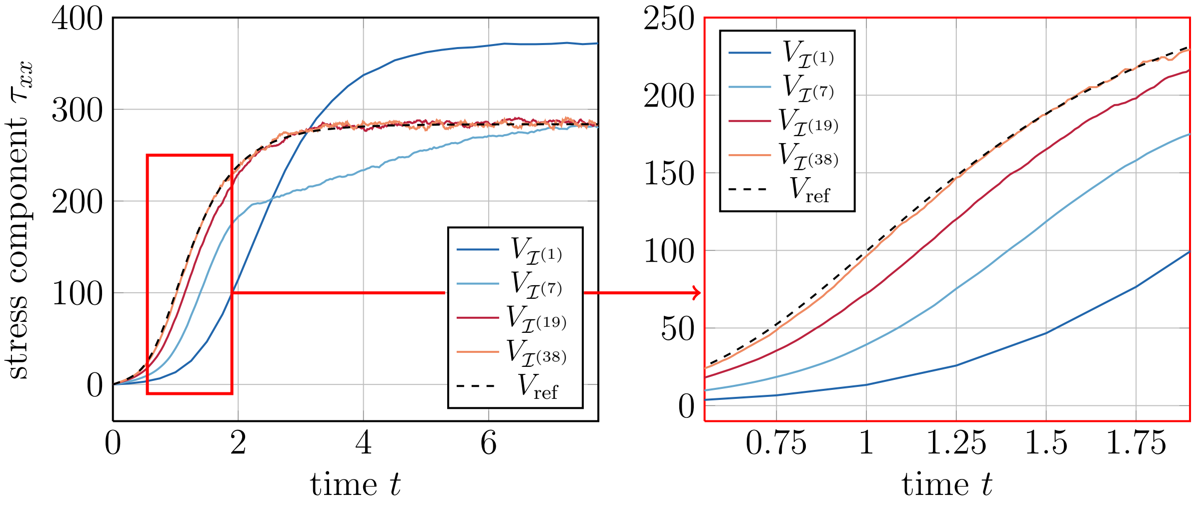


Figure 6.20.: Plot of the τ_{xx} stress component over time for a reference solution on $V_{12,12,5}$ and for the iteration steps $p = 1, 7, 19, 38$ of the dimension-adaptive algorithm.

our dimension-adaptive algorithm still terminates successfully. The only consequence of the different spring-chain dynamic behavior in Fig. 6.19 (c) is that the modeling dimension is fully refined which, in the end, reduces the efficiency of Algorithm 5.

The convergence of the algorithm is illustrated in Fig. 6.20. For this purpose, the temporal evolution for the iteration steps $p = 1, 7, 19, 38$ is compared with a numerical reference solution in $V_{12,12,5}$. Since the index set in the first iteration step only consists of the coarsest full grid solution such that $\mathcal{I}^{(1)} = \{(1, 1, 1)\}$, the corresponding solution strongly differs from the reference solution. This is due to the fact that the model equation in $V_{1,1,1}$ actually describes the dynamics of a dumbbell system and not the dynamics of a five-segment chain. But, with ongoing iteration of Algorithm 5 the numerical solution better approximates the reference solution. At iteration step $p = 38$ the combined solution correctly describes the dynamics of the five-segment chain. Furthermore, we show the dynamic behavior of the different numerical solutions in more detail in a zoomed extract on the right-hand side of Fig. 6.20. Here, we notice that the solution in $V_{\mathcal{I}^{(19)}}$ already gives an accurate approximation of the steady state stress tensor value but not of the early stage for $t \leq 2.0$. Interestingly, the approximation space $V_{\mathcal{I}^{(38)}}$ correctly describes the evolution of τ_{xx} in the complete time range, even though $\mathcal{I}^{(38)}$ still contains grids that approximate a dumbbell system. It might be that some of these coarse full grid solutions have to be removed from the index set if a higher accuracy as $E(\tau_{xx}) = 0.02$ is searched for. Again, this can be achieved by shifting the index set, see Section 5.2.1, and remove these grids from the sequence in Table 6.8.

Finally, Table 6.9 compares the relative cost and the relative accuracy of several sparse solution spaces with comparable full grid solution spaces with respect to the L^2-norm. We note that the final index set $\mathcal{I}^{(38)}$ has an accuracy which is similar to the accuracy of the full grid solution space $V_{8,6,5}$ but for 1/5 of its cost; cf. Table 6.9. The most expensive full grid spaces in $\mathcal{I}^{(38)}$ are $V_{1,12,1}$ with relative cost $n_{1,12,1}/n_{1,1,1} = 2048$ (11-times refined), $V_{1,10,2}$ with relative cost $n_{1,10,2}/n_{1,1,1} = 1024$ (10-times refined) and $V_{1,10,1}$ with relative cost $n_{1,10,1}/n_{1,1,1} = 512$ (9-times refined); cf. Fig. 6.16 (d). Interestingly, spaces with $l_3 = 5$ which are fully refined in the modeling dimension are not the most cost expensive spaces in $\mathcal{I}^{(38)}$. As mentioned before, this is due to the anisotropic cost increase in l_3 compared to l_1 and l_2.

