University of Cologne

Department of Mathematics

Numerical Solution of Integro-Differential equations

Master's thesis

October 15, 2019

in cooperation with the German Aerospace Center (SC-HPC)

Nils Dornbusch
first examiner: Prof. Dr-Ing. Gassner

Danksagung

Als Erstes möchte ich mich bei Herrn Professor Gassner für die sehr gute Betreuung und die Flexibilität bedanken, dass ich in Kooperation mit dem Deutschen Zentrum für Luft- und Raumfahrt meine Masterarbeit anfertigen konnte. Darüber hinaus gilt mein Dank Herrn Dr. Basermann für die Möglichkeit, Betreuung für meine Masterarbeit in der Abteilung High-Performance Computing (SC-HPC) zu erhalten. Genau dieser Abteilung möchte ich vom ganzen Herzen für die tolle kollegiale Zusammenarbeit schon vor Beginn meiner Masterarbeit und währenddessen sowie die vielfältige Unterstützung bedanken. Insbesondere Herrn Dr. Kontak und Herrn Dr. Knechtges gilt meine große Wertschätzung für die viele Zeit, das Engagement und die große fachliche und menschliche Unterstützung in dieser Zeit. Natürlich möchte ich mich in diesem Zuge auch bei Herrn Professor Sperl für die Bereitstellung des Themas bedanken.

Zusätzlich bedanke ich mich noch bei meiner Familie, meinen Freunden und Kommilitonen und meiner Freundin für ihre Geduld und ihr Verständnis.

Köln, 15. Oktober 2019

Nils Dornbusch

Contents

1 Introduction

This work aims to provides an analysis of the integro-differential equations which occur while modeling the behavior of non-NEWTONIAN fluids. The motivation to model these fluids arises when studying for example liquid glass or granular flow. This is often needed in space applications. This work should be seen as a first step in studying these governing equations. The long term goal is to develop a method to simulate an arbitrary non-NEWTONIAN fluid in three spatial dimensions. Such a method does not exist as of now. An approach taken in [14] and [15] is the deformation fields method, which we will follow in part. However, not the whole method is suited for our needs. This is due to the fact that this method would result in the calculation of an integral in each timestep even if the timesteps are chosen carefully. What is even worse is that the more timesteps one calculates the more expensive the integral becomes. As the full system of equations is five-dimensional, one can imagine that this is not efficient for a reasonable number of degrees of freedom. This works aims to provide a possibility to reduce the calculation effort. However, as this is a master's thesis only a reduced three-dimensional system will be considered. To obtain this reduction, sensible assumptions will be taken.

The goal is that every reader that is proficient in Mathematics on a master's level is able to follow the arguments given in this work. To achieve this, some basic mathematical facts will be given in the beginning of this work. We will also introduce some notation that will be used. The second chapter will deal with the Physics behind the governing equation. As this is a mathematical work most of the time we will only discuss ideas rather than discussing the whole physical theory. After that we will reduce the system of equations from three to two spatial dimensions. The elimination of a second time variable will take place in chapter 3, where we will use the LAPLACE transform. Chapter 4 will deal with all aspects of numerics. At first we will discretize the continuous equations from chapter 3 in space and time. After we have built the numerical scheme we will study the convergence and other properties of the algorithm. Some aspects regarding the implementation will also be discussed.

Chapter 5 will mostly take a look at a different integro-differential equation. It can be found in [12]. The equation from [12] is easier in the sense that is does not have a spatial component or multiple variables but it is as complex as the full system regarding the calculation of the integral. Therefore it is much easier to study the simpler equation and try out different ideas, which can then be transfered back to the original problem. This is necessary, because it turns out that we can only deduce a better algorithm under the assumptions we took to reduce the five-dimensional system of equations.

In the last chapter we will conclude all results and give an outlook on what could be

next sensible points to consider.

2 Mathematical basics

In this chapter, various definitions and theorems are presented, which will be used extensively throughout this work. Most of them are just a recap of well-known theorems and are therefore presented without proof. The interested reader may follow the literature referenced in each section.

2.1 Basics and notation

In the beginning, we will quickly state basic mathematical facts and introduce some notation, which we will use throughout this work. Most of this should be self-explanatory.

Notation 1. We will use $\mathbb{N}$ as the natural numbers starting from 1, $\mathbb{N}_0$ represent the natural numbers starting from 0. $\mathbb{R}, \mathbb{C}, \mathbb{Q}$ denote the real, complex and rational numbers. With $\mathbb{K}$ we denote a field, which can be $\mathbb{R}$ or $\mathbb{C}$. $\mathbb{R}_0^+$ shall denote $[0, \infty)$.

Definition 2 (Essential supremum)**.** Let $f\colon X \to \mathbb{R}$ be a function with $X \subset \mathbb{R}^n$ and $n \in \mathbb{N}$. We call

$$\operatorname*{ess\,sup}_{x \in X} f(x) := \inf\{a \in \mathbb{R}\colon f(x) \leq a \quad \text{almost everywhere}\} \tag{2.1}$$

the essential supremum of f.

