# Multiscale simulation of polymeric fluids using the sparse grid combination technique

A. Rüttgers[a,*], M. Griebel[b,c]

[a]German Aerospace Center (DLR), Simulation and Software Technology, Linder Höhe, 51147 Cologne, Germany
[b]Institute for Numerical Simulation (INS), University of Bonn, Wegelerstr. 6, 53115 Bonn, Germany
[c]Fraunhofer Institute for Algorithms and Scientific Computing (SCAI), Schloss Birlinghoven, 53754 Sankt Augustin , Germany

## Abstract

We present a computationally efficient sparse grid approach to allow for multiscale simulations of non-Newtonian polymeric fluids. Multiscale approaches for polymeric fluids often involve model equations of high dimensionality. A conventional numerical treatment of such equations leads to computing times in the order of months even on massively parallel computers.

For a reduction of this enormous complexity, we propose the sparse grid combination technique. Compared to classical full grid approaches, the combination technique strongly reduces the computational complexity of a numerical scheme but only slightly decreases its accuracy.

Here, we use the combination technique in a general formulation that balances not only different discretization errors but also considers the accuracy of the mathematical model. For an optimal weighting of these different problem dimensions, we employ a dimension-adaptive refinement strategy. We finally verify substantial cost reductions of our approach for simulations of non-Newtonian Couette and extensional flow problems.

*Keywords:* sparse grids, combination technique, dimension-adaptivity, multiscale simulation, Brownian configuration fields, multi-bead spring-chains

## 1. Introduction

Many fluids from the chemical industry and from nature show a non-Newtonian behavior. In the literature, the modeling of non-Newtonian fluids is usually based on an additional elastic stress tensor in the fluid equations. Then, macroscopic approaches compute the entries of this stress tensor by solving an additional differential or integral constitutive equation. An overview of macroscopic approaches is given in the book by Owens and Philips [1].

Macroscopic models have a low computational complexity but exhibit two serious drawbacks: They are prone to numerical instabilities, the so called *High Weissenberg number problem* (HWNP), and they have limited modeling capabilities. Both drawbacks can be avoided in a multiscale approach. There, the kinetic equations of the microscopic polymeric structure are directly modeled which allows for a better description of real polymers. Furthermore, immunity to the type of instability caused by the HWNP seems to result, as reported by Mangoubi, Hulsen and Kupferman [2]. A more detailed explanation of multiscale models can be found in a survey by Keunings [3].

Despite their advantages compared to macroscopic models, multiscale models are hardly used in practical applications. This is due to their enormous computational complexity. Depending on the degrees

*Corresponding author. Tel.: +49 2203 601-3754; fax: +49 2203 67310
*Email addresses:* alexander.ruettgers@dlr.de (A. Rüttgers), griebel@ins.uni-bonn.de (M. Griebel)

*The final publication is available at http://dx.doi.org/10.1016/j.amc.2017.04.025* *November 21, 2017*

of freedom of the polymeric microstructure, the configuration space for the polymer model can be high-dimensional. For instance, it is of dimensionality fifteen for a model that consists of five spring segments. This leads to an exponential growth in the complexity of grid-based numerical approaches which is often described by the term *curse of dimensionality*.

In the following, we aim for a reduction of this multiscale model complexity. For this purpose, we employ the sparse grid combination technique to cope with the curse of dimensionality. The combination technique [4] is a specific sparse grid representation [5]. The basic idea is to linearly combine a sequence of coarse grid numerical solutions to approximate a numerical solution on a fine grid. Since an optimal balancing of the different problem dimensions is not a priori clear, we employ a dimension-adaptive refinement strategy for the combination technique. Furthermore, our novel approach balances not only errors that result from the discretization schemes but also considers the error of the modeling equation which stems from the choice of the spring-chain length.

The first application of sparse grids to non-Newtonian fluids was given by Delaunay, Lozinski and Owens [6]. In contrast to our stochastic polymer model, the authors employ a deterministic Fokker-Planck-based approach to model the high-dimensional configuration space. Furthermore, there exist several literature results on sparse grids for Newtonian fluids, see, e.g., Griebel and Koster [7] for results on turbulent flows. An alternative approach in the literature to cope with the high complexity of multiscale non-Newtonian models is the proper generalized decomposition (PGD). The PGD is a model reduction technique that bases on a separated representation of the unknown field. An overview of the PGD for applications related to non-Newtonian fluids is given by Chinesta et al. [8].

The remainder of this article is organized as follows: First, in Section 2, we discuss the governing equations of our multiscale polymer model. Then, in Section 3 we cover techniques related to the numerical treatment of the high-dimensional equations. This includes the spatial and temporal discretization of the partial and stochastic differential equations in Section 3.1 and the dimension-adaptive combination technique in Section 3.2. Finally in Section 4, we present the resulting numerical outcomes for Couette and extensional flows. Moreover, we verify that our approach actually reduces the complexity compared to classical full grid approaches.

## 2. Governing equations for the multiscale model

We investigate fluid flow in a bounded domain $\mathcal{O} \subset \mathbb{R}^d$ with $d = 1, 2, 3$ depending on the specific application and refer to $\mathcal{O}$ as physical space. For given position $\boldsymbol{x} \in \mathcal{O}$ and at any time $t \in \mathcal{T} = [0, T] \subset \mathbb{R}$ the current state of a non-Newtonian fluid is described by the fluid's velocity field $\boldsymbol{u} : (\boldsymbol{x}, t) \in \mathcal{O} \times \mathcal{T} \mapsto \boldsymbol{u}(\boldsymbol{x}, t) \in \mathbb{R}^d$, the hydrodynamic pressure field $p : (\boldsymbol{x}, t) \in \mathcal{O} \times \mathcal{T} \mapsto p(\boldsymbol{x}, t) \in \mathbb{R}$ and the polymeric stress tensor field $\boldsymbol{\tau_p} : (\boldsymbol{x}, t) \in \mathcal{O} \times \mathcal{T} \mapsto \boldsymbol{\tau_p}(\boldsymbol{x}, t) \in \mathbb{R}^{3\times 3}$ with corresponding initial and boundary conditions.

The dimensionless conservation of mass and momentum is given by the coupled system

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

$$\nabla \cdot \boldsymbol{u}(\boldsymbol{x},t) = 0 \tag{2}$$

Equations (1) contains the dimensionless parameters $Re$ (Reynolds number) and $\beta$ (viscosity ratio). They are defined as

$$Re = \frac{\rho_c U_c L_c}{\eta_s + \eta_p}, \qquad \beta = \frac{\eta_s}{\eta_s + \eta_p} \tag{3}$$

with the characteristic units $L_c \in \mathbb{R}^+$ (characteristic length in macroscopic flow), $U_c \in \mathbb{R}^+$ (characteristic fluid velocity), $\rho_c \in \mathbb{R}^+$ (fluid density, scaling pressure term with $1/(\rho U_c{}^2)$), $\eta_s \in \mathbb{R}^+$ (Newtonian dynamic viscosity) and $\eta_p \in \mathbb{R}^+$ (zero shear rate polymer dynamic viscosity).

These equations are coupled with the initial conditions $\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0(\boldsymbol{x})$, $p(\boldsymbol{x}, 0) = p_0(\boldsymbol{x})$ and $\boldsymbol{\tau}_p(\boldsymbol{x}, 0) =$

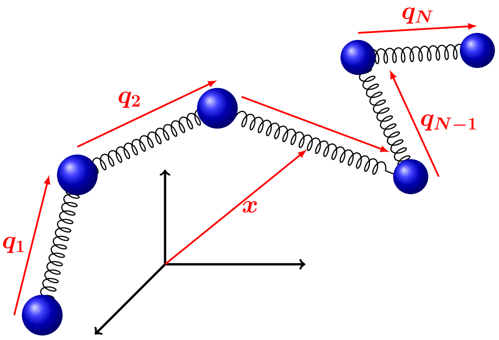


Figure 1: Modeling of a polymeric molecule as a multi-bead spring-chain.

$\boldsymbol{\tau}_0(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{O}$. Furthermore, one of the conditions

$$\boldsymbol{u}|_{\Gamma_1} = \boldsymbol{u}_0 \qquad \text{on the inflow boundary } \Gamma_1, \tag{4a}$$

$$\boldsymbol{u}|_{\Gamma_2} = 0 \qquad \text{on the no-slip boundary } \Gamma_2, \tag{4b}$$

$$\partial_{\boldsymbol{n}}(\boldsymbol{u} \cdot \boldsymbol{n})|_{\Gamma_3} = 0,\ \partial_{\boldsymbol{n}}(\boldsymbol{u} \cdot \boldsymbol{t})|_{\Gamma_3} = 0 \qquad \text{on the outflow boundary } \Gamma_3 \tag{4c}$$

holds for the velocity field on the boundary $\partial\mathcal{O} = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3$ with $\boldsymbol{n}$ as outward pointing unit normal and $\boldsymbol{t}$ as tangential vector on $\partial\mathcal{O}$.

The polymeric stress tensor $\boldsymbol{\tau}_p$ reflects the elastic stress contribution from the underlying polymeric structure. In a multiscale approach, the microscopic structure is modeled as an arrangement of $N+1$ beads that are connected with $N$ elastic springs. We illustrate the resulting multi-bead spring-chain in Fig. 1. A chain is fully described by its center position $\boldsymbol{x} \in \mathcal{O}$ and its spring segment configuration vectors $\boldsymbol{q_i} \in D_i \subset \mathbb{R}^3$ for $i = 1, \ldots, N$. For the description of a spring chain ensemble there are two different stochastic approaches. They can either be described by a sequence of $N+1$ stochastic processes with realizations in $\mathcal{O} \times D_1 \times \ldots \times D_N$ or, alternatively, by a sequence of $N$ time-dependent random fields $\boldsymbol{Q}_1(\boldsymbol{x},t), \ldots, \boldsymbol{Q}_N(\boldsymbol{x},t)$ with realizations in $D_1 \times \ldots \times D_N$ for any $(\boldsymbol{x},t) \in \mathcal{O} \times \mathcal{T}$. The numerical treatment of the first formulation leads to the CONNFFESSIT method (**C**alculation **o**f **N**on-**N**ewtonian **F**low: **F**inite **E**lements and **S**tochastic **Si**mulation **T**echniques) which was introduced by Laso and Öttinger [9]. In the following, we employ the second stochastic description in which the random fields are denoted as *Brownian configuration fields* (BCF) [10].

The BCF approach evolves each random field $i = 1, \ldots, N$ over time according to the dimensionless stochastic partial differential equation (SPDE)

$$\mathrm{d}\boldsymbol{Q}_i(\boldsymbol{x},t) = \Big(-\boldsymbol{u}(\boldsymbol{x},t)\nabla\boldsymbol{Q}_i(\boldsymbol{x},t) + (\nabla\boldsymbol{u}(\boldsymbol{x},t))^T\boldsymbol{Q}_i(\boldsymbol{x},t) - \frac{1}{4\,De(N)}\sum_{k=1}^{N} A_{ik}\cdot\boldsymbol{F}\left(\boldsymbol{Q}_i(\boldsymbol{x},t)\right)\Big)\,\mathrm{d}t \tag{5}$$
$$+ \frac{1}{\sqrt{2De(N)}}\Big(\mathrm{d}\boldsymbol{W}_{i+1}(t) - \mathrm{d}\boldsymbol{W}_i(t)\Big).$$

Here, we have used the dimensionless parameter $De(N) = \lambda(N)U_c/L_c$ (Deborah number) in which $\lambda(N) \in \mathbb{R}^+$ represents the relaxation time of a polymer segment. Furthermore, $\boldsymbol{A} = \mathrm{diag}(-1,2,-1) \in \mathbb{R}^{N\times N}$ denotes the *Rouse matrix*, $\boldsymbol{F} : D_i \subset \mathbb{R}^3 \to \mathbb{R}^3$ is the elastic spring force and $\boldsymbol{W}_1(t), \ldots, \boldsymbol{W}_{N+1}(t)$ is a sequence of $N+1$ independent and identically distributed (i.i.d.) Wiener processes with first moment $\langle\boldsymbol{W}_i(t)\rangle = 0$ and second moment $\langle\boldsymbol{W}_i(t)\boldsymbol{W}_i(t')\rangle = \min(t,t')\mathbf{Id}$. The Wiener processes represent Brownian forces that act on each of the $N+1$ beads. Theoretically, each process also depends on the spatial variable $\boldsymbol{x}$. However, the BCF method assumes a spatial correlation / homogeneity in space such that the Wiener processes depend on time only which simplifies the numerical treatment later on.

As mentioned before, the SPDE (5) contains a spring force term $\boldsymbol{F}$. Different choices for $\boldsymbol{F}$ are used in the literature. We employ the nonlinear FENE (**f**initely **e**xtensible **n**onlinear **e**lastic) spring force by Warner [11] that is defined as

$$\boldsymbol{F}(\boldsymbol{q}_i) = \frac{\boldsymbol{q}_i}{1 - \|\boldsymbol{q}_i\|^2/b(N)}, \qquad \|\boldsymbol{q}_i\|^2 \leq b(N). \tag{6}$$

The FENE spring force contains a parameter $b(N) \in \mathbb{R}^+$ that can be considered as the normalized maximum extension of each spring segment. The configuration space $D_i$ for a FENE spring is an open ball with radius $b(N)^{1/2} \in \mathbb{R}^+$ centered at the origin. In the asymptotic limit $b(N) \to \infty$, the FENE spring force simplifies to the linear Hookean spring force $\boldsymbol{F}(\boldsymbol{q}_i) = \boldsymbol{q}_i$ in which the spring segment extension is not restricted.