Table 6.9.: Comparison of the relative cost and the relative accuracy of different sparse and full solution spaces with respect to $V_{1,1,1}$.

approach	total cost / $n_{1,1,1}$	nr. of grids	rel. error $E(\tau_{xx})$
adaptive step 7	24	7	0.52
adaptive step 19	215	14	0.12
adaptive step 38	4,235	22	0.02
$V_{8,4,5}$	5,120	1	0.098
$V_{8,5,5}$	10,240	1	0.048
$V_{8,6,5}$	20,480	1	0.021
$V_{8,7,5}$	40,960	1	0.009

As a result of Table 6.9, we conclude that the dimension-adaptive combination technique is able to not only cope with grids that have an anisotropic numerical resolution but also to cope with different modeling accuracies. In this application on extensional flows, the approach was able to reduce the cost by a factor of five with respect to a single full grid solution. This is less than for an unsteady Couette flow described in Section 6.1.2 where the cost was reduced by a factor of about ten. This slightly reduced efficiency for the anisotropic numerical-modeling grid is due to a fully refinement of the modeling grid, see Fig. 6.16, and the low cost increase in dimension l_3. Nevertheless, our results clearly show the potential of a dimension-adaptive refinement in which the modeling equation is considered as a further problem dimension. We assume that this idea is generally applicable for other problems of interest. This invokes, however, a detailed understanding of the modeling equation and a sensible gauge theory for the corresponding units; see, e.g., Section 1.2.3 for a comparison of different spring models and relaxation time gauges.

Conclusion

Summary

In this thesis, we implemented and investigated a multiscale spring-chain model for the simulation of three-dimensional dilute polymeric fluids. A multiscale approach was chosen due to the fundamental modeling errors that often occur for classical macroscopic approaches even in simple flow fields such as the breakdown of the macroscopic Oldroyd-B model in homogeneous extensional flows. The major advantages of multiscale approaches as used in this thesis are

- a higher modeling accuracy since closure errors can be avoided,
- a better understanding of the microscopic polymer configurations since this quantity is directly modeled, and
- a higher numerical stability which, to some point, overcomes the *high Weissenberg number problem*; cf. Fattal and Kupferman [36, 37].

We described in this thesis the first implementation of a high-dimensional spring-chain model into a three-dimensional flow solver. Furthermore, we presented in Section 3.2.2 new multiscale simulation results for a 4:1 square-square contraction flow in a three-dimensional channel. Since there are no other simulation results on this problem so far, we compared our findings with experimental measurements from the literature. Besides, we verified our implementation on simple flow problems with previously published numerical results. This included homogeneous flows in Section 3.1 and two-dimensional contraction flows in Section 3.2.1. Note that a further benchmark for our multiscale model can be found in Griebel and Rüttgers [51] where we consider a viscoelastic unsteady Poiseuille flow for which an analytic solution exists.

Our multiscale simulation approaches delivered valuable insights into the underlying structure of the polymeric molecules. On the downside, the large number of degrees of freedom led to high-dimensional, time-dependent random fields. As an illustrative example, the Fokker-Planck equation that describes the temporal evolution of an N-segment spring-chain system in a three-dimensional flow space is $3N + 4$-dimensional (1D in time, 3D in flow space and $3N$-dimensional in configuration space for the polymer orientations). This enormous computational complexity explains the limited number of results available in the literature.

One approach to overcome the *curse of dimensionality* based on massively parallel computing and was discussed in Section 2.3.2 of this thesis. The approach allowed us to compute the multiscale simulation results in Chapter 3. For this purpose, the parallel CPU cluster *Atacama* with a total CPU number of 1248 processors was used. The cluster is operated by the Institute for Numerical Simulation and the Sonderforschungsbereich 1060 at the University of Bonn. Even then, some of the simulations required several weeks of computing time.

A more advanced approach to reduce the computational complexity described in Chapter 4 based on *sparse grid* discretizations. The main idea of this method is to achieve a comparable or only slightly reduced accuracy (by a logarithmic factor) on a grid with much lower complexity. For our specific application, it was advantageous to use the *combination technique*. This sparse grid variant approximated the solution in a sparse grid space by a superposition of solutions in several coarse full grid spaces. The reasons for using the combination technique were that

- we could reuse our multiscale model implementation for the full grid solver *NaSt3DPGF*, see first part of this thesis, and that
- the different full grid solutions could be computed independently since the combination technique is intrinsically parallel.

On a parallel cluster such as *Atacama*, the computational cost was therefore primarily defined by the cost of the most expensive full grid solution which could also be computed in parallel. This further verified that the combination technique is perfectly adapted for exascale computing as considered, for instance, in the project "An Exa-Scalable Two-Level Sparse Grid Approach for Higher-Dimensional Problems in Plasma Physics and Beyond" (EXAHD) of SPP 1648; see Pflüger et al. [104]. Furthermore, this thesis described the first application of the combination technique to multiscale viscoelastic flow problems.