Notation 3. We use ∇f to denote the spatial gradient of f. For the Laplacian operator we use the standard symbol Δ.

Notation 4 (Multi-index notation)**.** To generalize the idea of an integer index, we will consider $\alpha = (a_1, \ldots, a_n)$, where $a_i \in \mathbb{N}_0$ for all $i = 1, \ldots, n$. We call α a multi-index. We define $|\alpha| = \sum_{i=1}^n a_i$. Later on $\sum_{|\alpha| \leq n}$ will appear for a fixed $n \in \mathbb{N}$. By this we denote the sum over all possible α. We will also encounter D^α, where D is the standard differential operator. We define

$$D^\alpha = \frac{\partial^{|\alpha|}}{\partial x_1^{a_1} \cdots \partial x_n^{a_n}}. \tag{2.2}$$

2.2 Functional analysis

Here, we cover some basics of functional analysis that are necessary for this work and especially the finite element method. We will follow [9] for this.

Definition 5 (Norm)**.** Let X be a $\mathbb{K}$ vector space. A mapping $\|\cdot\| : X \to \mathbb{R}$ is called a *norm* on X if

$$1.\ \|x\| = 0 \Leftrightarrow x = 0 \qquad \forall x \in X \qquad \text{(positive definite)}, \tag{2.3}$$
$$2.\ \|\lambda x\| = |\lambda|\,\|x\| \qquad \forall x \in X, \lambda \in \mathbb{K} \qquad \text{(homogeneous)}, \tag{2.4}$$
$$3.\ \|x + y\| \leq \|x\| + \|y\| \qquad \forall x, y \in X \qquad \text{(triangle inequality)}. \tag{2.5}$$

We call the pair $(X, \|\cdot\|)$ a *normed space.*

Remark 6. It follows directly from the definition that $\|x\| \geq 0$ for all $x \in X$.

Definition 7 (Cauchy sequence)**.** A sequence $(x_n)_{n\in\mathbb{N}}$ is called Cauchy *sequence* if

$$\forall \varepsilon > 0\ \exists n_0(\varepsilon) \in \mathbb{N} \colon \forall m, n > n_0 \colon\ \|x_m - x_n\| < \varepsilon. \tag{2.6}$$

Definition 8 (Convergence)**.** A sequence $(x_n)_{n\in\mathbb{N}}$ converges to $x \in X$ if

$$\forall \varepsilon > 0\ \exists n_0(\varepsilon) \colon \forall n > n_0 \colon\ \|x_n - x\| < \varepsilon. \tag{2.7}$$

It is well-known that a Cauchy sequence does not converge in general. The class of spaces, for which this is true, is given in the following definition.

Definition 9. A normed space $(X, \|\cdot\|)$ is called *complete* if every Cauchy sequence in X converges in X. A complete normed space is also called a Banach *space.*

From now on let Ω be an open and bounded subset of $\mathbb{R}^n$, with $n \in \mathbb{N}$.

Definition 10. For a function $f \colon \Omega \to \mathbb{R}$ and $p \in [1, \infty)$ we define the norm

$$\|f\|_p := \left(\int_\Omega |f|^p \,\mathrm{d}\Omega \right)^{1/p} \tag{2.8}$$

and

$$\|f\|_\infty := \operatorname{ess\,sup}\{|f(x)| : x \in \Omega\}. \tag{2.9}$$

However, this is only a norm if we identify functions, which only differ on sets of zero measure. As usual the sets of equivalence classes of almost everywhere identical functions for which $\|f\|_p < \infty$ is denoted by $\mathrm{L}^p(\Omega)$.

Lemma 11 (Minkowski's inequality)**.** *For $f, g \in L^p(\Omega)$ and $p \in [1, \infty]$, we have*

$$\|f + g\|_p \leq \|f\|_p + \|g\|_p \tag{2.10}$$

Theorem 12. *The space $L^p(\Omega)$, $p \in \mathbb{N} \cup \{\infty\}$ is a* Banach *space.*

Definition 13. Let V be a $\mathbb{K}$ vector space. A mapping $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{R}$ is called an *inner product* or *dot product* if

1. $\langle u, u \rangle \geq 0 \quad \forall u \in V$ (positive definite),

2. $\langle u, u\rangle = 0 \Leftrightarrow u = 0$ (strictly positive),

3. $\langle u, v\rangle = \overline{\langle v, u\rangle} \quad \forall u, v \in V$ (conjugate symmetric),

4. $\langle u, \lambda v\rangle = \lambda \langle u, v\rangle \quad \forall u, v \in V, \lambda \in \mathbb{K}$ (homogeneous in second argument),

5. $\langle u, v + w\rangle = \langle u, v\rangle + \langle u, w\rangle \quad \forall u, v \in V$ (linear in second argument).