The initial distribution of the random field $\boldsymbol{Q}(\boldsymbol{x}, 0) = (\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_N)(\boldsymbol{x}, 0) \sim \psi_{eq}(\boldsymbol{q}_1, \ldots, \boldsymbol{q}_N)$ also depends on the spring force $\boldsymbol{F}$. Here, the density function $\psi_{eq}$ for the FENE spring force is defined as

$$\psi_{eq}(\boldsymbol{q}_1, \ldots, \boldsymbol{q}_N) = \frac{1}{\left(2\pi b(N)^{3/2} \mathcal{B}(\frac{3}{2}, \frac{b(N)+2}{2})\right)^N} \prod_{i=1}^{N} \left(1 - \frac{\|\boldsymbol{q}_i\|^2}{b(N)}\right)^{b(N)/2} \tag{7}$$

with the beta function $\mathcal{B}(.,.)$ that is defined by

$$\mathcal{B}(x, y) = \int_0^1 s^{x-1}(1-s)^{y-1} \mathrm{d}s \qquad \text{for } x, y > 0. \tag{8}$$

In the case of a Hookean spring force, the equilibrium density is

$$\psi_{eq}(\boldsymbol{q}_1, \ldots, \boldsymbol{q}_N) = \frac{1}{(2\pi)^{(3N)/2}} \prod_{i=1}^{N} \exp\left(-\frac{\|\boldsymbol{q}_i\|^2}{2}\right) \tag{9}$$

which is a 3N-dimensional normal distribution with zero mean vector and the identity as covariance matrix. Note here that both equilibrium densities (7) and (9) are product densities. Furthermore, we prescribe homogeneous Neumann boundary conditions for $\boldsymbol{Q}(\boldsymbol{x}, t)$ on the physical boundary $\partial\mathcal{O}$.

We next discuss the coupling between the macroscopic scale (1)–(2) and the microscopic scale (5). The downscaling from the macroscopic to the microscopic length scale results from the velocity field $\boldsymbol{u}$. The upscaling from the microscopic Brownian configuration fields $\boldsymbol{Q}(\boldsymbol{x}, t)$ to the macroscopic polymer stress tensor $\boldsymbol{\tau_p}$ is given by Kramers' relation [12]. Thus, the stress tensor for a spring chain is related to the random field according to

$$\boldsymbol{\tau_p}(\boldsymbol{x}, t) = \frac{3\,\alpha_{b,d}(1-\beta)}{De(N) \cdot ((N+1)^2 - 1)} \sum_{i=1}^{N} \Big( \langle \boldsymbol{Q}_i(\boldsymbol{x}, t) \otimes \boldsymbol{F}(\boldsymbol{Q}_i(\boldsymbol{x}, t)) \rangle - \mathbf{Id} \Big) \tag{10}$$

with $\otimes : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$ as tensor product of two random fields in $\mathbb{R}^3$, $\langle\cdot\rangle$ as the expectation of the random fields on the product space $D_1 \times \ldots \times D_N$ and $\alpha_{b,d}$ as a spring force dependent constant. This constant is $\alpha_{b,d} = (b(N)+5)/b(N)$ for FENE springs and simplifies to $\alpha_{b,d} = 1$ for Hookean springs when $b(N) \to \infty$.

The rheological behavior of the full polymer chain depends on the number of spring segments $N$. For comparisons, it is advantageous to match the length scale $b(N)$ and the time scale $\lambda(N)$ as two important characteristics of different polymer chains. First, we consider the dumbbell case $N = 1$ for which $b(1)$ and $\lambda(1)$ are not modified since the dumbbell already represents the full chain. In the multi-segment case $N > 1$, we set $b(N) = b(1)/N$ such that each segment represents the $N$th part of the full chain, cf. Ghosh et al. [13]. In the literature, there exist several approaches to gauge the time constant $\lambda(N)$. We equate the zero shear characteristic time scale of all models [14] which leads to

$$\lambda(N) \;=\; \lambda(1)/g \text{ with } g = \frac{b(1)+7}{15b(1)} \cdot \frac{b(N)}{b(N)+5} \cdot \left[ (2(N+1)^2+7) - \frac{12[(N+1)^2+1]}{(N+1)(b(N)+7)} \right]. \tag{11}$$

According to Koppol et al. [15], this mapping ensures a close match of different chain systems in both shear and extensional flows.

## 3. Numerical discretization

In this section we describe the numerical discretization of the multiscale polymer model from Section 2. First, Section 3.1 gives details on the discretization of the coupled system on a full grid and discusses its computational complexity. The discretization of the multiscale model in the dumbbell case, i.e. for $N = 1$, was already described in Griebel and Rüttgers [16]; this section now covers the general case $N > 1$. Then, Section 3.2 introduces the combination technique to reduce the enormous complexity of the polymer model.

### *3.1. Discretization on a full grid*

### *3.1.1. Spatial discretization*

The macroscopic equations for the conservation of momentum and mass (1) and (2) are discretized with a finite difference scheme. For this purpose, the physical flow domain $\mathcal{O}$ is subdivided into $M_g$ rectangular grid cells with discrete positions $\boldsymbol{x}_k$ for $k = 1, \ldots, M_g$. The unknowns $p$ and $\boldsymbol{p}$ are discretized in the cell centers while the three components of the velocity field $\boldsymbol{u}$ are placed on the cell faces. This staggered grid avoids numerical instabilities in the pressure field, the so called checkerboard pattern, since a strong coupling between the pressure and the velocity field can be ensured in the discretization.

The Brownian configuration field is approximated in every cell center by a sequence of $M_s$ stochastic realizations according to $\boldsymbol{Q}^{(j)}(\boldsymbol{x}_k, t)$ for $j = 1, \ldots, M_s$ and $k = 1, \ldots, M_g$. Since the discrete position of $\boldsymbol{Q}$ and the macroscopic stress tensor $\boldsymbol{\tau_p}$ overlap, no additional interpolation is necessary. Furthermore, the spatial derivative of $\boldsymbol{Q}(\boldsymbol{x}, t)$ in (5) is also approximated by a finite difference scheme.

We generate the samples at initial time $t = 0$ from the FENE equilibrium distribution (7) by using von Neumann's rejection sampling [17]. Here, the comparison function is the uniform density restricted on an open ball with radius $b(N)^{1/2}$. Furthermore, we explicitly make use of the fact that (7) is a product density such that the realizations for each spring segment $i$ can be generated independently.

### *3.1.2. Temporal discretization*

We subsequently describe a semi-implicit time integration scheme for the case of a general three-dimensional flow space $\mathcal{O}$. The approach uses an implicit second-order Crank-Nicolson scheme for the diffusive velocity terms in the momentum equations (1). This ansatz avoids CFL-type restrictions on the time step width for the laminar flow regime considered in Section 4. For the convective velocity terms we apply an explicit time discretization scheme since we do not have to solve a nonlinear system of equations then. Next, a fractional step method that leads to a pressure-correction scheme is used to solve the Navier-Stokes equations. This scheme was proposed by Bell et al. [18] and ensures a second-order temporal accuracy of velocity and pressure field provided all velocity terms are discretized with second-order accuracy in time.

Let $\boldsymbol{u}^n$, $p^n$, $\boldsymbol{\tau}_p^n$ and $\boldsymbol{Q}^n$ denote the discretized variables in time step $t^n \in [0, T]$. We then obtain the discrete solution at $t^n + \Delta t^n = t^{n+1} \in (0, T]$ by the following steps:

1. For the discretization of FENE spring force model in (5), we apply one step of a semi-implicit Euler-Maruyama method for all spring segments $i = 1, \ldots, N$, for all $j = 1, \ldots, M_s$ stochastic samples and in all cell centers $\boldsymbol{x}_k$ according to

$$\left(1 + \frac{A_{ii}\Delta t^n}{4\,De(N)}\right) \boldsymbol{F}\left(\boldsymbol{Q}_i^{n+1,(j)}(\boldsymbol{x}_k)\right) = \boldsymbol{Q}_i^{n,(j)}(\boldsymbol{x}_k) + \Bigg( -\boldsymbol{u}^n \nabla \boldsymbol{Q}_i^{n,(j)}(\boldsymbol{x}_k) + (\nabla_{\boldsymbol{x}} \boldsymbol{u}^n)^T \boldsymbol{Q}_i^{n,(j)} - \frac{1}{4\,De(N)} \cdot \\ \sum_{\substack{k=1 \\ k \neq i}}^{N} A_{ik} \cdot \boldsymbol{F}\left(\boldsymbol{Q}_k^{n,(j)}(\boldsymbol{x}_k)\right) \Bigg) \Delta t^n + \sqrt{\frac{\Delta t^n}{2De(N)}} \left(\boldsymbol{\eta}_{i+1}^{(j)} - \boldsymbol{\eta}_i^{(j)}\right) \quad (12)$$

   and solve this equation for $\boldsymbol{Q}_i^{n+1,(j)}(\boldsymbol{x}_k)$. This discrete approach is implicit in the spring force term $\boldsymbol{F}(\boldsymbol{Q}_i)$ for $i = 1, \ldots, N$ and was proposed by Öttinger [19].

Here, $\boldsymbol{\eta}_i^{(j)}, \boldsymbol{\eta}_{i+1}^{(j)} \sim \boldsymbol{N}(0,1)^3$ denote triples of independent Gaussian random variables with zero mean and variance one for $j = 1, \ldots, M_s$.
In case of the linear Hooke model, we perform one step of an explicit Euler-Maruyama method according to

$$\boldsymbol{Q}_i^{n+1,(j)}(\boldsymbol{x}_k) = \boldsymbol{Q}_i^{n,(j)}(\boldsymbol{x}_k) + \Bigg( -\boldsymbol{u}^n \nabla \boldsymbol{Q}_i^{n,(j)}(\boldsymbol{x}_k) + (\nabla_{\boldsymbol{x}} \boldsymbol{u}^n)^T \boldsymbol{Q}_i^{n,(j)}(\boldsymbol{x}_k)$$

$$-\frac{1}{4\,De(N)} \sum_{k=1}^{N} A_{ik} \cdot \boldsymbol{F}\left(\boldsymbol{Q}_k^{n,(j)}(\boldsymbol{x}_k)\right) \Bigg) \Delta t^n + \sqrt{\frac{\Delta t^n}{2De(N)}} \left(\boldsymbol{\eta}_{i+1}^{(j)} - \boldsymbol{\eta}_i^{(j)}\right). \qquad (13)$$

2. The new stress contribution is obtained from Kramers' relation by using Monte Carlo integration according to

$$\boldsymbol{\tau}_p^{n+1}(\boldsymbol{x}_k) = \frac{\alpha_{b,d}(1-\beta)}{De(N)} \cdot \frac{3}{(N+1)^2 - 1} \sum_{i=1}^{N} \left( \frac{1}{M_s} \sum_{j=1}^{M_s} \left( \boldsymbol{Q}_i^{n+1,(j)}(\boldsymbol{x}_k) \otimes \boldsymbol{F}(\boldsymbol{Q}_i^{n+1,(j)}(\boldsymbol{x}_k)) \right) - \mathbf{Id} \right).$$

3. We solve the Helmholtz equation

$$\left(\boldsymbol{Id} - \frac{\Delta t^n \beta}{2Re} \Delta\right) \boldsymbol{u}^*(\boldsymbol{x}_k) = \boldsymbol{u}^n(\boldsymbol{x}_k) - \Delta t^n \left( \nabla p^n(\boldsymbol{x}_k) + \boldsymbol{u}^n \cdot \nabla \boldsymbol{u}^n(\boldsymbol{x}_k) - \frac{\beta}{2Re} \Delta \boldsymbol{u}^n(\boldsymbol{x}_k) - \frac{1}{Re} \nabla \cdot \boldsymbol{\tau}_p^{n+1}(\boldsymbol{x}_k) \right) \qquad (14)$$

to obtain an intermediate velocity field $\boldsymbol{u}^*(\boldsymbol{x}_k)$.

4. The intermediate velocity field $\boldsymbol{u}^*$ is used on the right-hand side of a Poisson problem

$$\Delta \phi^{n+1}(\boldsymbol{x}_k) = \frac{1}{\Delta t^n} \nabla \cdot \boldsymbol{u}^*(\boldsymbol{x}_k). \qquad (15)$$

The resulting pressure correction $\phi^{n+1}$ ensures that $\boldsymbol{u}^{n+1}$ is divergence free. We then compute

$$\boldsymbol{u}^{n+1}(\boldsymbol{x}_k) = \boldsymbol{u}^*(\boldsymbol{x}_k) - \Delta t^n \nabla \phi^{n+1}(\boldsymbol{x}_k). \qquad (16)$$

5. The new pressure field is finally derived as

$$p^{n+1}(\boldsymbol{x}_k) = p^n(\boldsymbol{x}_k) + \phi^{n+1}(\boldsymbol{x}_k) - \frac{\Delta t^n \beta}{2Re} \Delta \phi^{n+1}(\boldsymbol{x}_k). \qquad (17)$$

This description also applies to the case of a lower-dimensional flow space than $d = 3$, i.e. to the cases $d = 1$ and $d = 2$. Then, steps 3-5 for computing the new velocity field simplify accordingly.