As mentioned before, we related the macroscopic fluid equations with the microscopic polymer orientations in the multiscale approach. The accuracy of this combined system thus depended on a complex coupling between macroscopic and microscopic discretization errors. Therefore, it was crucial for an efficient use of the combination technique to identify the quantities of interest and the important problem dimensions. In our application, these problem dimensions were

- the *spatial resolution* of the velocity, pressure and stress tensor field in a three-dimensional flow space,
- the *temporal resolution* of the coupled system of equations with an isotropic temporal resolution of microscopic and macroscopic equations, and
- the *stochastic resolution*, i.e. the number of discrete realizations (Brownian configuration fields) of the vector-valued random field that described the polymer orientations in the stochastic approach.

In Section 6.1, we combined grids with an anisotropic refinement of these problem dimensions to simulate Couette flow problems. In a first step in Section 6.1.1, a simplified homogeneous Couette flow was considered. Here, only the temporal and the stochastic resolution of the corresponding stochastic ODE was varied similar to the Multilevel Monte Carlo approach. Then in Section 6.1.2, we used the approach to simulate general Couette flows in which macroscale and microscale were directly coupled. Here, we also combined anisotropic spatial grids. In the end, the dimension-adaptive combination technique reduced the computation complexity by one order of magnitude compared to full grid solutions with a comparable accuracy. This underlines the enormous cost reduction that was possible by using the combination technique.

Furthermore, we demonstrated in Section 6.1.3 that the cost for creating the index set $\mathcal{I}$, i.e. the dimension-adaptive refinement process itself, was negligible if compared to the cost of all full grid solutions with non-zero weight in $\mathcal{I}$. This was due to the fact that, at least for laminar flows which typically occur in our application, the index set $\mathcal{I}$ could be reused for modified flow conditions. This becomes especially important in applications that are related to *Uncertainty Quantification* (UQ). In a non-intrusive UQ approach, certain quantities of interest are defined and computed with a large sequence of simulation runs. Usually, every simulation run is performed with modified flow conditions. The index set used in Section 6.1.3 is perfectly suited to perform a non-intrusive UQ analysis of an unsteady Couette flow with the numerical schemes from Chapter 2. Furthermore, note that our multiscale model already contained an UQ-related type of uncertainty due to the stochastic nature of Brownian motion.

In Section 6.2 we further extended the concept of problem dimensions which were refined in the dimension-adaptive algorithm to *modeling dimensions*. This was a novel extension for the combination technique. The basic idea resulted from the observation that different FENE spring-chains became more and more alike for an increasing number of spring segments. This required, however, a careful gauge fixing of the different spring parameters such as maximum extension or spring stiffness. In the literature is, as far as we know, no gauge theory which fully matches the full temporal dynamics of different spring-chain models. In these situations, the dimension-adaptive algorithm fully resolved the modeling dimension and reduced to a full grid in the corresponding dimension.

The concept of anisotropic modeling accuracies in a refinement algorithm becomes even more important in applications that involve high modeling costs. In molecular dynamics of correlated systems, for instance, the cost increases exponentially with the number of many-body potentials that are taken into account. In such an application, the computational effort can be dramatically reduced if a reduced modeling complexity can be achieved. This thesis demonstrated that the general idea of a modeling dimension which was then used in a refinement algorithm is applicable. Beside the numerical accuracy, this was a further key concept to overcome the curse of dimensionality in this multiscale physical problem.

Future perspectives

We conclude this thesis by considering possible future directions of research related to high dimensional problems in polymer physics:

Modeling of spring-chain systems. In this thesis we considered spring-chain systems with up to five spring segments. As indicated before, it is expected that the dynamical behavior of these system converges to the complex behavior of polymer molecules in nature. Consequently, we propose the simulation of systems with hundreds of spring-segments which, due to the huge complexity, is not possible at the moment. Furthermore, an optimal gauge theory for such a system is still unclear and maybe even problem-dependent.

Different spring forces. We primarily used the nonlinear FENE spring force to account for the elasticity of the fluid. Although the FENE spring force is widely accepted in the literature, we could very well consider more general spring force models such as Cohen's Padé approximation described in Section 1.2.3. A recent analysis of different spring chain models is performed in Kröger [79].

Uncertainty Quantification. As mentioned before, our adaptively computed index sets could be used for a non-intrusive moment estimation in UQ. A typical application related to UQ and to global optimization would be the turbulent drag reduction of an oil flow in a pipe. It is known from the literature that a small amount of a polymeric additive can reduce the amount of turbulence by 30%. Consequently, the oil flow rate in a pipe can be increased which leads to enormous cost reductions. However, the optimal material properties of non-Newtonian polymer additives are an area of active research.

Fokker-Planck based modeling. If we directly solved the Fokker-Planck equation instead of its stochastic interpretation, we would obtain a noise free solution. On the downside, the deterministic Fokker-Planck equation of an N-segment spring chain is of dimensionality $3N + 4$. On a full grid, the numerical problem cannot be tackled so that only some early stage results exist in the literature; see e.g. Knezevic and Süli [73]. It is therefore essential to employ a more advanced approach to overcome the limitations of the exponential cost increase. A first approach on sparse grids is given by Delaunay et al. [34] for a steady flow problem with a low-dimensional configuration space. We propose the application of sparse grids, such as the combination technique used in this thesis, to high-dimensional Fokker-Planck equations. Even then, however, the problem is barely solvable on current parallel clusters if $N > 1$, i.e. if more than one spring segment is considered.