Remark 14. In the case that V is a real-valued vector space, the inner product is linear and homogeneous in both arguments. In case V is a complex-valued vector space we obtain linearity in the second argument. Let $\lambda \in \mathbb{K}$ and $u, v, w \in V$, then

$$\langle \lambda u + v, w\rangle = \overline{\langle w, \lambda u + v\rangle} = \overline{\lambda} \langle u, w\rangle + \langle v, w\rangle . \tag{2.11}$$

Theorem 15. *Let V be a vector space and $\langle \cdot, \cdot \rangle$ be an inner product. By setting $\|v\|_V := \sqrt{\langle v, v\rangle}$, we obtain a norm. Thus every vector space with an inner product is a normed space.*

Definition 16. A complete vector space with an inner product is called a HILBERT *space.*

Example 17. $V = \mathrm{L}^2(\Omega)$ with the inner product

$$\langle f, g\rangle := \int_\Omega f(x) g(x) \,\mathrm{d}x \tag{2.12}$$

is a HILBERT space.

Theorem 18 (CAUCHY-SCHWARZ inequality)**.** *Let V be a vector space. Then*

$$|\langle u, v\rangle| \leq \|u\|_V \|v\|_V \quad \forall u, v \in V. \tag{2.13}$$

Notation 19. We write $\mathrm{C}^k(\Omega)$, $k \in \mathbb{N} \cup \{\infty\}$, for the space of k-times differentiable functions on Ω. $\mathrm{C}^0(\Omega)$ shall be the space of continuous functions on Ω.

Definition 20. The *support* of a function $f \colon \Omega \to \mathbb{R}$ is defined by

$$\operatorname{supp} f = \overline{\{x \in \Omega \colon f(x) \neq 0\}}. \tag{2.14}$$

Notation 21. In the following we will often use functions with compact support, so we introduce the notation

$$\mathrm{C}_0^k(\Omega) = \{f \in \mathrm{C}^k(\Omega) \colon \operatorname{supp} f \text{ is compact in } \Omega\}, \tag{2.15}$$

where we mean "compact" in the topological sense, so in our cases bounded and closed.

Remark 22. One can proof that if $f \in \mathrm{C}_0^k$, then f tends to zero on the boundary $\partial\Omega$. This motivates the notation.

Definition 23. We define

$$\mathrm{L}^1_{\mathrm{loc}}(\Omega) := \{f\colon \Omega \to \mathbb{R}\colon\ f \in \mathrm{L}^1(A) \text{ for all compact} A \subset \Omega\}, \tag{2.16}$$

as the space of *locally integrable functions*.

Example 24. The space $\mathrm{L}^1_{\mathrm{loc}}$ is a true superset of L^1. Consider $f(x) = \frac{1}{x}$ on $(0,1)$. f belongs to $\mathrm{L}^1_{\mathrm{loc}}(0,1)$, but not to $\mathrm{L}^1(0,1)$.

Definition 25. Let $\alpha \in \mathbb{N}_0^n$ be a multi index. A function $f \in \mathrm{L}^1_{\mathrm{loc}}(\Omega)$ has a *weak derivative* ν, which is denoted by $D^\alpha f \in \mathrm{L}^1_{\mathrm{loc}}(\Omega)$ if $\forall \varphi \in \mathrm{C}_0^\infty$

$$\int_\Omega \nu\varphi \,\mathrm{d}\Omega = (-1)^{|\alpha|} \int_\Omega f D^\alpha \varphi \,\mathrm{d}\Omega, \tag{2.17}$$

where

$$D^\alpha \varphi = \frac{\partial^{|\alpha|}\varphi}{\partial x_1^{\alpha_1} \cdots \partial x_n^{\alpha_n}}. \tag{2.18}$$

Remark 26. This definition is motivated by the integration by parts method. One wants to find functions that obey integration by parts but are not necessarily differentiable in the classical sense. However, if f is differentiable then the weak and classic derivatives match. It is also important to note that the weak derivative of a function, if it exists, is unique if we identify functions, which only differ on sets of zero measure.

Definition 27. We denote by $W^{m,p}(\Omega)$, for $p \in \mathbb{N} \cup \{\infty\}$ and $m \in \mathbb{N}_0$, the set of all functions $f \in \mathrm{L}^p(\Omega)$ with weak derivatives in $\mathrm{L}^p(\Omega)$ up to the order $|\alpha| \leq m$. These sets are called Sobolev *spaces*. We define $\mathrm{H}^m(\Omega) := W^{m,2}(\Omega)$.