### *3.1.3. Computational complexity*

Compared to classical macroscopic approaches for non-Newtonian fluids, multiscale approaches have a much higher computational complexity due to the large number of stochastic realizations. For the case of an $N$-segment chain, the Brownian configuration field $\boldsymbol{Q}(\boldsymbol{x}_k, t)$ is approximated with $j = 1, \ldots, M_s$ samples $\boldsymbol{Q}^{(j)}(\boldsymbol{x}_k, t) \in \mathbb{R}^{3N}$ for every $\boldsymbol{x}_k$. Therefore, the full grid storage complexity is of order $\mathcal{O}(3N \cdot M_g \cdot M_s)$ double precision floating point units in the computer's memory.

As an example, in Griebel and Rüttgers [16] multiscale FENE dumbbell simulations with one segment ($N = 1$) are conducted on a three-dimensional grid with $M_g = 260 \times 64 \times 64$ grid cells. Moreover, the number of discrete samples per grid cells is $M_s = 800$. Therefore, the stochastic field is approximated with about $2.6 \cdot 10^9$ double precision floating points which require 20 *GB* of main memory. Since this complexity scales linearly in the number of spring segments, the same simulation for an $N = 5$ spring segment system would require a main memory of about 100 *GB*.

The large number of stochastic sample particles leads to enormous computing times even on massively parallel computers as observed in Griebel and Rüttgers [16]. For this reason, we propose in the following the combination technique as a viable approach to cope with this complexity.

### 3.2. *Dimension-adaptive combination technique*

The combination technique [4, 20] allows to build approximation spaces of reduced complexity compared to classical full grid approximation spaces. In the following, we introduce the basic concepts of the combination technique as a specific sparse grid representation. A more detailed introduction on sparse grids is given in [5] and in [21].

Let $\Omega \subset \mathbb{R}^d$ be a Lipschitz domain with $d \in \mathbb{N}$ and $\boldsymbol{m} = (m_1, \dots, m_d) \in \mathbb{N}^d$. We then define an anisotropic grid $\Omega_{\boldsymbol{m}}$ on $\Omega$ with $2^{m_i} - 1$ inner grid points in each of the $i = 1, \dots, d$ dimensions. The inner grid consists of the points $x_{\boldsymbol{m},\boldsymbol{j}} = (x_{m_1,j_1}, \dots, x_{m_d,j_d})$ with $j_i = 1, \dots, 2^{m_i} - 1$ for $i = 1, \dots, d$. Here, the multi-index $\boldsymbol{j}$ denotes the position of a grid point in a mesh of resolution $\boldsymbol{m}$. Next, a set of basis functions is defined on $\Omega_{\boldsymbol{m}}$. For this purpose, we employ one-dimensional basis functions $\phi_{m_i,j_i}(x_i)$ in every dimension and build a $d$-dimensional basis by a tensor product approach according to

$$\phi_{\boldsymbol{m},\boldsymbol{j}}(\boldsymbol{x}) = \prod_{i=1}^{d} \phi_{m_i,j_i}(x_i), \quad \boldsymbol{x} \in \Omega. \tag{18}$$

As an example for $\phi_{m_i,j_i}(x_i)$ we may consider the piecewise linear hat functions centered in point $x_i$. Note here that also piecewise polynomials of higher order, interpolets or (pre-)wavelets may be employed, see e.g. Bungartz and Griebel [5]. The full grid approximation space

$$V_{\boldsymbol{m}} = \operatorname{span}\{\phi_{\boldsymbol{m},\boldsymbol{j}} | j_i = 1, \dots, 2^{m_i} - 1,\ i = 1, \dots, d\} \tag{19}$$

is built by using this standard *nodal point basis*. The approximation space (19) can alternatively be defined as direct sum of hierarchical subspaces $W_{\boldsymbol{l}}$. These *hierarchical difference spaces* are defined as

$$W_{\boldsymbol{l}} = \operatorname{span}\{\phi_{\boldsymbol{l},\boldsymbol{j}} | j_i = 1, \dots, 2^{l_i} - 1,\ j_i \text{ odd for all } i = 1, \dots, d\}. \tag{20}$$

The elements of $W_{\boldsymbol{l}}$ for which $\boldsymbol{l} \leq \boldsymbol{m}$ holds component-wise form a basis of $V_{\boldsymbol{m}}$ denoted as *hierarchical basis*. Thus, alternatively to (19), we can then define $V_{\boldsymbol{m}}$ as

$$V_{\boldsymbol{m}} = \bigoplus_{l_1=1}^{m_1} \cdots \bigoplus_{l_d=1}^{m_d} W_{l_1,\dots,l_d} = \bigoplus_{\boldsymbol{l} \leq \boldsymbol{m}} W_{\boldsymbol{l}}. \tag{21}$$

For $m_1 = \cdots = m_d = n$ the full grid approximation space $V_{\boldsymbol{m}}$ has a number of degrees of freedom of the order $\mathcal{O}(N^d)$ with $N = 2^n$. This approach can easily be generalized to function spaces with non-zero boundary conditions and thus grid points on $\partial\Omega$, see e.g. Feuersänger [22].

Sparse grids can be used to decrease the complexity of full grid approximation spaces. A regular sparse grid approximation space of level $n = m_1 = \ldots = m_d$ is defined by

$$V_{\boldsymbol{m}}^{(s)} = \bigoplus_{l_1=1}^{n+d-1} \cdots \bigoplus_{l_d=1}^{n+d-1-\sum_{i<d} l_i} W_{l_1}^{(1)} \otimes \ldots \otimes W_{l_d}^{(d)} = \bigoplus_{|\boldsymbol{l}|_1 \leq n+d-1} W_{\boldsymbol{l}}. \tag{22}$$

Compared to the approximation space $V_{\boldsymbol{m}}$, the number of degrees of freedom reduces from $\mathcal{O}(N^d)$ to $\mathcal{O}(N \cdot \log_2(N)^{d-1})$. This underlines the enormous benefits that are possible with sparse grids. On the other hand, approximation spaces as in (22) require a hierarchical data structure and rely on additional smoothness requirements for the solution of a PDE problem. Typically, this requirement is that the solution is an element of a Sobolev space of mixed smoothness. Sobolev spaces of mixed smoothness are subspaces of Sobolev spaces, the classical solution spaces of PDE problems, see e.g. Bungartz and Griebel [5] for further details.

The combination technique now is a specific sparse grid representation that avoids hierarchical data structures such that existing algorithms on a nodal basis as in Section 3.1 can be reused. For this purpose, the combination technique combines functions $u_{\boldsymbol{l}}$ from different coarse full grid spaces $V_{\boldsymbol{l}}$ with $\boldsymbol{l} \in \mathbb{N}^d$ for

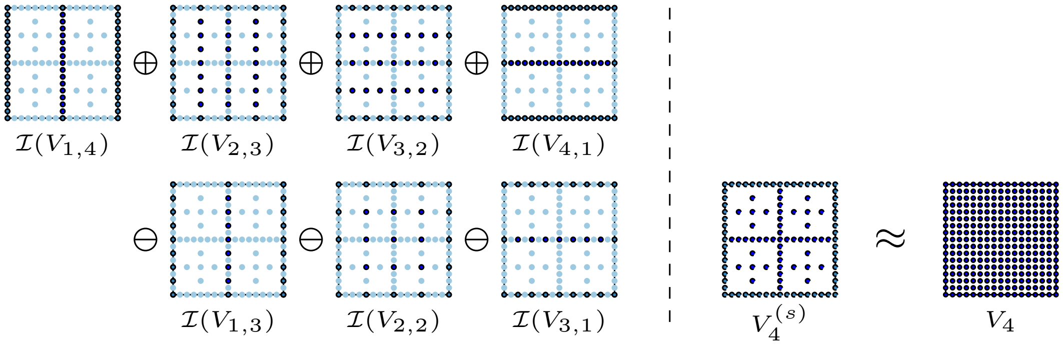


Figure 2: Approximation of the 2D sparse grid space $V_4^{(s)}$ as a linear combination of seven bilinearly interpolated coarse full grid spaces. The coloring was chosen such that interpolated points are colored in light blue.

which $\boldsymbol{l} \leq \boldsymbol{m}$ holds component-wise. The $d$-dimensional combination formula of level $n = m_1 = \cdots = m_d$ is given by

$$u_{\boldsymbol{m}}^c = \sum_{i=0}^{d-1} (-1)^i \binom{d-1}{i} \sum_{|\boldsymbol{l}|_1 = n+(d-1)-i} u_{\boldsymbol{l}} \qquad \text{for } l_1, \ldots, l_d = 1, \ldots, n. \tag{23}$$

As an example, we visualize the two-dimensional combination formula of level $n = 4$ in Fig. 2. Note here that the combination of the different solutions in (23) requires interpolation to a common subspace, i.e. the approximated sparse grid space. Then, the error of the combination technique is of the same order as the sparse grid solution, cf. Griebel and Harbrecht [23]. A further advantage of this representation is that each coarse grid solution can be computed independently such that the combination technique is intrinsically parallel.

Finally, we generalize the sparse approximation space (22) to arbitrary index sets $\mathcal{I} \subset \mathbb{N}^d$. To this end, we define the generalized sparse approximation space as

$$V_{\mathcal{I}} = \bigoplus_{\boldsymbol{l} \in \mathcal{I}} W_{\boldsymbol{l}} = \bigcup_{\boldsymbol{l} \in \mathcal{I}} V_{\boldsymbol{l}}. \tag{24}$$

Then, the generalized combination technique combines all discrete functions $u_{\boldsymbol{l}} \in V_{\boldsymbol{l}}$ with $\boldsymbol{l} \in \mathcal{I}$ according to the formula

$$u_{\mathcal{I}}^c = \sum_{\boldsymbol{l}} c_{\boldsymbol{l}} u_{\boldsymbol{l}} = \sum_{\boldsymbol{l} \in \mathcal{I}} \left( \sum_{\boldsymbol{z}=\boldsymbol{0} \in \mathbb{N}^d}^{(1,\ldots,1)} (-1)^{|\boldsymbol{z}|_1} \cdot \chi^{\mathcal{I}}(\boldsymbol{l} + \boldsymbol{z}) \right) u_{\boldsymbol{l}} \tag{25}$$

where $\chi^{\mathcal{I}}$ is a characteristic function on the index set $\mathcal{I}$ according to

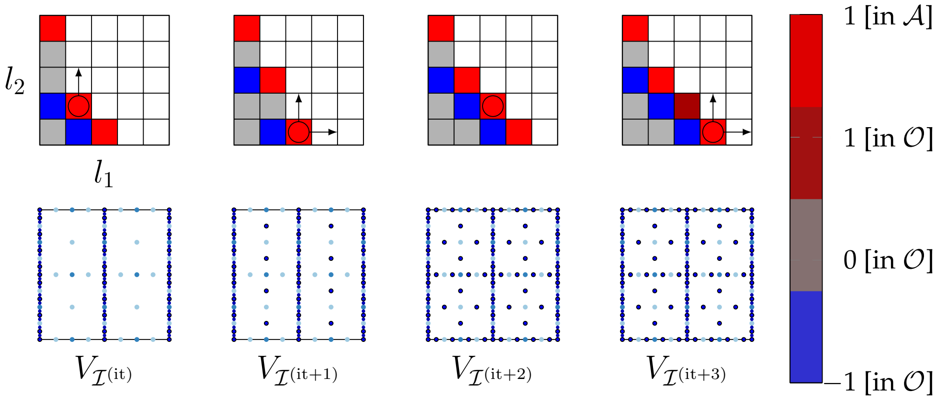
$$\chi^{\mathcal{I}}(\boldsymbol{l}) = \begin{cases} 1 & \text{if } \boldsymbol{l} \in \mathcal{I} \\ 0 & \text{else} \end{cases} \tag{26}$$

and $c_{\boldsymbol{l}} \in \mathbb{R}$ is a combination coefficient. In our application, $u_{\boldsymbol{l}}$ either denotes the discrete velocity field $\boldsymbol{u}$ or the discrete stress tensor field $\boldsymbol{\tau_p}$.