In summary, this thesis represents a first yet very important step towards a complete numerical description of non-Newtonian fluids in nature. And still, much further research is required in order to tackle the complexity of high-dimensional polymer models.

Bibliography

[1] H. Alt. *Lineare Funktionalanalysis: Eine anwendungsorientierte Einführung*. Springer, 2012.

[2] J. Barrett, C. Schwab, and E. Süli. Existence of global weak solutions for some polymeric flow models. *Math Models Methods Appl Sci*, 15(06):939–983, 2005.

[3] J. Barrett and E. Sueli. Existence and equilibration of global weak solutions to kinetic models for dilute polymers ii: Hookean-type models. *Math Model Method Appl Sci*, 22(05):1150024, 2012.

[4] J. Barrett and E. Süli. Existence of global weak solutions to some regularized kinetic models for dilute polymers. *Multiscale Model Simul*, 6(2):506–546, 2007.

[5] J. Barrett and E. Süli. Existence of global weak solutions to dumbbell models for dilute polymers with microscopic cut-off. *Math Model Method Appl Sci*, 18(06):935–971, 2008.

[6] J. Barrett and E. Süli. Existence and equilibration of global weak solutions to kinetic models for dilute polymers i: finitely extensible nonlinear bead-spring chains. *Math Model Method Appl Sci*, 21(06):1211–1289, 2011.

[7] J. Barrett and E. Süli. Existence of global weak solutions to finitely extensible nonlinear bead–spring chain models for dilute polymers with variable density and viscosity. *J Differ Equ*, 253(12):3610–3677, 2012.

[8] H. Bauer. *Measure and Integration Theory*, volume 26. Walter de Gruyter, 2001.

[9] J. Bell, P. Colella, and H. Glaz. A second-order projection method for the incompressible Navier-Stokes equations. *J Comput Phys*, 85(2):257–283, 1989.

[10] H. Bend. Non-Newtonian fluid on a speaker cone. http://www.youtube.com/watch?v=3zoTKXXNQIU, November 2008. accessed: 2016-05-30.

[11] R. Bird, R. Armstrong, O. Hassager, and C. Curtiss. *Dynamics of Polymeric Liquids, Volume 1: Fluid Mechanics*. Wiley New York, 1987.

[12] R. Bird, O. Hassager, R. Armstrong, and C. Curtiss. *Dynamics of Polymeric Liquids, Volume 2: Kinetic Theory*. Wiley New York, 1987.

[13] G. Böhme. *Strömungsmechanik Nichtnewtonscher Fluide*. Vieweg + Teubner Verlag, 2000.

[14] J. Bonvin. *Numerical simulation of viscoelastic fluids with mesoscopic models*. PhD thesis, École polytechnique fédérale de Lausanne, 2000.

[15] J. Bonvin and M. Picasso. A finite element/ Monte-Carlo method for polymer dilute solutions. *Comput Vis Sci*, 4(2):93–98, 2001.

[16] Z. Botev, J. Grotowski, and D. Kroese. Kernel density estimation via diffusion. *Ann Stat*, 38(5):2916–2957, 2010.

[17] H. Bungartz and M. Griebel. A note on the complexity of solving Poisson's equation for spaces of bounded mixed derivatives. *J Complex*, 15(2):167–199, 1999.

[18] H. Bungartz and M. Griebel. Sparse grids. *Acta Numerica*, 13:147–269, 2004.

[19] H. Bungartz, M. Griebel, D. Röschke, and C. Zenger. Pointwise convergence of the combination technique for the Laplace equation. *East-West J Numer Math*, 2:21–45, 1994.

[20] H. Bungartz, M. Griebel, D. Röschke, and C. Zenger. A proof of convergence for the combination technique for the Laplace equation using tools of symbolic computation. *Math Comput Simul*, 42(4):595–605, 1996.

[21] H. Bungartz, M. Griebel, and U. Rüde. Extrapolation, combination, and sparse grid techniques for elliptic boundary value problems. *Comput Meth Appl Mech Eng*, 116(1):243–252, 1994.

[22] R. E. Caflisch. Monte Carlo and quasi-Monte Carlo methods. *Acta Numerica*, 7:1–49, 1998.

[23] E. Carew, P. Townsend, and M. Webster. Taylor-Galerkin algorithms for viscoelastic flow: application to a model problem. *Numer Method Partial Differ Equ*, 10(2):171–190, 1994.

[24] J. Charney, R. Fjörtoft, and J. von Neumann. Numerical integration of the barotropic vorticity equation. *Tellus*, 2(4):237–254, 1950.

[25] S. Claus. Numerical simulation of unsteady three-dimensional viscoelastic Oldroyd-B and Phan-Thien Tanner flows. Diplomarbeit, Institut für Numerische Simulation, Universität Bonn, 2008.

[26] A. Cohen. A Padé approximant to the inverse Langevin function. *Rheologica Acta*, 30(3):270–273, 1991.

[27] O. Coronado, D. Arora, M. Behr, and M. Pasquali. A simple method for simulating general viscoelastic fluid flows with an alternate log-conformation formulation. *J Non-Newton Fluid Mech*, 147(3):189–199, 2007.