Theorem 28. *The* Sobolev *space $W^{m,p}(\Omega)$ with the norm*

$$\|f\|_{W^{m,p}} := \left(\int_\Omega \sum_{|\alpha| \leq m} |D^\alpha f(x)|^p \,\mathrm{d}\Omega \right)^{1/p}, \tag{2.19}$$

for $p \in \mathbb{N}$, and

$$\|f\|_{W^{m,p}} := \max_{|\alpha| \leq m} \left(\operatorname*{ess\,sup}_{x \in \Omega} |D^\alpha f(x)| \right) \tag{2.20}$$

if $p = \infty$, is a Banach *space. $\mathrm{H}^m(\Omega)$ with the inner product*

$$\langle u, v \rangle_{\mathrm{H}^m} := \sum_{|\alpha| \leq m} \langle D^\alpha u, D^\alpha v \rangle = \sum_{|\alpha| \leq m} \int_\Omega D^\alpha u D^\alpha v \,\mathrm{d}\Omega \quad \forall u, v \in \mathrm{H}^m(\Omega) \tag{2.21}$$

is a Hilbert *space.*

Remark 29. The spaces H^m are very important for many numerical methods because of the property that they are Hilbert spaces. The whole theory, which follows hereafter, relies heavily on this assumption. We will also use H_0^m to denote the closure of C_0^∞ with respect to $\|\cdot\|_{\mathrm{H}^m}$. Further explanation on this process is given in [16].

2.3 The finite element method

This section covers the basics of the finite element method (FEM). We still use [9] as a reference. Let us start by introducing the concept of *weak solutions*. Consider the POISSON problem

$$-\Delta u = f \quad \text{in } \Omega \tag{2.22}$$

and $u = 0$ on the boundary $\partial\Omega$ of Ω, where $u, f \colon \Omega \to \mathbb{R}$. This formulation requires $u \in \mathrm{C}^2(\Omega)$ if $f \in \mathrm{C}^0(\Omega)$ is prescribed. In other examples this often excludes the physical solution, so we want to reduce the required regularity. For this, we integrate both sides over Ω and multiply with a test function $\varphi \in \mathrm{C}_0^\infty(\Omega)$ to obtain

$$-\int_\Omega \Delta u \varphi \,\mathrm{d}\Omega = \int_\Omega f\varphi \,\mathrm{d}\Omega, \quad \forall \varphi \in \mathrm{C}_0^\infty(\Omega). \tag{2.23}$$

If we now apply GREEN's theorem on the left side, we obtain

$$\int_\Omega \nabla u \cdot \nabla \varphi \,\mathrm{d}\Omega = \int_\Omega f\varphi \,\mathrm{d}\Omega \quad \forall \varphi \in \mathrm{C}_0^\infty(\Omega). \tag{2.24}$$

We call this equation the *weak formulation* of (2.22). The boundary term has vanished because φ is 0 on the boundary as we have stated in Remark 22. We can observe that we could use GREEN here because we assume $u \in \mathrm{C}^2(\Omega)$. But (2.24) imposes much fewer regularity requirements than the classical formulation. We only need $u, \varphi \in \mathrm{H}^1(\Omega)$ to satisfy the equation. The space of acceptable solutions is now more extensive than before. Also, every solution of (2.22) automatically satisfies (2.24).

Definition 30. Let V be a HILBERT space. A bilinear form $a \colon V \times V \to \mathbb{R}$ is said to be *continuous* if there exists a constant $\beta > 0$ such that

$$a(u, v) \leq \beta \|u\| \|v\| \quad \forall u, v \in V, \tag{2.25}$$

and *coercive* if

$$\exists \alpha > 0 \colon \quad a(u, u) \geq \alpha \|u\|^2 \quad \forall u \in V. \tag{2.26}$$

Remark 31. We see that if a is the inner product of V and induces the norm, then both properties are satisfied. The first identity is simply the CAUCHY-SCHWARZ inequality and the second is fulfilled since $\langle v, v \rangle = \|v\|^2$. The term $\sqrt{a(v, v)}$ is called *energy norm*, which is the natural norm for error analysis of this problem.

Theorem 32 (LAX-MILGRAM)**.** *Let V be a* HILBERT *space, $a(\cdot, \cdot)$ be a continuous coercive bilinear form and F a continuous linear functional. Then there exists a unique $u \in V$ such that*

$$a(u, v) = F(v) \quad \forall v \in V. \tag{2.27}$$

Remark 33. Why is this relevant to us? At the first glance it is not obvious what this theorem has to do with numerical approximations. It turns out that any linear partial differential equation (linear PDE) in weak formulation can be written in the form (2.27), where u are the unknowns and v is a test function.

Example 34. Let us revisit (2.24). We can see that we can rewrite it in terms of a and F by setting

$$a(u,v) := \int_\Omega \nabla u \cdot \nabla v \,\mathrm{d}\Omega \tag{2.28}$$

and $F(v) = \int fv \,\mathrm{d}\Omega$ for $u, v \in \mathrm{H}^1(\Omega)$. Note, that a is not coercive on $\mathrm{H}^1(\Omega)$. Consider $u \equiv c$ for $c \in \mathbb{R}$ constant. Then $a(u,u) = 0$ but u is not necessarily zero. However, on $\mathrm{H}^1_0(\Omega)$ a is an inner product and, thus, coercive (POINCARÉ inequality), so we can still deduce that the weak formulation (2.24) has a unique solution for zero boundary conditions.