The index set $\mathcal{I}$ contains all grid indices for the combination formula. According to Gerstner and Griebel [24], the index set $\mathcal{I}$ has to fulfill an admissibility condition: If $\boldsymbol{l} \in \mathcal{I}$ with unit vector $\boldsymbol{e}_j$, then it holds

$$\boldsymbol{l} - \boldsymbol{e}_j \in \mathcal{I} \quad \text{for } 1 \leq j \leq d, \quad l_j \geq 1. \tag{27}$$

For an optimal index set $\mathcal{I}$ the error contributions of the different subspaces $V_{\boldsymbol{l}}$ have to be balanced. In many cases, it is not a priori clear how this balancing can be achieved and whether the additional smoothness requirements for the solution are fulfilled. For this purpose, Griebel and Gerstner [25] have developed



$l_2$
$l_1$
$V_{\mathcal{I}^{(\mathrm{it})}}$
$V_{\mathcal{I}^{(\mathrm{it}+1)}}$
$V_{\mathcal{I}^{(\mathrm{it}+2)}}$
$V_{\mathcal{I}^{(\mathrm{it}+3)}}$
1 [in $\mathcal{A}$]
1 [in $\mathcal{O}$]
0 [in $\mathcal{O}$]
−1 [in $\mathcal{O}$]

Figure 3: Evolution of the dimension-adaptive algorithm in two dimensions. The color coding in the first row represents the coefficient $c_{\boldsymbol{l}}$ in (25). The second row illustrates the grid points of the approximated sparse grid space $V_{\mathcal{I}^{(it)}}$.

a dimension-adaptive algorithm that allows to create a close to optimal index set by ongoing refinement $\mathcal{I}^{(1)} = \{(1,\ldots,1) \in \mathbb{N}^d\} \subseteq \mathcal{I}^{(2)} \subseteq \ldots \mathcal{I}^{(it)} \subseteq \ldots$. Furthermore, adaptive algorithms are usually better suited to cope with unsmooth solutions that might occur in our applications. The algorithm ensures the admissibility condition by splitting the index set into two subsets such that $\mathcal{I}^{(it)} = \mathcal{O}^{(it)} \cup \mathcal{A}^{(it)}$ and $\mathcal{O}^{(it)} \cap \mathcal{A}^{(it)} = 0$ in every iteration step *it*. The sets $\mathcal{A}$ and $\mathcal{O}$ are denoted as *active* and *old index set*, respectively. In each iteration step, an element $\boldsymbol{l} \in \mathcal{A}$ is selected that determines adjacent new grids $\boldsymbol{l} + \boldsymbol{e}_j \notin \mathcal{I}$ for $1 \leq j \leq d$ whose associated function $u_{\boldsymbol{l}+\boldsymbol{e}_j}$ might be computed and added to the combination formula if the admissibility condition is fulfilled. Consequently, up $d$ discrete functions are added to the combination formula in every iteration step. Next, the selected $\boldsymbol{l} \in \mathcal{A}$ is removed from $\mathcal{A}$ and added to $\mathcal{O}$. We illustrate the process of refinement with both subsets $\mathcal{A}$ and $\mathcal{O}$ in two dimensions in Fig. 3. Here, the selected index $\boldsymbol{l} \in \mathcal{A}$ is indicated with a circle. Furthermore, arrows indicate the up to $d = 2$ directions of refinement. Since the admissibility condition is violated in both directions from iteration step $it + 2 \rightarrow it + 3$, the sparse grid solution space is not extended in this iteration.

For balancing the different error contributions, we require an appropriate error estimator. To this end, we here do not only consider discretization errors as in the original approach [25], see also Bohn, Garcke and Griebel [26], but also take the modeling accuracy of the spring-chain system due to its number of individual segments into account. Altogether, we have:

- *spatial grid*: the spatial discretization of the fields $\boldsymbol{u}(\boldsymbol{x},t)$, $p(\boldsymbol{x},t)$, $\boldsymbol{\tau}_p(\boldsymbol{x},t)$ and $\boldsymbol{Q}(\boldsymbol{x},t)$ in the coordinate directions $x$, $y$ and $z$ leads to an anisotropic three-dimensional spatial grid.
- *temporal grid*: time integration of the macroscopic flow equations and the microscopic stochastic equation is performed with the same time step width $\Delta t$ and leads to a one-dimensional grid.
- *stochastic grid*: the stochastic partial differential equation (5) is solved for $M_s$ stochastic samples. We consider the number of samples $M_s$ as a one-dimensional problem dimension in the algorithm.
- *spring model grid*: we consider the number of spring segments $N \in \mathbb{N}$ as a modeling dimension.

We illustrate the different dimensions in Fig. 4. Note that the spatial, temporal and stochastic grids affect the accuracy of the corresponding discretization scheme while the spring model grid actually affects the dimensionality of the underlying model equation (5).

Next, we give the details on an appropriate error estimator. Let $\boldsymbol{u}^C_{\mathcal{I}}$ and $\boldsymbol{\tau}^C_{p,\mathcal{I}}$ denote the discrete solutions

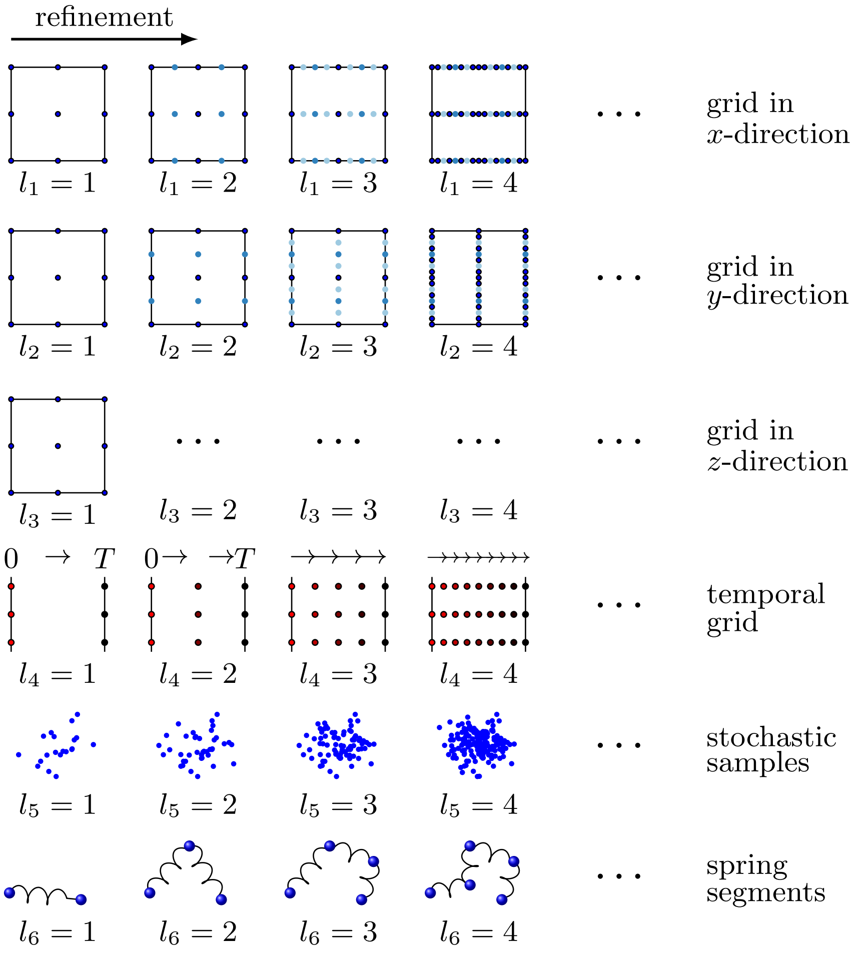


Figure 4: Different problem dimensions for the multiscale polymeric system.

from the combination formula (25). The adaptive scheme builds $\mathcal{I}$ such that

$$\begin{cases} \boldsymbol{u}^C_{\mathcal{I}} & \to \boldsymbol{u} \\ \boldsymbol{\tau}^C_{p,\mathcal{I}} & \to \boldsymbol{\tau}_p \end{cases} \quad \text{for cardinality}(\mathcal{I}) \to \infty. \tag{28}$$

Therefore, we actually require an error estimator for $\boldsymbol{u}$ and $\boldsymbol{\tau_p}$ and a parameter to weight both error terms depending on the specific application. In both cases, the size of the hierarchical surpluses $\|W_{\boldsymbol{l}}(\boldsymbol{u})\|$ and $\|W_{\boldsymbol{l}}(\boldsymbol{\tau_p})\|$ is used as an error indicator. Here, the hierarchical surplus for $\boldsymbol{u_l} \in V_{\boldsymbol{l}}$ is defined as

$$W_{\boldsymbol{l}}(\boldsymbol{u}) \;=\; \otimes_{j=1}^{d}\Delta_j\,\boldsymbol{u_l} \;=\; \Delta_d(\otimes_{j=1}^{d-1}\Delta_j)\,\boldsymbol{u_l} \;=\; (\Delta_d\ldots\Delta_1)\,\boldsymbol{u_l} \tag{29}$$

with the first-order difference operator

$$\Delta_j \boldsymbol{u_l} = \begin{cases} \boldsymbol{u_l} - \boldsymbol{u}_{\boldsymbol{l}-\boldsymbol{e}_j}, & \text{if } l_j > 1, \\ \boldsymbol{u_l}, & \text{if } l_j = 1, \end{cases} \tag{30}$$

and with $\boldsymbol{e}_j$ as canonical unit vector. Again, the first-order difference operator requires interpolation of the functions to a common approximation space. As an illustrative example, we consider (29) for $d = 2$ and

$l_j > 1$ for $j = 1, 2$. In this case, we obtain

$$W_{(l_1,l_2)}(\boldsymbol{u}) = \Delta_2(\Delta_1 \boldsymbol{u}_{(l_1,l_2)}) = \Delta_2(\boldsymbol{u}_{(l_1,l_2)} - \boldsymbol{u}_{(l_1-1,l_2)}) = \boldsymbol{u}_{(l_1,l_2)} - \boldsymbol{u}_{(l_1-1,l_2)} - \boldsymbol{u}_{(l_1,l_2-1)} + \boldsymbol{u}_{(l_1-1,l_2-1)}. \tag{31}$$

Analogously, we calculate the surplus for $\boldsymbol{\tau_p}$. In both cases, we require adequate error norms. The size of the velocity field $\boldsymbol{u}(\boldsymbol{x}, t)$ at fixed position in space-time $(\boldsymbol{x}, t)$ is measured by the Euclidean norm $\|\boldsymbol{u}(\boldsymbol{x}, t)\|_E = \left(\boldsymbol{u}(\boldsymbol{x}, t)^T \cdot \boldsymbol{u}(\boldsymbol{x}, t)\right)^{1/2}$. This is extended to the product space $\mathcal{S} = \mathcal{O} \times \mathcal{T}$ according to

$$\|\boldsymbol{u}\|_{E,2} = \left(\int_{\mathcal{S}} \|\boldsymbol{u}\|_E^2 \, \mathrm{d}\mu\right)^{1/2} \approx \left(\sum_{(\boldsymbol{x}_k, t^n)} \|\boldsymbol{u}(\boldsymbol{x}_k, t^n)\|_E^2 \, \Delta t^n \, \Delta \boldsymbol{x}_k\right)^{1/2} \tag{32}$$

with $\mu$ as product measure on $\mathcal{S}$. The right-hand side of (32) approximates the $L^2$-norm on a discrete grid that results from our algorithm in Section 3.1.2. For the stress tensor $\boldsymbol{\tau_p}$, we also employ definition (32) but use the Frobenius tensor norm $\|\boldsymbol{\tau_p}(\boldsymbol{x}, t)\|_F = (\mathrm{trace}\left(\boldsymbol{\tau_p}(\boldsymbol{x}, t)\boldsymbol{\tau_p}(\boldsymbol{x}, t)^T\right))^{1/2}$ instead of the Euclidean norm in the integrand. This gives the norm $\|\boldsymbol{\tau_p}\|_{F,2}$.

We opt for a cost-benefit optimized refinement criterion. For this purpose, we define the cost $n_{\boldsymbol{l}}$ that is associated with the subspace $V_{\boldsymbol{l}}$ as

$$n_{\boldsymbol{l}} = \lceil T/\Delta t \rceil M_g M_s N. \tag{33}$$

Here, $\lceil T/\Delta t \rceil$ denotes the number of time steps, $M_g$ is the number of grid cells, $M_s$ is the number of stochastic samples and $N$ is the number of spring segments. This definition of the cost differs from the number of grid point operations only by a constant factor of proportionality that is independent from $V_{\boldsymbol{l}}$ and does not affect the refinement procedure.

Using the previous definitions (29)–(33) we define a local refinement indicator as

$$g_{\boldsymbol{l}} = \max\left\{\frac{\omega \cdot \|W_{\boldsymbol{l}}(\boldsymbol{u})\|_{E,2}}{n_{\boldsymbol{l}} \, \|W_{(1,\ldots,1)}(\boldsymbol{u})\|_{E,2}}, \frac{(1-\omega) \cdot \|W_{\boldsymbol{l}}(\boldsymbol{\tau_p})\|_{F,2}}{n_{\boldsymbol{l}} \, \|W_{(1,\ldots,1)}(\boldsymbol{\tau_p})\|_{F,2}}\right\} \tag{34}$$

with $\omega \in [0, 1]$. Definition (34) states an $\omega$-balanced profit indicator in which the benefit that is approximated by the size of the hierarchical surplus is divided by the cost $n_{\boldsymbol{l}}$ for every index $\boldsymbol{l} \in \mathcal{I}$. Furthermore, both hierarchical surpluses are normalized by the size of the surpluses on the coarsest grid $(1, \ldots, 1) \in \mathbb{N}^d$ to make both quantities comparable such that they can be combined via $\omega$ in a balanced way.

The dimension-adaptive combination technique is summarized in Algorithm 1. The iteration steps to build up the index set by ongoing refinement are now cached by the while-loop. Note again that the combination technique, in contrast to classical sparse grids, combines full grid solutions spaces so that hierarchical data structures are not required. More details on the algorithm can be found in Griebel and Gerstner [25] and in Rüttgers [27].

## 4. Results and discussion

In the following, we demonstrate the reductions in complexity using our sparse grid algorithm instead of classical full grid discretizations. For this purpose, we consider two important flow types which are

- shear-type flows/ Couette flows in Section 4.1 and
- extensional-type flows in Section 4.2.

In these applications, we adaptively build index sets to approximate individual components of $\boldsymbol{u}$ and $\boldsymbol{\tau}_p$. Thus, the norms $\|\cdot\|_{E,2}$ and $\|\cdot\|_{F,2}$ from (32) reduce to the usual $L^2$-norm $\|\cdot\|_2$ for scalar fields.