[28] R. Courant, K. Friedrichs, and H. Lewy. On the partial difference equations of mathematical physics. *IBM J*, 11(2):215–234, 1967.

[29] R. Croce. *Numerische Simulation der Interaktion von inkompressiblen Zweiphasenströmungen mit Starrkörpern in drei Raumdimensionen*. Dissertation, Institut für Numerische Simulation, Universität Bonn, Bonn, Germany, 2010.

[30] R. Croce, M. Griebel, and M. Schweitzer. Numerical simulation of bubble and droplet-deformation by a level set approach with surface tension in three dimensions. *Int J Numer Meth Fluids*, 62(9):963–993, 2009.

[31] R. Croce, M. Griebel, and M. Schweitzer. Numerical simulation of bubble and droplet-deformation by a level set approach with surface tension in three dimensions. *Int J Numer Methods Fluids*, 62(9):963–993, 2009.

[32] I. Csiszár. Information-type measures of difference of probability distributions and indirect observations. *Studia Sci Math Hungar*, 2:299–318, 1967.

[33] P. de Gennes. *Scaling Concepts in Polymer Physics*. Cornell University Press, 1979.

[34] P. Delaunay, A. Lozinski, and R. Owens. Sparse tensor-product Fokker-Planck-based methods for nonlinear bead-spring chain models of dilute polymer solutions. In *CRM Proc Lect Notes*, volume 41, 2007.

[35] J. Dick, F. Kuo, and I. Sloan. High-dimensional integration: the quasi-Monte Carlo way. *Acta Numerica*, 22:133–288, 2013.

[36] R. Fattal and R. Kupferman. Constitutive laws for the matrix-logarithm of the conformation tensor. *J Non-Newton Fluid Mech*, 123(2):281–285, 2004.

[37] R. Fattal and R. Kupferman. Time-dependent simulation of viscoelastic flows at high Weissenberg number using the log-conformation representation. *J Non-Newton Fluid Mech*, 126(1):23–37, 2005.

[38] C. Fefferman. Existence and smoothness of the Navier-Stokes equation. *The millennium prize problems*, pages 57–67, 2006.

[39] W. Feller. *An Introduction to Probability Theory and its Applications*, volume 2. John Wiley & Sons, 2008.

[40] J. Garcke. *Maschinelles Lernen durch Funktionsrekonstruktion mit verallgemeinerten dünnen Gittern*. Doktorarbeit, Institut für Numerische Simulation, Universität Bonn, 2004.

[41] J. Garcke. Regression with the optimised combination technique. In W. Cohen and A. Moore, editors, *Proc 23rd ICML 2006*, pages 321–328, New York, NY, USA, 2006. ACM Press.

[42] J. Garcke and M. Griebel. Classification with sparse grids using simplicial basis functions. *Intell Data Anal*, 6(6):483–502, 2002.

[43] T. Gerstner and M. Griebel. Numerical integration using sparse grids. *Numer Algorithm*, 18(3-4):209–232, 1998.

[44] T. Gerstner and M. Griebel. Dimension–adaptive tensor–product quadrature. *Comput*, 71(1):65–87, 2003.

[45] T. Gerstner and S. Heinz. Dimension- and time-adaptive multilevel Monte Carlo methods. In *Lecture Notes in Comp Sci and Eng 88 - Sparse Grids and Appl*, pages 107–120. Springer, 2013.

[46] I. Ghosh, G. McKinley, R. Brown, and R. Armstrong. Deficiencies of FENE dumbbell models in describing the rapid stretching of dilute polymer solutions. *J Rheol*, 45(3):721–758, 2001.

[47] M. Giles. Multilevel Monte Carlo path simulation. *Operations Research*, 56:607–617, 2008.

[48] M. Griebel, T. Dornseifer, and T. Neunhoeffer. *Numerical Simulation in Fluid Dynamics, a Practical Introduction*. SIAM, Philadelphia, 1998.

[49] M. Griebel, H. Harbrecht, and M. Peters. Multilevel quadrature for elliptic parametric partial differential equations on non-nested meshes. 2015. INS Preprint No. 1521. Submitted to Stoch Partial Differ Equ: Anal Comput.

[50] M. Griebel, W. Huber, and C. Zenger. Numerical turbulence simulation on a parallel computer using the combination method. In E. Hirschel, editor, *Flow Simulation with High-performance Computers, DFG-SPP*, volume 52 of *Notes Numer Fluid Mech*, pages 34–47. Vieweg, Wiesbaden, 1996.

[51] M. Griebel and A. Rüttgers. Multiscale simulations of three-dimensional viscoelastic flows in a square-square contraction. *J Non-Newtonian Fluid Mech*, 205:41–63, 2014.

[52] M. Griebel and A. Rüttgers. Simulation of dilute polymeric fluids in a three-dimensional contraction using a multiscale FENE model. volume 1593 of *AIP Conf Proc*, pages 539–543, 2014.

[53] M. Griebel, M. Schneider, and C. Zenger. A combination technique for the solution of sparse grid problems. In P. de Groen and R. Beauwens, editors, *Iterative Meth Lin Alg*, pages 263–281. Elsevier, 1992.