2.3.1 Standard GALERKIN method

Recall that we require $u \in \mathrm{H}^1_0(\Omega)$ in our example earlier. The problem is that $\mathrm{H}^1(\Omega)$ is a space of infinite dimension. Because no computer can handle infinitely many cases, we have to reduce the problem somehow. Thus we use finite dimensional subspaces of the solution space and calculate the best approximation in this subspace. How these spaces can be constructed, will be explained later on.

Let V be a HILBERT space. Furthermore let $V_h \subset V$ be an n dimensional subspace, $n \in \mathbb{N}$, with a so called discretization parameter h. Our new problem is given by

$$a(u_h, v_h) = F(v_h) \quad \forall v_h \in V_h \tag{2.29}$$

for the discrete solution $u_h \in V_h$. We want that the discrete solution converges to the continuous one for $h \to 0$. Because the properties of a and F are also fulfilled restricted to V_h, the LAX-MILGRAM theorem still guarantees existence and uniqueness of the solution. We will rewrite (2.29) as a linear system of equations. By $\{\varphi_i\}$ for $i = 1, \dots, n$, we denote a basis of V_h. Obviously, there exist coefficients $(U_j)_1^n \subset \mathbb{R}$ such that

$$u_h = \sum_{j=1}^n U_j \varphi_j, \quad \nabla u_h = \sum_{j=1}^n U_j \nabla \varphi_j. \tag{2.30}$$

almost everywhere. The coefficients U_j are also called degrees of freedom or unknowns. Because we can use the basis $\{\varphi_j\}_j$ for the test functions in V_h, we can rewrite (2.29) as

$$a\left(\sum_{j=1}^n U_j \varphi_j(x), \varphi_i\right) = F(\varphi_i) \quad \forall i = 1, \dots, n. \tag{2.31}$$

If we recall that a is a bilinear form, we can also write

$$\sum_{j=1}^n U_j a(\varphi_j, \varphi_i) = F(\varphi_i) \quad \forall i = 1, \dots, n. \tag{2.32}$$

This already looks like a matrix-vector multiplication. If we define $a_{i,j} = a(\varphi_j, \varphi_i)$ and $b_i = F(\varphi_i)$, we obtain the linear problem

$$AU = b, \tag{2.33}$$

which can be solved by any solver for linear systems of equations. The matrix A is often called *stiffness matrix* and b the *load vector*.

Remark 35. What happens if the problem is not linear? One can then still derive a weak formulation of the form

$$F(u, v) = 0 \quad \forall v \in V \tag{2.34}$$

with some nonlinear operator $F : V \times V \to \mathbb{R}$. This problem can also be projected to a finite-dimensional subspace with basis functions of V_h as above. The equation (2.34) can then be rewritten as

$$F(u_h, \varphi_i) = 0 \quad \forall i = 1, \ldots, n, \tag{2.35}$$

to obtain a discrete solution $u_h \in V_h$. This would then be plugged into a non-linear solver. However, the existence or even uniqueness of a solution cannot be guaranteed and must often be assessed on a case-by-case basis. This is an open research topic. There exist some approaches to this problem. A few of them can be found in [10].

2.3.2 Finite element spaces

For now, we handled the problem with an arbitrary subspace V_h. But how would we choose such space and its basis functions?

Notation 36. Let $K \subset \Omega$. We denote by

$$P_k(K) := \left\{ p \colon K \to \mathbb{R} \colon p(x) = \sum_{|\alpha| \leq k} c_\alpha x^\alpha, \quad x \in K \right\} \tag{2.36}$$

the set of all polynomials of degree less than or equal to k and c_α are constant coefficients.

We will construct a mesh by starting with a decomposition $\mathcal{T}_h$ of Ω into open cells K with $\overline{\Omega} = \bigcup_{K \in \mathcal{T}_h} \overline{K}$.

Definition 37. $(K, P_k(K), \Sigma_k)$ is called a *finite element*, where Σ_k denotes the set of degrees of freedom.

Not all $\mathcal{T}_h$ are admissible.

Definition 38. A decomposition of the domain $\Omega \subset \mathbb{R}^d$ into simplices $K \in \mathcal{T}_h$ is admissible if

1. $\overline{\Omega} = \bigcup_{K \in \tau_h} \overline{K}$,
2. any nonempty intersection of two cells $\overline{K_1} \cap \overline{K_2}$ is either a vertex, an edge or a face of both cells.

The following theorem ensures that the combination of all cells still results in V_h being a subset of $\mathrm{H}^1(\Omega)$.

Theorem 39. *If for every $K \in \mathcal{T}_h$, $P_k(K) \subset \mathrm{H}^1(K)$, and $V_h \subset \mathrm{C}^0(\overline{\Omega})$, then $V_h \subset \mathrm{H}^1(\Omega)$.*

Remark 40. If we take a look back at the last theorem, we can see that the continuity of the degrees of freedom across neighboring faces is enough for our solution to be in $\mathrm{H}^1(\Omega)$.