**Algorithm 1:** Dimension-adaptive algorithm for multiscale simulation of polymeric fluids.

```
Determine important $d$ problem dimensions ;                    /* compare with Fig. 4 */
Restrict grid maximum refinement $\boldsymbol{l}_{\max}$ ;      /* e.g. restrictions on spring model grid */
$\boldsymbol{l} := (1, \ldots, 1)$ ;          /* coarsest full grid solution that resolves flow structure */
$A := \{\boldsymbol{l}\}$ ;                                       /* active index set */
$O := \emptyset$ ;                                                /* old index set */
Compute $g_{\boldsymbol{l}}$ as defined in (34) ;               /* 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 (34);
    $O = O \cup \{\boldsymbol{l}\}, \quad A = A \backslash \{\boldsymbol{l}\}$;
    for $j \leftarrow 1$ to $d$ do
        $\boldsymbol{i} = \boldsymbol{l} + \boldsymbol{e}_j$;
        if $\boldsymbol{i} - \boldsymbol{e}_l \in O \quad \forall\, l = 1, \ldots, d$ and if;
        $\boldsymbol{i} \leq \boldsymbol{l}_{max}$ then
            $A = A \cup \{\boldsymbol{i}\}$;
            Apply multiscale flow solver from Section 3.1.2 on full grid space $V_{\boldsymbol{i}}$;
            Compute local profit indicator $g_{\boldsymbol{i}}$ as defined in (34);
        end
    end
    Compute new global error indicator $E$ ;   /* terminate algorithm if $r$-times below TOL */
end
Compute $(\boldsymbol{u}^C_{\mathcal{I}}, \boldsymbol{\tau}^C_{p,\mathcal{I}})$ on index set $\mathcal{I} = O \cup A$ according to (25);
```

*4.1. Couette flows*

Let $\boldsymbol{x} = (x, y, z)$ denote the components of the position vector in the flow space $\mathcal{O}$. We consider a Couette flow in $x$-direction between two parallel plates in a vertical distance $L$ in $y$-direction. At $t = 0$ the fluid is at rest. Then, at $t > 0$ the upper plate at $y = L$ moves with a constant velocity $V_{\text{con}}$. We are interested in the temporal evolution of the $x$-velocity component $u(y, t)$ of $\boldsymbol{u} = (u, v, w)$ for $t \in [0, 1]$, $V_{\text{con}} = L = 1.0$. Furthermore, we employ a Hookean dumbbell model to model viscoelastic flow behavior. In this case, the multiscale system of equations (1)–(11) simplifies to

$$\frac{\partial u(y,t)}{\partial t} = \frac{1}{Re}\beta\frac{\partial^2 u(y,t)}{\partial y^2} + \frac{1}{Re}\frac{\partial \tau_{xy}(y,t)}{\partial y} \tag{35}$$

$$\mathrm{d}\boldsymbol{Q}(y,t) = \left[(\nabla \boldsymbol{u}(y,t))^T \boldsymbol{Q}(y,t) - \frac{1}{2\,De}\boldsymbol{Q}(y,t)\right]\mathrm{d}t + \frac{1}{\sqrt{De}}\mathrm{d}\boldsymbol{W}(t) \tag{36}$$

$$\tau_{xy}(y,t) = \frac{1-\beta}{De}\left\langle Q_x(y,t)\, Q_y(y,t)\right\rangle \tag{37}$$

with $\boldsymbol{Q}(y,t) = (Q_x, Q_y, Q_z)(y,t) \in D \subset \mathbb{R}^3$ and $\boldsymbol{\tau_p} = (\tau_{xx}, \tau_{xy}, \ldots, \tau_{zz}) \in \mathbb{R}^{3\times 3}$.

Moreover, the initial and boundary conditions are

$$\boldsymbol{u}(y,0) = 0, \quad \boldsymbol{u}(0,t) = 0, \quad \boldsymbol{u}(1,t) = V_{\text{con}} \tag{38a}$$

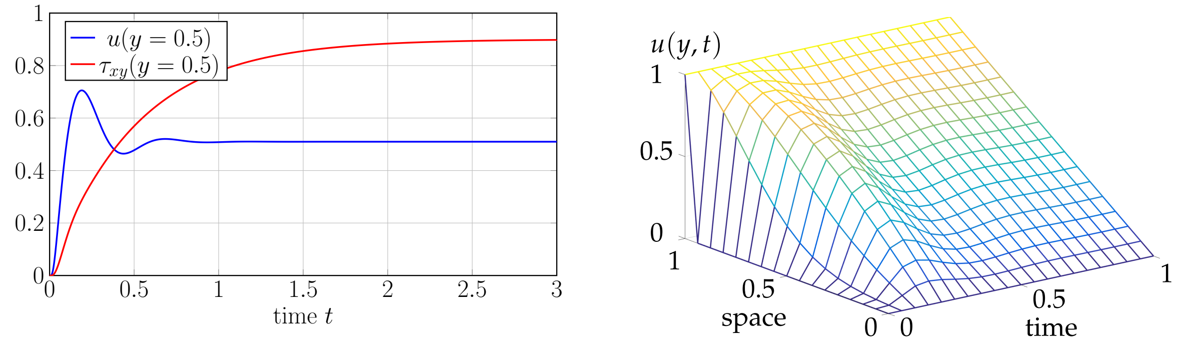
$$\tau_{xy}(y,0) = 0, \quad \partial\tau_{xy}(0,t)/\partial y = \partial\tau_{xy}(L,t)/\partial y = 0 \tag{38b}$$

$$\boldsymbol{Q}(y,0) \sim \psi_{eq}(\boldsymbol{q}) = \frac{1}{(2\pi)^{3/2}}\exp\left(-\frac{\|\boldsymbol{q}\|^2}{2}\right) \tag{38c}$$

for all $t \in [0,1]$ and for all $y \in ]0, L[$. Furthermore, we set $Re = 0.1$, $De = 0.5$ and $\beta = 0.1$. In Fig. 5 (a) we visualize the velocity component $u(y = 0.5, t)$ and the shear stress component $\tau_{xy}(y = 0.5, t)$ in the channel center over time. Furthermore, Fig. 5 (b) plots $u(y,t)$ over space and time. Due to the non-Newtonian fluid behavior, the velocity field first shows some under- and overshoots which are damped out over time.

In the following, we focus on an optimal approximation of the velocity field. For this purpose, we set $\omega = 1$ in (34). First in Table 1, we list a sequence of multi-indices from which $\mathcal{I}$ is built and state the resolutions of the corresponding approximation spaces. The three-dimensional index set is derived from a one-dimensional grid in space, a one-dimensional grid for the stochastic dimension and a one-dimensional grid in time, cf. Fig. 4. In each dimension, the grid is refined by a factor of two such that the cost increases uniformly in each dimension. These rates of refinement do not coincide with the rates of convergence for the corresponding numerical schemes which are one for the spatial grid (central differences with first-order upwind scheme at the boundary), one for the temporal grid (explicit Euler-Maruyama scheme) and 1/2 for the variance reduction of the polymeric stress tensor (Monte Carlo approach). The underlying idea of a refinement factor of two in Table 1 is that the dimensions and the actual amount of refinement shall be determined by the algorithm itself by using the error indicator. Using a larger refinement factor, e.g. a factor of four in some dimensions, would just decrease the accuracy of the dimension-adaptive refinement procedure since an overrefinement in some dimensions becomes more probable.

As mentioned before, the refinement algorithm selects in step $it$ the index $\boldsymbol{l} \in \mathcal{I}^{(it)}$ for which $u_{\boldsymbol{l}} \in V_{\mathcal{I}^{(it)}}$ maximizes $g_{\boldsymbol{l}}$ according to (34). For a better illustration of the refinement directions, Fig. 6 visualizes the profit indicator $g_{\boldsymbol{l}} = \|W_{l_1,l_2,l_3}(u)/n_{l_1,l_2,l_3}\|_2$ for the two-dimensional cuts $g_{(l_1,l_2,l_3=3)}$ and $g_{(l_1,l_2,l_3=5)}$. Note that the normalization factor $\|W_{1,1,1}(u)\|_2$ from (34) is not necessary in this case since only the velocity field is optimized. The isolines of the profit indicator lead to a slightly stronger refinement in the stochastic dimension than in the spatial dimension. At first view, a stronger refinement of the stochastic grid would have been expected due to the lower order of convergence of the Monte Carlo approach for the stress tensor. However, the directions of grid refinement are influenced by two further effects: First, the profit indicator also considers the cost of an approximation space. Therefore, due to an isotropic cost increase in all three dimensions, the profit indicator tends to a more isotropic refinement of all dimension than expected. Second, the velocity field does not directly depend on the stress tensor but only on its divergence. Consequently, an error decrease in the stress tensor that results from a fine stochastic grid does not necessarily



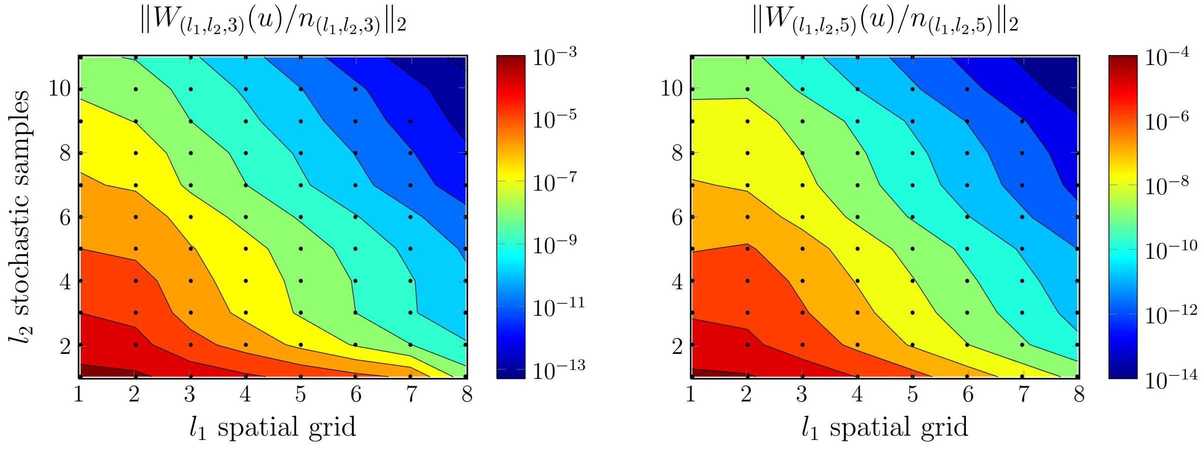
(a) $u$ and $\tau_{xy}$ in the channel center

(b) $u$ over space and time

Figure 5: Non-Newtonian velocity and stress components $u$ and $\tau_{xy}$ in the channel center (left) and $u$ over space and time (right).

Table 1: Sequence of different full grid resolutions $(l_1, l_2, l_3)$ for the simulated Couette flow. The first index $l_1$ denotes the number of finite difference grid cells in the spatial grid, the second index $l_2$ lists the number of stochastic samples and the last index specifies the number of time steps in the time discretization scheme of the approximation space $V_{(l_1,l_2,l_3)}$.

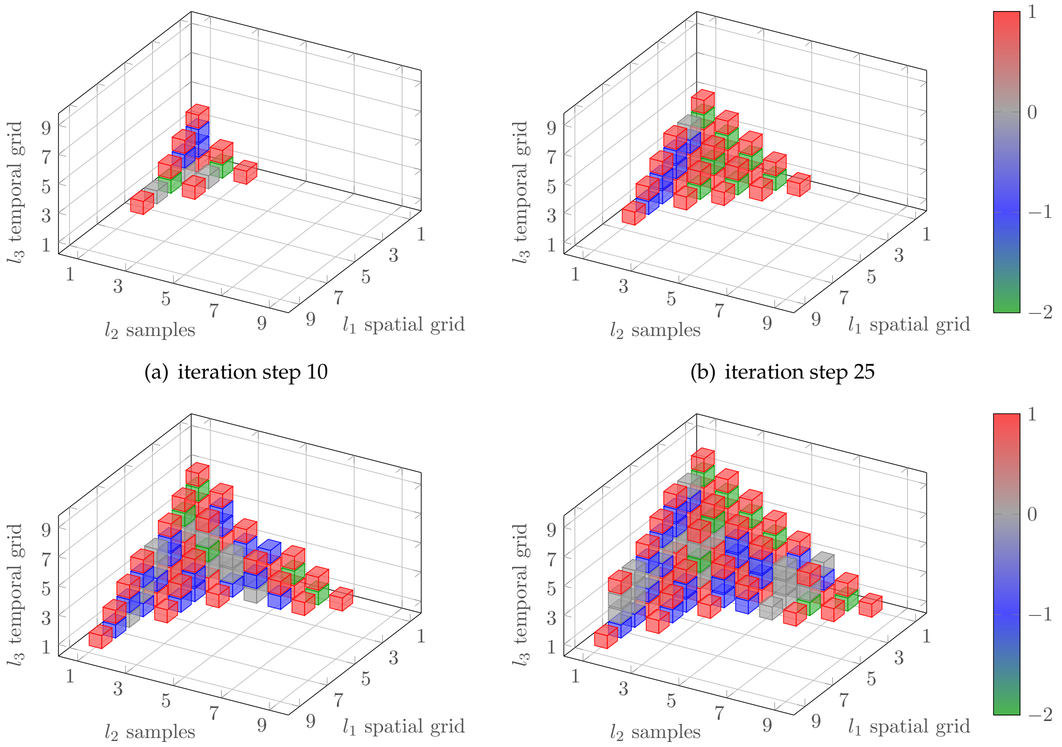
| level $\boldsymbol{l}$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... | 11 |
|---|---|---|---|---|---|---|---|---|---|---|
| space $1/\Delta x_{l_1}$ | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 | | |
| samples $l_2$ | 256 | 512 | $2^{10}$ | $2^{11}$ | $2^{12}$ | $2^{13}$ | $2^{14}$ | $2^{15}$ | ... | $2^{18}$ |
| $1/\Delta t_{l_3}$ | 16 | 32 | 64 | 128 | 256 | 512 | 1024 | 2048 | | |



(a) profit indicators with $l_3 = 3$

(b) profit indicators with $l_3 = 5$

Figure 6: Visualization of a 2D cut of the profit indicator with the index pairs $(l_1, l_2, 3)$ (left) and $(l_1, l_2, 5)$ (right). Note that the first index $l_1$ specifies the number of finite difference cells in the spatial grid and the second index $l_2$ denotes the number of stochastic realizations.