[54] M. Griebel and V. Thurner. The efficient solution of fluid dynamics problems by the combination technique. *Intern J Numer Math Heat & Fluid Flow*, 5(3):251–269, 1995.

[55] A. Haji-Ali, F. Nobile, and R. Tempone. Multi-index Monte Carlo: when sparsity meets sampling. *Numer Math*, pages 1–40, 2015.

[56] H. Harbrecht, M. Peters, and M. Siebenmorgen. On multilevel quadrature for elliptic stochastic partial differential equations. In *Lecture Notes in Comp Sci and Eng 88 - Sparse Grids and Appl*, pages 161–179. Springer, 2013.

[57] M. Hegland. Adaptive sparse grids. *Anziam J*, 44:335–353, 2003.

[58] M. Hegland, J. Garcke, and V. Challis. The combination technique and some generalisations. *Lin Alg Appl*, 420(2–3):249–275, 2007.

[59] M. Herrchen and H. Öttinger. A detailed comparison of various FENE dumbbell models. *J Non-Newtonian Fluid Mech*, 68(1):17–42, 1997.

[60] L. Höök, T. Johnson, and T. Hellsten. Randomized quasi-Monte Carlo simulation of fast-ion thermalization. *Comput Sci Discov*, 5(1):014010, 2012.

[61] M. Hulsen, A. Van Heel, and B. Van Den Brule. Simulation of viscoelastic flows using Brownian configuration fields. *J Non-Newton Fluid Mech*, 70(1):79–101, 1997.

[62] F. Industries. How TURBOFLO works. `http://www.flowchem-dra.com/index.php/how-turboflo-works.html`, 2015. accessed: 2016-05-21.

[63] G. Jiang and C. Shu. Efficient implementation of weighted ENO schemes. *J Comput Phys*, 126:202–228, 1996.

[64] S. Jin and L. Collins. Dynamics of dissolved polymer chains in isotropic turbulence. *New J Phys*, 9(10):360, 2007.

[65] D. Joseph. *Fluid Dynamics of Viscoelastic Liquids*. Springer New York, 1990.

[66] B. Jourdain, C. Le Bris, T. Lelièvre, and F. Otto. Long-time asymptotics of a multiscale model for polymeric fluid flows. *Arch Ration Mech Anal*, 181(1):97–148, 2006.

[67] B. Jourdain, T. Lelièvre, and C. Le Bris. Existence of solution for a micro–macro model of polymeric fluid: the FENE model. *J Funct Anal*, 209(1):162–193, 2004.

[68] R. Keunings. On the high Weissenberg number problem. *J Non-Newton Fluid Mech*, 20:209–226, 1986.

[69] R. Keunings. On the Peterlin approximation for finitely extensible dumbbells. *J Non-Newton Fluid Mech*, 68(1):85–100, 1997.

[70] R. Keunings. Micro-macro methods for the multiscale simulation of viscoelastic flow using molecular models of kinetic theory. *Rheol. Rev.*, pages 67–98, 2004.

[71] S. Knapek. *Approximation und Kompression mit Tensorprodukt-Multiskalenräumen*. Dissertation, Universität Bonn, April 2000.

[72] P. Knechtges, M. Behr, and S. Elgeti. Fully-implicit log-conformation formulation of constitutive laws. *J Non-Newton Fluid Mech*, 214:78–87, 2014.

[73] D. Knezevic and E. Süli. Spectral galerkin approximation of Fokker-Planck equations with unbounded drift. *ESAIM: Math Model Numer Anal*, 43(03):445–485, 2009.

[74] H. Koch and D. Tataru. Well-posedness for the Navier–Stokes equations. *Adv Math*, 157(1):22–35, 2001.

[75] A. Koppol, R. Sureshkumar, A. Abedijaberi, and B. Khomami. Anomalous pressure drop behaviour of mixed kinematics flows of viscoelastic polymer solutions: a multiscale simulation approach. *J. Fluid Mech.*, 631:231–251, 2009.

[76] A. Koppol, R. Sureshkumar, and B. Khomami. An efficient algorithm for multiscale flow simulation of dilute polymeric solutions using bead-spring chains. *J Non-Newton Fluid Mech*, 141(2):180–192, 2007.

[77] C. Kowitz, D. Pflüger, F. Jenko, and M. Hegland. The combination technique for the initial value problem in linear gyrokinetics. In *Sparse Grids and Applications*, pages 205–222. Springer, 2012.

[78] C. Kranz. *Untersuchungen zur Kombinationstechnik bei der numerischen Strömungssimulation auf versetzten Gittern*. PhD thesis, Technische Universität München, 2002.

[79] M. Kröger. Simple, admissible, and accurate approximants of the inverse Langevin and Brillouin functions, relevant for strong polymer deformations and flows. *J Non-Newton Fluid Mech*, 223:77–87, 2015.

[80] O. Ladyzhenskaya. Solution in the large to the boundary value problem for the Navier-Stokes equations in two space variables. *Sov Phys Dokl*, 123(3):1128–1131, 1958.

[81] B. Lapeyre, E. Pardoux, and R. Sentis. *Introduction to Monte Carlo Methods for Transport and Diffusion Equations*, volume 6. Oxford University Press, 2003.