Proof taken from [9]. Let $v \in V_h$ be given. If we can show that the weak derivatives $D^\alpha v$ are in $\mathrm{L}^2(\Omega)$ for $|\alpha| = 1$, we showed the theorem. The natural choice would be the piecewise defined function $w_K := D^\alpha(v|_K) \in \mathrm{L}^2(K)$. Let $\varphi \in \mathrm{C}_0^\infty(\Omega)$ be arbitrary. It holds that

$$\int_\Omega (w\varphi + vD^\alpha\varphi)\,\mathrm{d}x = \sum_{K\in\mathcal{T}_h} \int_K (D^\alpha(v|_K)\varphi + v|_K D^\alpha\varphi)\,\mathrm{d}x, \tag{2.37}$$

where we just used the linearity of the integral. If we recall the definition of the weak derivative (Definition 25), we can see how we can rewrite $D^\alpha(v|_k)$. We obtain

$$\sum_{K\in\tau_h} \int_K (D^\alpha(v|_K)\varphi + v|_K D^\alpha\varphi)\,\mathrm{d}x = \sum_{K\in\tau_h} \left(-\int_K v|_K D^\alpha\varphi + \int_K v|_K D^\alpha\varphi + \int_{\partial K} v|_K \varphi n_K\,\mathrm{d}\partial K \right), \tag{2.38}$$

which is obviously just

$$= \sum_{K\in\tau_h} \int_{\partial K} v|_K \varphi n_K\,\mathrm{d}\partial K, \tag{2.39}$$

where n_k is the outer unit normal for K. We just applied integration by parts to the first term. Because we chose φ to be zero on the boundary of Ω, the integral over each boundary face of Ω becomes zero. For all inner faces $\partial K_1 \cap \partial K_2$ we get the term twice, once for every cell. However both times the normal points in the opposite direction. So our equation reduces to

$$\sum_{K\in\tau_h} \int_{\partial K} v|_K \varphi n_K\,\mathrm{d}\partial K = 0 \quad \forall\varphi \in \mathrm{C}_0^\infty. \tag{2.40}$$

That proofs that w is indeed the weak derivative of v over the whole domain. Because v was arbitrary, it follows that $V_h \subset \mathrm{H}^1(\Omega)$ and this proofs the theorem. □

Another interesting topic to consider, is the order of convergence. If we choose $\dim P_k$ equally on all cells, one can expect a convergence of order $k+1$. However, in practice this is influenced by the time stepping scheme if the problem is time dependent. If the geometry is not rectangular, the mesh approximation to the geometry will also influence the order of convergence. To measure the order of convergence empirically, one defines the *EOC*.

Definition 41. For a set of errors ε_k and discretizations h_k, where $h \to 0$ for $k \to \infty$, we define the *experimental order of convergence (EOC)* as the limit of

$$EOC_k(\varepsilon_k, h_k) = -\frac{\log \frac{\varepsilon_k}{\varepsilon_{k-1}}}{\log \frac{h_k}{h_{k-1}}}., \tag{2.41}$$

as $k \to \infty$, where $h_k \to 0$ as $k \to \infty$.

Remark 42. In practice, it is not possible to compute a real limit. Usually, one calculates four or five refinements.

This short discussion of the finite element method does not claim to be comprehensive, but rather a short overview of important facts, which are necessary to interpret the numerical results presented later.

2.4 Laplace transform

In this section we will introduce the Laplace transform and some of its basic properties. We will loosely follow [29] for this.

Definition 43. For a function $f\colon \Omega \times \mathbb{R}_0^+ \to \mathbb{R}$, we call

$$\mathcal{L}_t\{f\}(s) := \int_0^\infty f(x,t)e^{-st}\,\mathrm{d}t \tag{2.42}$$

the Laplace *transform* of f, where $s \in \mathbb{C}$ is the so-called frequency parameter.

Typically t will be the time variable. In this case this is a transformation of f from the time domain to the frequency domain. Of course this definition only makes sense if (2.42) converges for some s. From complex analysis we know that if it converges for some $s_c = a_c + ib_c$, it converges in $A_c := \{s \in \mathbb{C}\colon a > a_c\}$. Note, that a_c may be $\pm\infty$.

Definition 44. If a_c is minimal such that the integral diverges in $A_d := \{s \in \mathbb{C}\colon a < a_c\}$ we call a_c the *abscissa of convergence.*

Remark 45. If we replace the lower limit of the integral in (2.42) with $-\infty$, we get the *bilateral* Laplace *transform.*

Example 46. An important example for a bilateral Laplace transform is the well-known Gamma function

$$\Gamma(s) = \int_0^\infty x^{s-1}e^{-x}\,\mathrm{d}x = \int_{-\infty}^\infty e^{-st}e^{-e^{-t}}\,\mathrm{d}t. \tag{2.43}$$

This transformation is obtained by setting $x := e^{-t}$, which is valid because x should be positive. The substitution for differentials has been considered.