(a) iteration step 10

(b) iteration step 25

(c) iteration step 55

(d) iteration step 90

Figure 7: Index set $\mathcal{I}^{(it)}$ for the iteration steps $it = 10, 25, 55, 90$ of the dimension-adaptive algorithm.

decrease the error in $u$ by the same amount. This explains an asymptotically isotropic refinement of the index set as indicated in Fig. 6.

In Fig. 7 we illustrate the resulting index sets $\mathcal{I}^{(it)}$ for the iteration steps $it = 10, 25, 55, 90$ of the

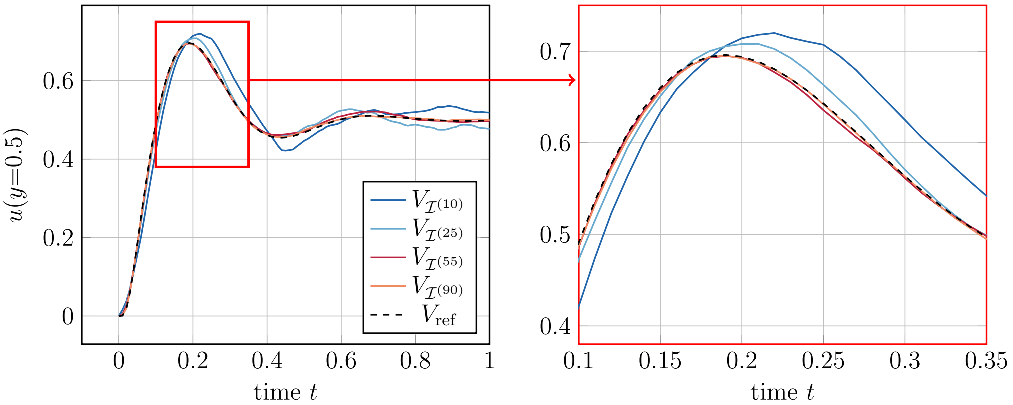


Figure 8: Velocity field in the channel center $y = 0.5$ for different approximation spaces $V_{\mathcal{I}(it)}$ of the iteration steps $it = 10, 25, 55, 90$ in the dimension-adaptive algorithm.

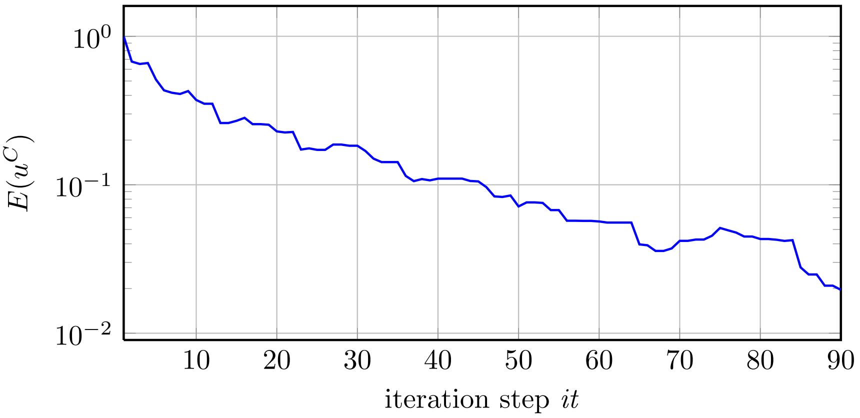


Figure 9: Relative $L^2$-error $E(u^C)$ with respect to the initial approximation $u_{1,1,1} \in V_{1,1,1}$ for each iteration step $it$.

dimension-adaptive algorithm. The color coding of each box in Fig. 7 represents its weight coefficient $c_{\mathbf{l}} \in \{-2, -1, 0, 1\}$ in the combination formula (25). In iteration step $it = 90$, a global error threshold of 0.02 for the relative $L^2$-error $E(u^C) = \|u^C - u_{\text{ref}}\|_2 / \|u^{(1,1,1)} - u_{\text{ref}}\|_2$ has been reached and the algorithm terminates. The relative error is defined with respect to the coarse grid approximation space $V_{1,1,1}$ from Table 1. Again, we note that the profit indicator (34) leads to a more isotropic refinement of each problem dimension than a plain error indicator. If only the hierarchical surplus was considered, the refinement would better correspond with the asymptotic rates of convergence. This is due to the fact that the size of the hierarchical surplus, as an approximation to the error contribution of the corresponding approximation space, should asymptotically decrease with a similar rate as the measured relative error $E(u^C)$.

The approximation spaces $V_{\mathcal{I}(it)}$ shown in Fig. 7 converge to the reference solution with ongoing refinement. We illustrate this convergence in Fig. 8. Here, the velocity field in the channel center $y = 0.5$ is compared for different iteration steps $it = 10, 25, 55, 90$ with a reference solution on a fine grid. Note that the reference solution of the velocity field is also visualized in Fig. 5 (a). On the right-hand side of Fig. 8 we show a magnification around the first maximum of $u$ to better illustrate the higher accuracy of the refined approximation spaces.

For a better analysis of the error decrease, Fig. 9 plots the relative error $E(u^C) = \|u^C - u_{\text{ref}}\|_2 / \|u^{(1,1,1)} - u_{\text{ref}}\|_2$ over all iteration steps $it$. Furthermore, we note that the absolute $L^2$-error of the coarsest full grid solution space is $\|u^{(1,1,1)} - u_{\text{ref}}\|_2 \approx 0.107$. Besides, Fig. 9 shows that the error does not decrease mono-

Table 2: Comparison of cost and accuracy of the full grid spaces $V_{6,6,6}$ and $V_{7,7,7}$ with the sparse grid approximation space $V_{\mathcal{I}^{(90)}}$.

| discrete approach | total cost / $n_{1,1,1}$ | nr. of grids | rel. error $E(u)$ |
|---|---|---|---|
| full grid space $V_{6,6,6}$ | 32,768 | 1 | $\approx 0.04$ |
| full grid space $V_{7,7,7}$ | 262,144 | 1 | $\approx 0.01$ |
| sparse grid space $V_{\mathcal{I}^{(90)}}$ | 5608 | 67 | 0.02 |

Table 3: Absolute and relative $L^2$-error for flows with modified Deborah numbers. The dimension-adaptive approach always uses the index set $\mathcal{I}^{(it)}$ that has been obtained for $De = 0.5$.

| | $De = 0.5$ | $De = 1.0$ | $De = 2.0$ | $De = 4.0$ |
|---|---|---|---|---|
| relative error | $\|u^{C,it} - u_{\text{ref}}\|_2 / \|u^{C,1} - u_{\text{ref}}\|_2$ | | | |
| iteration step $it = 1$ | 1 | 1 | 1 | 1 |
| iteration step $it = 10$ | 0.373 | 0.344 | 0.349 | 0.379 |
| iteration step $it = 25$ | 0.172 | 0.162 | 0.174 | 0.195 |
| iteration step $it = 55$ | 0.068 | 0.063 | 0.067 | 0.075 |
| iteration step $it = 90$ | 0.02 | 0.023 | 0.018 | 0.012 |
| absolute error | $\|u^{C,1} - u_{\text{ref}}\|_2$ | | | |
| iteration step $it = 1$ | 0.107 | 0.097 | 0.086 | 0.075 |

tonically and that the approximation spaces between two neighboring iterations can be identical, see for instance steps $it = 62$ and $it = 63$ in Fig. 9.

In the following, we compare the computational complexity and the accuracy of our sparse grid approach with different full grid solution spaces. For this purpose, Table 2 lists the full grid spaces $V_{6,6,6}$ and $V_{7,7,7}$ that have a uniform refinement according to Table 1 and the sparse grid space $V_{\mathcal{I}^{(90)}}$ after the algorithm has been terminated. The second column plots the relative cost of $V_{6,6,6}$, $V_{7,7,7}$ and $V_{\mathcal{I}^{(it)}}$ with respect to $V_{1,1,1}$. As an example, the relative cost of $V_{6,6,6}$ is $n_{6,6,6}/n_{1,1,1} = 2^{15} = 32768$ since this approximation space is 5-times refined in every problem dimension. Next, the third column of Table 2 lists the number of sub-grid spaces $V_l$ that build $V_{\mathcal{I}^{(90)}}$. We note that this number only contains those grids that have a non-zero weight in the combination formula. Consequently, we exclude the gray colored grids in Fig. 7 (d) with zero weight from the cost consideration. We will justify this approach subsequently. The last column in Table 2 specifies the relative error of all approximations. It can be seen that the sparse grid solution space $V_{\mathcal{I}^{(90)}}$ has an accuracy which is between that of $V_{6,6,6}$ and $V_{7,7,7}$ but it has a computational complexity that is roughly one order of magnitude lower than these full grid spaces. This demonstrates the enormous benefits from using the combination technique.

Finally, we discuss the computational effort of the dimension-adaptive algorithm itself. First, we note that for the considered examples, the computational effort is primarily determined by the cost for approximating the sub-grid spaces $V_l$, cf. line 16 of Algorithm 1, so that we focus on this cost contribution. This simplification is justified by the fact that the flow space $\mathcal{O}$ is one-dimensional such that interpolation on a common grid, required for calculating the profit indicator in line 17 and for evaluating the combination formula in line 22 of Algorithm 1, requires little effort. Obviously, this simplified cost consideration is not valid for two- or three-dimensional flow spaces $\mathcal{O}$ in which interpolation becomes more cost intensive.

Table 2 specifies the cost estimation of the subgrid spaces in $\mathcal{I}^{(90)}$ with non-zero weight in (25). This cost consideration requires that the index set has already been determined. In the first run of Algorithm 1, however, all sub-grid solutions in $\mathcal{I}^{(90)}$ have to be computed to be able to build the hierarchical surplus / profit indicator specified in (34). In that case, 111 sub-grid solutions have to be determined instead of 67 that are finally used in the combination formula. If all 111 grids were taken into account for the cost consideration in Table 2, the total cost with respect to the coarsest approximation space would have

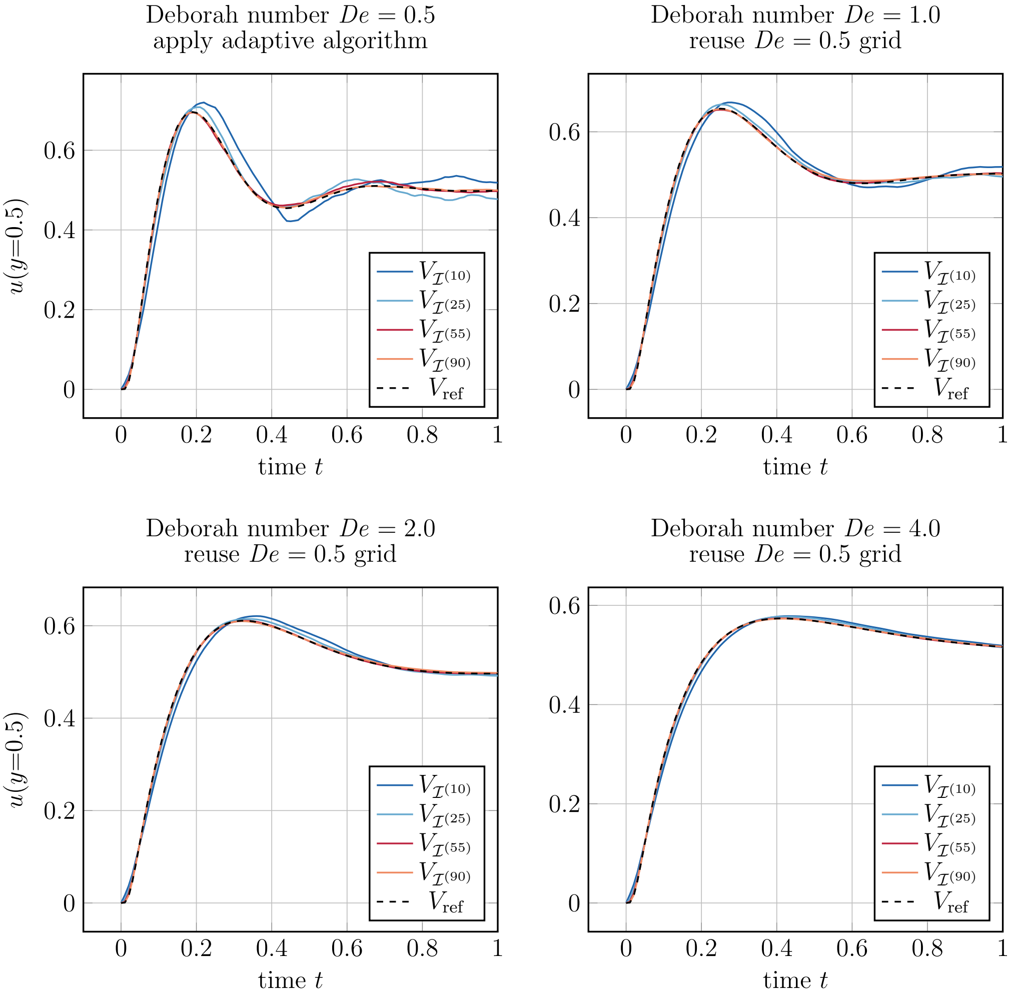


Figure 10: Couette flow simulations with modified Deborah numbers. All simulations use the index sets $\mathcal{I}^{(it)}$ of the iteration steps $it = 10, 25, 55, 90$ that were previously built for $De = 0.5$.

increased from 5608 to 7615. But even then, our algorithm reduces the computational effort by roughly one order of magnitude.