[82] M. Laso and H. Öttinger. Calculation of viscoelastic flow using molecular models: the CONNFFESSIT approach. *J Non-Newton Fluid Mech*, 47:1–20, 1993.

[83] C. Le Bris and T. Lelièvre. Multiscale modelling of complex fluids: a mathematical initiation. In *Multiscale Modeling and Simulation in Science*, pages 49–137. Springer, 2009.

[84] C. Le Bris and T. Lelièvre. Micro-macro models for viscoelastic fluids: modelling, mathematics and numerics. *Sci China Math*, 55(2):353–384, 2012.

[85] B. Leonard. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Comput Meth Appl Mech Eng*, 19(1):59–98, 1979.

[86] J. Leray. Sur le mouvement d'un liquide visqueux emplissant l'espace. *Acta Math*, 63(1):193–248, 1934.

[87] P. Lions and N. Masmoudi. Global existence of weak solutions to some micro-macro models. *Comptes Rendus Mathematique*, 345(1):15–20, 2007.

[88] A. Lozinski. *Spectral methods for kinetic theory models of viscoelastic fluids*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 2003.

[89] A. Lozinski, R. Owens, and T. Phillips. The Langevin and Fokker-Planck equations in polymer rheology. In *Handbook of Numerical Analysis*, volume 16, pages 211–304. Elsevier, 2011.

[90] C. Macosko and R. Larson. *Rheology: Principles, Measurements, and Applications*. Wiley-Vch New York, 1994.

[91] C. Mangoubi, M. Hulsen, and R. Kupferman. Numerical stability of the method of Brownian configuration fields. *J Non-Newton Fluid Mech*, 157(3):188–196, 2009.

[92] N. Masmoudi. Well-posedness for the fene dumbbell model of polymeric flows. *Commun Pure Appl Math*, 61(12):1685–1714, 2008.

[93] N. Masmoudi. Global existence of weak solutions to the FENE dumbbell model of polymeric flows. *Inven Math*, 191(2):427–500, 2013.

[94] M. Melchior and H. Öttinger. Variance reduced simulations of stochastic differential equations. *J Chem Phys*, 103:9506–9509, 1995.

[95] M. Melchior and H. Öttinger. Variance reduced simulations of polymer dynamics. *J Chem Phys*, 105:3316–3331, 1996.

[96] B. Metsch. *Algebraic Multigrid (AMG) for Saddle Point Systems*. Dissertation, Institut für Numerische Simulation, Universität Bonn, 2013.

[97] N. Nguyen and G. Ökten. The acceptance-rejection method for low-discrepancy sequences. *arXiv preprint arXiv:1403.5599*, 2014.

[98] J. Oldroyd. On the formulation of rheological equations of state. In *Proc Royal Soc Lond A: Math, Phys Eng Sci*, volume 200, pages 523–541. The Royal Society, 1950.

[99] H. Öttinger. *Stochastic Processes in Polymeric Fluids: Tools and Examples for Developing Simulation Algorithms*. Springer, Berlin, 1996.

[100] R. Owens and T. Phillips. *Computational Rheology*. Lectures in Environmental Sciences. Imperial College Press, London, 2. edition, 2005.

[101] E. Parzen. On estimation of a probability density function and mode. *Ann Math Stat*, 33(3):1065–1076, 1962.

[102] A. Peterlin. Hydrodynamics of macromolecules in a velocity field with longitudinal gradient. *J Polym Sci Part B: Polym Lett*, 4(4):287–291, 1966.

[103] D. Pflüger. *Spatially Adaptive Sparse Grids for High-Dimensional Problems*. Verlag Dr. Hut, München, 2010.

[104] D. Pflüger, H. Bungartz, M. Griebel, F. Jenko, T. Dannert, M. Heene, C. Kowitz, A. P. Hinojosa, and P. Zaspel. Exahd: an exa-scalable two-level sparse grid approach for higher-dimensional problems in plasma physics and beyond. In *Euro-Par 2014: Parallel Processing Workshops*, pages 565–576. Springer, 2014.

[105] N. Phan-Thien. A nonlinear network viscoelastic model. *J Rheol*, 22(3):259–283, 1978.

[106] N. Phan-Thien and R. Tanner. A new constitutive equation derived from network theory. *J Non-Newton Fluid Mech*, 2(4):353–365, 1977.

[107] L. Plaskota. The exponent of discrepancy of sparse grids is at least 2.1933. *Adv Comput Math*, 12(1):3–24, 2000.

[108] J. Prieto, R. Bermejo, and M. Laso. A semi-Lagrangian micro–macro method for viscoelastic flow calculations. *J Non-Newtonian Fluid Mech*, 165(3):120–135, 2010.

[109] Psidot. Fano flow. `http://www.youtube.com/watch?v=aY7xiGQ-7iw`, July 2007. accessed: 2016-05-28.

[110] Psidot. The Barus effect. `http://www.youtube.com/watch?v=KcNWLIpv8gc`, July 2007. accessed: 2016-05-30.

[111] Psidot. The Weissenberg effect. `http://www.youtube.com/watch?v=npZzlgKjs0I`, July 2007. accessed: 2016-05-30.