We can give a relationship between the Laplace and Fourier transforms.

Remark 47. Consider a bilateral LAPLACE transform with $s := ib$. We then get

$$g(x, b) = L_f(x, ib) = \int_{-\infty}^{\infty} e^{-ibt} f(x, t) \, \mathrm{d}t, \tag{2.44}$$

which is essentially the FOURIER transform of f into g.

Theorem 48. *If $L_f(x, s)$ is the* LAPLACE *transform as defined, we can compute f via the inverse transform*

$$f(x, t) = \frac{1}{2\pi i} \oint_{\gamma - i\infty}^{\gamma + i\infty} L_f(x, s) e^{st} \, \mathrm{d}s, \tag{2.45}$$

where γ is a real number such that the path of integration is inside the convergence domain of the LAPLACE *transform.*

In this chapter, we gathered all the important general mathematical background information, which we will use heavily from now on.

3 Modeling

Up to now, our work was in no way specific to the problem that we want to solve. In this chapter, we will be more specific. For this, we will give a (short) introduction into the macroscopic modeling of non-NEWTONIAN fluids. The introduction for this will, however, be intentionally vague as this work is more focused on mathematical than the physical aspects. After that, we will employ some assumptions to make the problem accessible for a master's thesis.

3.1 NAVIER-STOKES Equation

Many books have been written about the NAVIER-STOKES equations. We will use some ideas from [23] and [28] to outline the main facts. Just the incompressible variant of the equations will be considered and the density is normalized to one and therefore will be omitted.

3.1.1 Basics of fluid dynamics

To understand fluid dynamics one must consider two coordinate systems. At first, there is the usual fixed EULERIAN coordinate system (spatial coordinates) that is stationary. Secondly, there is the LAGRANGIAN coordinate system (material coordinates) that follows one particle in the flow. We use u for the velocity, p for the pressure with their normal physical interpretations.

Definition 49 (material derivative)**.** If we want to take the derivative of a quantity $f(x,t)$ in time with respect to the fluid flow direction we get the *material derivative*, which is given by

$$\frac{D}{Dt}f(x,t) := \frac{\partial f}{\partial t} + u \cdot \nabla f. \tag{3.1}$$

The main idea behind the NAVIER-STOKES equation is the concept of *conservation*.

Definition 50. If for a property P, it holds that on an arbitrary finite volume Ω

$$\frac{d}{dt}\int_\Omega P \,\mathrm{d}\Omega = -\int_{\partial\Omega} Pu \cdot n \,\mathrm{d}\partial\Omega + \int_\Omega s \,\mathrm{d}\Omega, \tag{3.2}$$

with s some source or sink and n the unit normal (outward), is valid, then P is *conserved*.

Remark 51. Conservation means that over time a property can only change over the boundary or through sources or sinks inside. This is motivated from the mass, energy and momentum conservation in physics.

3.1.2 Momentum equation

Because we assumed that the density of the fluid is one, the momentum is simply the velocity. The momentum is a conserved property so it holds that

$$\frac{d}{dt}\int_\Omega u \,\mathrm{d}\Omega = -\int_{\partial\Omega} uu^T n \,\mathrm{d}\partial\Omega + \int_\Omega s \,\mathrm{d}\Omega, \tag{3.3}$$

where we get a rank two tensor in the boundary term because we write uu^T as the dyadic product of the two vectors. Using GAUSS's theorem, we can rewrite the boundary integral. This leads to

$$\frac{d}{dt}\int_\Omega u \,\mathrm{d}\Omega = -\int_\Omega \nabla\cdot(uu^T) \,\mathrm{d}\Omega + \int_\Omega s \,\mathrm{d}\Omega. \tag{3.4}$$

Theorem 52 (REYNOLDS transport theorem)**.** *Let f be a vector valued function, u_e be the velocity of* $\mathrm{d}\Omega$*. It holds that*

$$\frac{d}{dt}\int_{\Omega(t)} f \,\mathrm{d}\Omega = \int_{\Omega(t)} \partial_t f \,\mathrm{d}\Omega + \int_{\partial\Omega(t)} (u_e\cdot n) f \,\mathrm{d}\partial\Omega, \tag{3.5}$$

where n is the outward unit normal vector.