Furthermore, we will subsequently show that the process of creating an optimized index set has to be applied only once for a specific class of problems. As long as the solution continuously depends on the flow parameters, the index set can be reused for modified flow conditions. Since non-Newtonian effects usually occur for highly elastic flows with low Reynolds numbers, the flow field is laminar in most cases. Therefore, this condition holds for most applications in non-Newtonian fluid mechanics. As an example, we consider Couette flows in which the Deborah number has been increased from $De = 0.5$ to $De = 1.0, 2.0$ and 4.0. Fig. 10 shows the velocity field in the channel center at $y = 0.5$ for all four elasticity numbers. Here, the index sets $\mathcal{I}^{(it)}$ for the higher Deborah numbers have not been adaptively built but are reused from the simulation with $De = 0.5$, cf. Fig. 7. We observe that the sequence of velocity over- and undershoots that occur for $De = 0.5$ dampens out for a higher fluid elasticity. This simplifies the numerical approximation of the velocity field for higher Deborah numbers. Consequently, the reused index sets achieve excellent results in all cases. As a final result, Table 3 specifies the absolute and relative errors for the original flow problem and the modified simulations. As observed in Fig. 10, due to the more simple velocity profile, the absolute and relative errors slightly decrease. A further modification of the flow field is considered in

Rüttgers [27]. There, the original index set was reused for higher inflow velocities. In this case, the absolute error increases but the relative error is still unchanged. As a result, the adaptively created index set can be reused for higher inflow velocities as well. We therefore conclude that the dimension-adaptive procedure leads to an index set that is applicable to a wide range of modified Couette flow simulations. For all these modified flow conditions, the cost considerations are exactly as specified in Table 2. Then, the combination technique is able to reduce the computational complexity of multiscale simulations by more than one order of magnitude.

### *4.2. Extensional flows*

In this section, we apply the combination technique to three-dimensional homogeneous extensional flows for which the velocity field is given by $\boldsymbol{u} = (\dot{\epsilon}x, -\frac{\dot{\epsilon}}{2}y, -\frac{\dot{\epsilon}}{2}z)$ for a dimensionless extension rate $\dot{\epsilon} \in \mathbb{R}$. Consequently, since the velocity gradient does not depend on space and time, the multiscale system (1)–(11) for $i = 1, \dots, N$ segments reduces to

$$\mathrm{d}\boldsymbol{Q}_i(t) = \Big[\nabla \boldsymbol{u}^T \boldsymbol{Q}_i(t) - \frac{1}{4\,De(N)} \sum_{k=1}^{N} A_{ik} \cdot \boldsymbol{F}\left(\boldsymbol{Q}_i(t)\right)\Big]\,\mathrm{d}t + \frac{1}{\sqrt{2De(N)}} \Big(\mathrm{d}\boldsymbol{W}_{i+1}(t) - \mathrm{d}\boldsymbol{W}_i(t)\Big) \tag{39}$$

$$\boldsymbol{\tau_p}(t) = \frac{3(1-\beta)\,(b(N)+5)}{b(N)\,De(N)\,((N+1)^2-1)} \sum_{i=1}^{N} \left(\langle \boldsymbol{Q}_i(t) \otimes \boldsymbol{F}_i(\boldsymbol{Q}_i(t))\rangle - \mathbf{Id}\right). \tag{40}$$

We now consider more general spring-chain systems with the nonlinear FENE spring force model (6). The other simulations parameters in (39) and (40) are $De = 1.0$, $\beta = 0.0$, $\dot{\epsilon} = 2.0$ and a maximum full chain extension $b = 120$. We now aim for an accurate approximation of the elastic stress tensor $\boldsymbol{\tau_p}$ and set $\omega = 0$ in (34). In this application, only the diagonal components are non-zero such that $\boldsymbol{\tau_p} = \mathrm{diag}(\tau_{xx}, \tau_{yy}, \tau_{zz})$.

In Table 4 we list three different problem dimensions that will be used for the dimension-adaptive algorithm, cf. Fig. 4. These problem dimensions differ from the dimensions in Section 4.1. Analogously to the previous section, the number of stochastic realizations and the time step width are balanced and refined by a factor of two for each level increase. Moreover, we now also consider the number of spring-chain segments as a further modeling dimension. As mentioned before, this is not a modification of a discretization parameter but a modification of the physical model. To allow for comparisons with a numerical reference solution, we restrict the spring model grid to a maximum number of five spring segments.

A spring model dimension as in Table 4 is only justified if the stress predictions of the modeling equations converge with ongoing refinement. In general this is not the case since the dynamic behavior of a spring-chain depends on the gauge of the modeling equations as in (11) and cannot be fully matched. There are, however, certain situations in which complex spring-chain systems can be reasonably well approximated by more simple chains with less spring segments. As an example, Fig. 11 displays the extensional viscosity $\tau_{xx} - \tau_{yy}/\dot{\epsilon}$ for different spring segment numbers $N$ with the flow parameters as specified before. Here, we note that the results with four and five segments closely resemble each other on a fine grid. In such situations, an optimal sparse-grid-type balancing of discretization errors on the one hand with modeling errors on the other hand can reduce the computational complexity of the full system.

Table 4: Different grids spaces $(l_1, l_2, l_3)$ for an adaptive-refinement of a homogeneous extensional flow. The first index denotes the number of stochastic samples, the second index varies the number of time steps and the last index states the number of spring segments in the FENE chain.

| 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 |
| spring segments $l_3$ | 1 | 2 | 3 | 4 | 5 | | | | |

The result of the dimension-adaptive refinement with a global relative error threshold of 0.02 is shown in Fig 12. As mentioned before, a further refinement for $l_3 > 5$ has been excluded in the dimension-adaptive algorithm due to our choice of a 5-segment reference solution. We note that the refinement is primarily performed in the spring model and in the temporal dimension. This results from the fact that

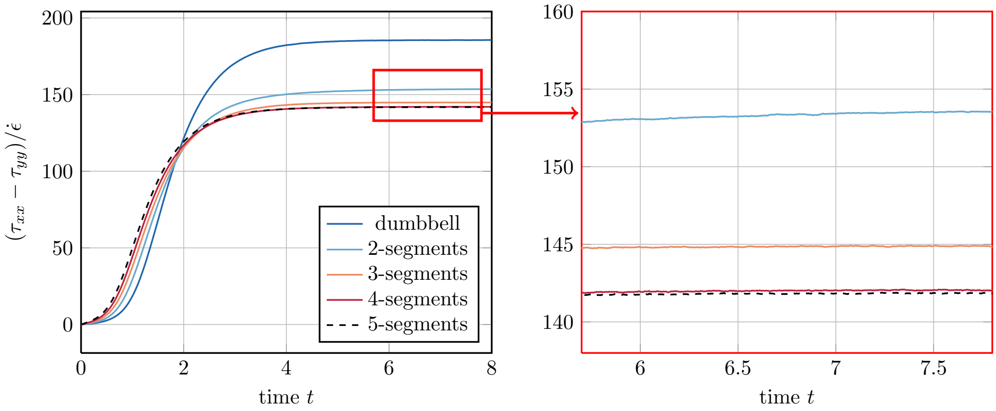
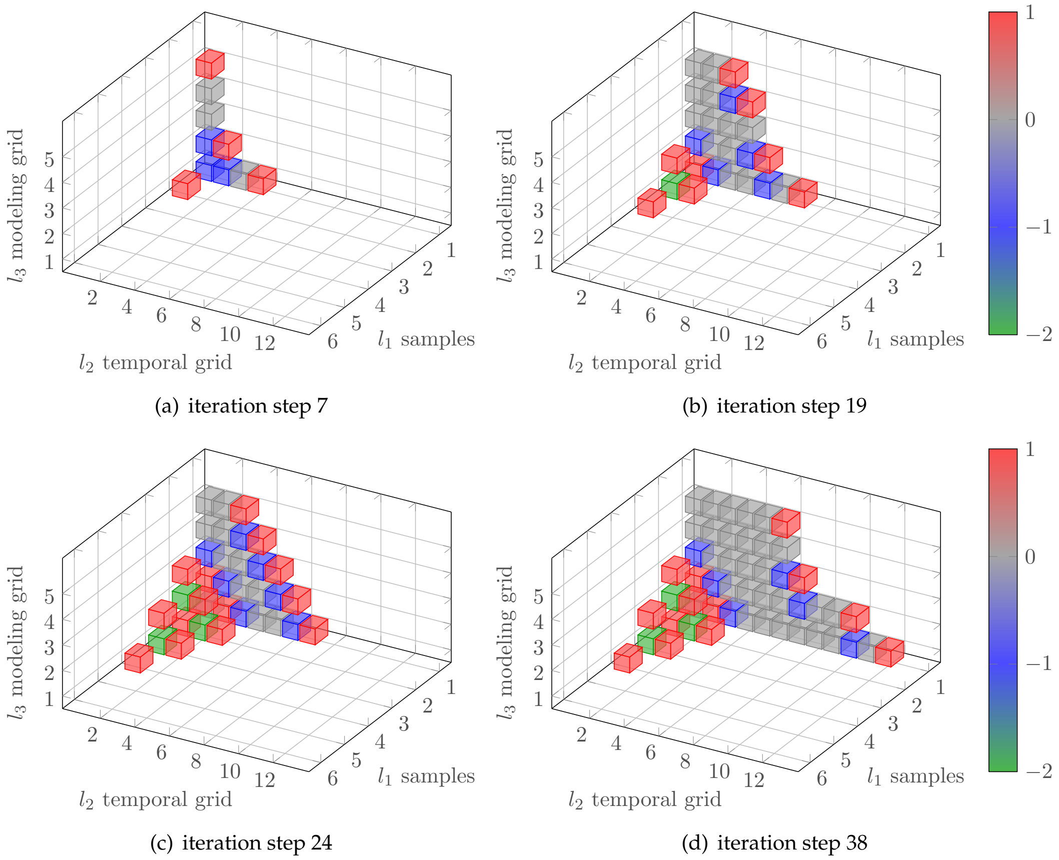


Figure 11: Comparison of the extensional viscosity for different spring-chain systems over time.



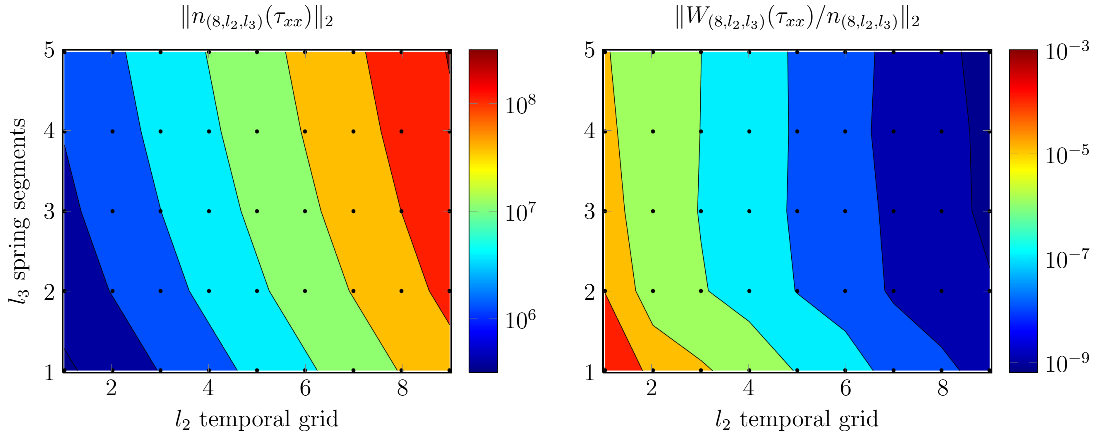
(a) iteration step 7

(b) iteration step 19

(c) iteration step 24

(d) iteration step 38

Figure 12: Visualization of the adaptive index set $\mathcal{I}$ for a homogeneous extensional flow. The triple $(l_1, l_2, l_3)$ denotes the number of stochastic samples, the number of time steps, and the number of spring segments in the discrete approximation spaces, cf. Table 4.



(a) cost of full grid spaces with $l_1 = 8$

(b) profit indicators with $l_1 = 8$

Figure 13: Comparison of the computational complexity (left) and the profit indicator (right) of the full grid spaces $V_{8,l_2,l_3}$, i.e. all involved solution spaces employ $2^{17}$ stochastic samples and only the remain two problem dimensions are varied.