[112] Psidot. Three viscoelastic effects. `http://www.youtube.com/watch?v=nX6GxoiCneY`, July 2007. accessed: 2016-05-28.

[113] C. Reisinger. Analysis of linear difference schemes in the sparse grid combination technique. *IMA J Numer Anal*, pages 544–581, 2012.

[114] M. Renardy. Mathematical analysis of viscoelastic flows. *Annu Rev Fluid Mech*, 21(1):21–34, 1989.

[115] M. Renardy. An existence theorem for model equations resulting from kinetic theories of polymer solutions. *SIAM J Math Anal*, 22(2):313–327, 1991.

[116] C. Roth. *Stochastische Partielle Differentialgleichungen 1. Ordnung*. Dissertation, Halle-Wittenberg, Germany, 2002.

[117] A. Rüttgers. Multiscale modelling of dilute polymeric fluids with stochastic and Fokker-Planck-based methods. Diplomarbeit, Institut für Numerische Simulation, Universität Bonn, 2010.

[118] A. Rüttgers, M. Griebel, L. Pastrik, H. Schmied, D. Wittmann, A. Scherrieble, A. Dinkelmann, and T. Stegmaier. Simulation of the oil storage process in the scopa of specialized bees. *Comput & Fluids*, 119:115–130, 2015. also available as INS Preprint No. 1404.

[119] J. Schieber and H. Öttinger. The effects of bead inertia on the Rouse model. *J Chem Phys*, 89(11):6972–6981, 1988.

[120] B. Smith, P. Bjørstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, Cambridge, 2004.

[121] S. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. In *Dokl Akad Nauk SSSR*, volume 4, page 123, 1963.

[122] P. Sousa, P. Coelho, M. Oliveira, and M. Alves. Three-dimensional flow of Newtonian and Boger fluids in square–square contractions. *J Non-Newtonian Fluid Mech*, 160(2):122–139, 2009.

[123] P. Sousa, P. Coelho, and M. Oliveira, Mand Alves. Effect of the contraction ratio upon viscoelastic fluid flow in three-dimensional square–square contractions. *Chem Eng Sci*, 66(5):998–1009, 2011.

[124] A. Squillacote and J. Ahrens. *The Paraview Guide*, volume 366. Kitware, 2007.

[125] R. Srinivasan. *Importance Sampling: Applications in Communications and Detection*. Springer Science & Business Media, 2013.

[126] D. Trebotich, P. Colella, and G. Miller. A stable and convergent scheme for viscoelastic flow in contraction channels. *J Comput Phys*, 205(1):315–342, 2005.

[127] E. Vanmarcke. *Random Fields: Analysis and Synthesis*. World Scientific, 2010.

[128] R. Vargas, O. Manero, and T. Phillips. Viscoelastic flow past confined objects using a micro–macro approach. *Rheol Acta*, 48(4):373–395, 2009.

[129] G. Venkiteswaran and M. Junk. A QMC approach for high dimensional Fokker-Planck equations modelling polymeric liquids. *Math Comput Simul*, 68(1):43–56, 2005.

[130] B. Verleye, M. Klitz, R. Croce, D. Roose, S. Lomov, and I. Verpoest. Computation of the permeability of textiles with experimental validation for monofilament and non crimp fabrics. In *Computational Textile*, volume 55 of *Studies Comput Intel*, chapter 6, pages 93–110. Springer Berlin / Heidelberg, 2007.

[131] J. von Neumann. Various techniques used in connection with random digits. Monte Carlo methods. *Nat Bur Stand*, 12:36–38, 1951.

[132] X. Wang. Improving the rejection sampling method in quasi-Monte Carlo methods. *J Comput Appl Math*, 114(2):231–246, 2000.

[133] P. Wapperom, R. Keunings, and V. Legat. The backward-tracking Lagrangian particle method for transient viscoelastic flows. *J Non-Newtonian Fluid Mech*, 91(2):273–295, 2000.

[134] H. Warner. Kinetic theory and rheology of dilute suspensions of finitely extendible dumbbells. *Ind Eng Chem Fundam*, 11(3):379–387, 1972.

[135] G. Wasilkowski and H. Woźniakowski. Weighted tensor product algorithms for linear multivariate problems. *J Complex*, 15(3):402–447, 1999.

[136] N. Waters and M. King. Unsteady flow of an elastico-viscous liquid. *Rheol. Acta*, 9(3):345–355, 1970.

[137] D. Werner. *Funktionalanalysis*. Springer, 2011.

[138] F. White. *Viscous Fluid Flow*. McGraw-Hill, New York, 1991.

[139] J. Wiest and R. Tanner. Rheology of bead-nonlinear spring chain macromolecules. *J Rheol*, 33(2):281–316, 1989.

[140] P. Zaspel and M. Griebel. Solving incompressible two-phase flows on multi-GPU clusters. *Comput & Fluids*, 80(0):356 – 364, 2013.

[141] C. Zenger. Sparse grids. In W. Hackbusch, editor, *Parallel Algorithms for Partial Differential Equations*, volume 31 of *Notes Numer Fluid Mech*, pages 241–251. Vieweg, 1991.