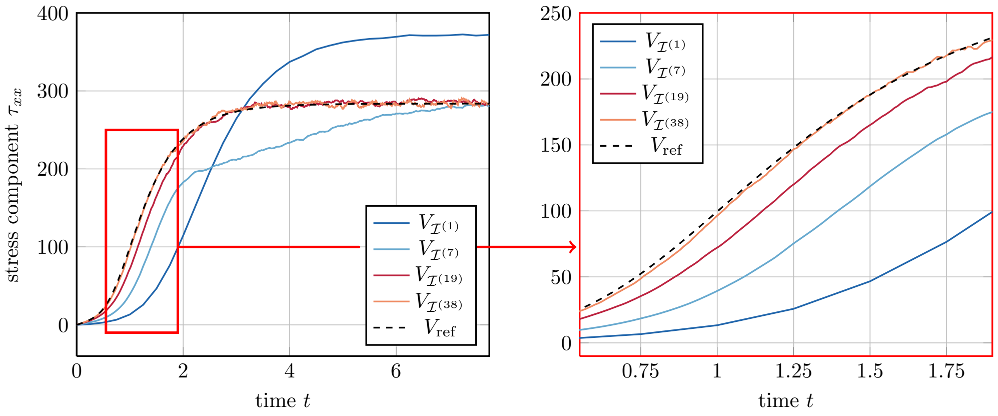


Figure 14: Temporal evolution of the $\tau_{xx}$ stress component for the iteration steps $it = 1, 7, 19, 38$ of the dimension-adaptive algorithm.

the initial grid in the stochastic dimension is relatively fine with 1024 stochastic realizations. This fine initial grid in the stochastic dimension was necessary to increase the numerical stability of the algorithm. Consequently, the temporal error and the spring modeling error are the dominant terms. Moreover, we note that the algorithm fully resolves the spring modeling dimension and therefore leads to a full grid between temporal and modeling dimension. The refinement of the spring model grid primarily results from an anisotropic cost increase. While the number of time steps and stochastic samples doubles between adjacent refinement levels, the cost in the spring model grid increases only linearly. Fig. 13 (a) plots the computational complexity $n_{\boldsymbol{l}}$ of the full grid spaces $V_{\boldsymbol{l}}$ with $l_1 = 8$ and illustrates this anisotropy. Since the profit indicator in (34) also considers the computational complexity, the moderate cost increase for refining the spring model grid leads to a stronger refinement of that dimension, cf. the isolines of the profit grid $W_{(8,l_2,l_3)}/n_{(8,l_2,l_3)}$ in Fig. 13 (b).

Fig. 14 plots the $\tau_{xx}$ stress component for the approximation spaces shown in Fig. 12 in comparison to a reference solution on $V_{12,12,5}$. First, we observe that the approximation space $V_{\mathcal{I}(1)}$ describes just a dumbbell system which of course has a different physical behavior than a system with a 5-segment chain. Then, with ongoing refinement the approximation spaces better correspond to the reference solution. Indeed, the approximation space $V_{\mathcal{I}(38)}$, after which the algorithm has terminated, closely resembles the reference

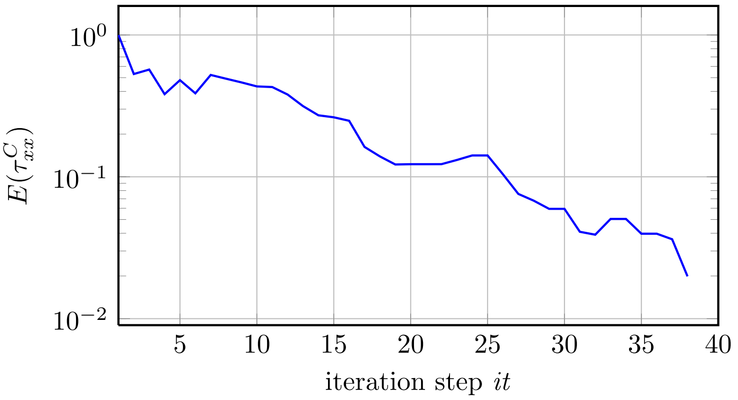


Figure 15: Relative error of $\tau_{xx}$ for different dimension-adaptive refinement steps *it*.

Table 5: Comparison of the cost and accuracy of different sparse and full solution spaces.

| discrete approach | total cost / $n_{1,1,1}$ | nr. of grids | rel. error $E(\tau_{xx})$ |
|---|---|---|---|
| full grid space $V_{8,4,5}$ | 5120 | 1 | 0.098 |
| full grid space $V_{8,5,5}$ | 10,240 | 1 | 0.048 |
| full grid space $V_{8,6,5}$ | 20,480 | 1 | 0.021 |
| full grid space $V_{8,7,5}$ | 40,960 | 1 | 0.009 |
| sparse grid space $V_{\mathcal{I}^{(7)}}$ | 24 | 7 | 0.52 |
| sparse grid space $V_{\mathcal{I}^{(19)}}$ | 215 | 14 | 0.12 |
| sparse grid space $V_{\mathcal{I}^{(38)}}$ | 4235 | 22 | 0.02 |

solution.

We next investigate the convergence of the dimension-adaptive scheme. For this purpose, Fig. 15 displays the relative error of $\tau_{xx}$ with respect to the first iteration in each iteration step *it* according to $E(\tau_{xx}^C) = \|\tau_{xx}^C - \tau_{xx,\mathrm{ref}}\|_2 / \|\tau_{xx}^{(1,1,1)} - \tau_{xx,\mathrm{ref}}\|_2$. Again, we observe that the algorithm converges to the reference solution. Note here that the error does not decrease monotonically. Furthermore, the approximation space between two iteration steps is unaltered if the admissibility condition from Section 3.2 is violated since the index set is not extended then.

Finally, in Table 5 we compare the complexity of different sparse and full grid solution spaces for a specified accuracy. Again, the sparse grid solution spaces lead to considerable reductions in the computational cost. For instance, the spaces $V_{8,6,5}$ and $V_{\mathcal{I}^{(38)}}$ both have a comparable relative error of about 0.02 but the adaptive sparse grid space $V_{\mathcal{I}^{(38)}}$ requires about $1/5$ of grid point operations. This factor of five in the cost reduction is valid for a sequence of simulations, cf. Fig. 10, in which only sub-grid solutions with non-zero weight have to be determined. In the first run for creating the index set, 51 subgrids have to determined to build $V_{\mathcal{I}^{(38)}}$. Since most of these additional sub-grids have a coarse spatial resolution, the cost of the sparse approximation space moderately increases from 4235 to 7149 then. In the end, the cost of the first run to generate $\mathcal{I}^{(38)}$ and the corresponding space $V_{\mathcal{I}^{(38)}}$ is reduced by factor of three compared to $V_{8,6,5}$.

The cost reductions that we achieve in this second example are slightly lower than the factor of ten that was achieved in Section 4.1. We believe that this reduction in efficiency is due to the full grid that is built between the spring model and the temporal grid in the initial stochastic dimension. On the other hand, this result underlines the robustness of the algorithm which reduces to a classical full grid if necessary. In all situations, the algorithm obtains accurate results.

## 5. Conclusions

Multiscale simulations of dilute polymeric fluids were conducted with full and sparse grid numerical approaches. As a sparse grid approach, we employed the combination technique such that our existing full grid solver could be reused. Since a balancing of the different problem dimensions was not a priori known, we employed a dimension-adaptive algorithm that iteratively constructed a sparse approximation space.

For a non-Newtonian Couette flow with a Hookean spring model, our approach was able to reduce the computational complexity of the simulations by one order of magnitude. Furthermore, we also presented numerical results for a multi-segment FENE chain. Here, the computational effort was reduced by a factor of five. Thus, both results clearly show the potential of the dimension-adaptive combination technique. It is planned to employ our sparse grid approach for flow problems with three-dimensional flow spaces, e.g. for three-dimensional contraction flows as in Griebel and Rüttgers [16]. The three-dimensional multiscale simulations necessarily have to be performed in parallel. Since the combination technique determines the partial solutions independently, this adds a second level of parallelism with an ideal scaling behavior that can be made use of. Therefore, we expect even higher cost reductions in the case of three-dimensional flow spaces.

Interestingly, the dimension-adaptive algorithm itself does not lead to a noticeable increase in the computational effort when a sequence of similar flow problems is considered since the constructed index sets can be reused for modified flow conditions. We verified this in Section 4.1 where Couette flows with modified Deborah numbers were simulated. This property is important for applications in the context of *uncertainty quantification* in CFD where a large number of problems is considered with slightly modified flow conditions.

Finally, we note again that our sparse grid approach balances different problem dimensions. More precisely, it considers not only different discretization dimensions but also takes a further spring modeling dimension, the number of spring segments, into account. This is a novel application for the combination technique. We assume that this idea can be generally employed for a range of other problems. A high cost reduction can be expected if the cost increases exponentially with the modeling dimension as, for instance, in molecular dynamics of correlated systems where many-body potentials are considered.

## References

[1] R. G. Owens, T. N. Phillips, Computational Rheology, volume 2, Imperial College Press, London, 2002.

[2] C. Mangoubi, M. Hulsen, R. Kupferman, Numerical stability of the method of Brownian configuration fields, J. Non-Newtonian Fluid Mech. 157 (2009) 188–196.

[3] R. Keunings, Micro-macro methods for the multiscale simulation of viscoelastic flow using molecular models of kinetic theory, in: D. Binding, K. Walters (Eds.), Rheol. Rev., British Society of Rheology, 2004, pp. 67–98.

[4] M. Griebel, M. Schneider, C. Zenger, A combination technique for the solution of sparse grid problems, in: P. de Groen, R. Beauwens (Eds.), Iterative Meth. Lin. Alg., Elsevier, 1992, pp. 263–281.

[5] H. Bungartz, M. Griebel, Sparse grids, Acta Numerica 13 (2004) 147–269.

[6] P. Delaunay, A. Lozinski, R. Owens, Sparse tensor-product Fokker-Planck-based methods for nonlinear bead-spring chain models of dilute polymer solutions, in: CRM Proc. and Lect. Notes, volume 41.

[7] M. Griebel, F. Koster, Multiscale methods for the simulation of turbulent flows, in: E. Hirschel (Ed.), Numer. Flow Simul. III, volume 82 of *Notes on Numer. Fluid Mech. and Multidiscip. Des.*, Springer-Verlag, 2003, pp. 203–214.

[8] F. Chinesta, A. Ammar, A. Leygue, R. Keunings, An overview of the proper generalized decomposition with applications in computational rheology, J. Non-Newtonian Fluid Mech. 166 (2011) 578–592.

[9] M. Laso, H. Öttinger, Calculation of viscoelastic flow using molecular models: the CONNFFESSIT approach, J. Non-Newtonian Fluid Mech. 47 (1993) 1–20.

[10] M. Hulsen, A. P. G. Van Heel, B. Van Den Brule, Simulation of viscoelastic flows using Brownian configuration fields, J. Non-Newtonian Fluid Mech. 70 (1997) 79–101.

[11] H. Warner, Kinetic theory and rheology of dilute suspensions of finitely extendible dumbbells, Ind. Eng. Chem. Fundam. 11 (1972) 379–387.

[12] R. Bird, O. Hassager, R. Armstrong, C. Curtiss, Dynamics of Polymeric Liquids, Volume 2: Kinetic Theory, Wiley New York, 1987.

[13] I. Ghosh, G. McKinley, R. Brown, R. Armstrong, Deficiencies of FENE dumbbell models in describing the rapid stretching of dilute polymer solutions, J. Rheol. 45 (2001) 721–758.

[14] J. Wiest, R. Tanner, Rheology of bead-nonlinear spring chain macromolecules, J. Rheol. 33 (1989) 281–316.

[15] A. Koppol, R. Sureshkumar, B. Khomami, An efficient algorithm for multiscale flow simulation of dilute polymeric solutions using bead-spring chains, J. Non-Newtonian Fluid Mech. 141 (2007) 180–192.

[16] M. Griebel, A. Rüttgers, Multiscale simulations of three-dimensional viscoelastic flows in a square-square contraction, J. Non-Newtonian Fluid Mech. 205 (2014) 41–63.

[17] J. von Neumann, Various techniques used in connection with random digits. Monte Carlo methods, Nat. Bur. Stand. 12 (1951) 36–38.

[18] J. Bell, P. Colella, H. Glaz, A second-order projection method for the incompressible Navier-Stokes equations, J. Comput. Phys. 85 (1989) 257–283.

[19] H. C. Öttinger, Stochastic Processes in Polymeric Fluids: Tools and Examples for Developing Simulation Algorithms, Springer, Berlin, 1996.

[20] H. Bungartz, M. Griebel, U. Rüde, Extrapolation, combination, and sparse grid techniques for elliptic boundary value problems, Comput. Meth. Appl. Mech. Eng. 116 (1994) 243–252.

[21] J. Garcke, Sparse grids in a nutshell, in: Sparse grids and applications, Springer, 2012, pp. 57–80.

[22] C. Feuersänger, Sparse grid methods for higher dimensional approximation, Dissertation, Institut für Numerische Simulation, Universität Bonn, 2010.

[23] M. Griebel, H. Harbrecht, On the convergence of the combination technique, in: Sparse grids and Applications, volume 97 of *Lect. Notes in Comput. Sci. and Eng.*, Springer, 2014, pp. 55–74.

[24] T. Gerstner, M. Griebel, Numerical integration using sparse grids, Numer. Algorithm. 18 (1998) 209–232.

[25] T. Gerstner, M. Griebel, Dimension–adaptive tensor–product quadrature, Comput. 71 (2003) 65–87.

[26] B. Bohn, J. Garcke, M. Griebel, A sparse grid based method for generative dimensionality reduction of high-dimensional data, J. Comput. Phys. 309 (2016) 1–17.

[27] A. Rüttgers, Multiscale simulation of polymeric fluids using sparse grids, Dissertation, Institut für Numerische Simulation, Universität Bonn, 2016.