Using the previous theorem and realizing that Ω is constant in time, we get

$$\int_\Omega \partial_t u + \nabla\cdot(uu^T) - s \,\mathrm{d}\Omega = 0. \tag{3.6}$$

Theorem 53. *Let f and g be functions $f, g\colon \Omega\subset\mathbb{R}^n\to\mathbb{R}^n$. It holds that*

$$\nabla\cdot(fg^T) = (\nabla\cdot f)g + f\cdot\nabla g. \tag{3.7}$$

As (3.6) holds for every Ω the integrand has to be zero (Fundamental lemma of calculus of variations). This realization and the previous theorem lead to

$$\partial_t u + u\cdot\nabla u + u\nabla\cdot u - s = 0. \tag{3.8}$$

But because we have mass conservation and incompressibility, it holds that $\nabla\cdot u = 0$ (in general $\partial_t\rho + \nabla\cdot(\rho u) = 0$). Therefore, our equation reduces to

$$\partial_t u + u\cdot\nabla u - s = 0 \tag{3.9}$$

This is, apart from s, just the material derivative of u. The only unknown remaining is s. To compute s, we refer to CAUCHY's momentum equation. But first, we have to realize that one can split up s into so-called body forces and surface forces. Hence,

$$\partial_t u + u\cdot\nabla u = f_{\text{body}} + f_{\text{surf}}. \tag{3.10}$$

The body forces can be of various types so we will collect all of them in a function f. In the CAUCHY momentum equation the surface forces are modeled using a so-called symmetric stress tensor τ, such that

$$\partial_t u + u \cdot \nabla u = f + \nabla \cdot \tau. \tag{3.11}$$

It is interesting to note that the entries on the diagonal of τ are the stresses normal to the fluid direction, and all others are parallel to the fluids direction. This motivates to set

$$\tau = -pI + \sigma, \tag{3.12}$$

where p is the pressure and I the corresponding identity matrix. Plugging this into our equation yields

$$\partial_t u + u \cdot \nabla u = -\nabla p + f + \nabla \cdot \sigma. \tag{3.13}$$

Note that this equation is not complete. One has to give a model for the so-called stress tensor σ and for the pressure p. This however, is very fluid dependent. The next section will deal with the derivation of such models for σ, p for non-NEWTONIAN fluids.

3.2 Non-NEWTONIAN fluids

Before we start discussing the two related models used to model the stress tensor, we will give a short introduction into non-NEWTONIAN fluids. A general introduction can be found in [25].

3.2.1 What is a non-NEWTONIAN fluid and why is it relevant?

Generally speaking, a non-NEWTONIAN fluid differs from a NEWTONIAN fluid in the behavior of the viscosity. NEWTON formulated several laws, which an ideal fluid should obey. One of them was that the viscosity should be independent of the stress that the fluid encounters. All fluids, which obey this law, are called NEWTONIAN fluids and the others are called non-NEWTONIAN. Non-NEWTONIAN fluids are far more spread as one would imagine: from ketchup to honey and toothpaste to the famous water-cornstarch mixture that jumps on loudspeakers; these types of fluids are everywhere. Even granular movement in space and liquid plastic in 3D printers behaves that way. So there are plenty of reasons to care about the difficult task of modeling these fluids.

There are many different types of non-NEWTONIAN behavior. Ketchup for example has a so called *pseudoplastic* behavior, that means that the viscosity decreases with increased stress. In this work we will cover the fluids that have the so-called *viscoelastic* behavior. That means that elastic and viscous effects are working together as a linear combination. Liquid plastic and granular flows often show this type of non-NEWTONIAN behavior.

Figure 3.1: Illustration of the Maxwell model (from [27])

3.2.2 Maxwell and Oldroyd-B models

The Maxwell model is one of the easiest models used to describe such viscoelastic fluids. It is obtained by combining dampers and springs and model them with known physical laws. The easiest case is shown in Figure 3.1. The damper accounts for the viscous and the spring for the elastic part of the fluid's behavior. For more information see [26]. The Maxwell model states that

$$\frac{\lambda}{\mu_p}\partial_t\sigma + \frac{1}{\mu_p}\sigma = \partial_t \boldsymbol{B}, \tag{3.14}$$

where $\mu_p \in \mathbb{R}$ is the polymer viscosity, λ the so called relaxation time and $\boldsymbol{B}$ the strain or Finger tensor. If we solve this ordinary differential equation for σ, we obtain

$$\sigma(t) = \int_{-\infty}^{t} -\mu_p \exp\left(-\frac{t-t'}{\lambda}\right) \partial_{t'}\boldsymbol{B}(t,t')\,\mathrm{d}t'. \tag{3.15}$$

We now deduced a model for σ. However, it is unclear how $\boldsymbol{B}$ should be calculated. Oldroyd proposed to use the so called upper-convected time derivative $\overset{\triangledown}{\boldsymbol{B}}$ to obtain an ordinary differential equation for $\boldsymbol{B}$. This leads to the governing equation

$$\overset{\triangledown}{\boldsymbol{B}} := \partial_t \boldsymbol{B} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{B} - (\nabla\boldsymbol{u})\boldsymbol{B} - \boldsymbol{B}(\nabla\boldsymbol{u})^T = 0, \tag{3.16}$$

where $\boldsymbol{u}$ denotes the velocity field. The difference between the upper-convected Maxwell model (UCM) and the Oldroyd-B model (both introduced by James G. Oldroyd) lies in the solvent viscosity. The UCM model calculates the flow using the Euler's equations and the Oldroyd-B model uses the Navier-Stokes equations. Note, that one can extend the formula for σ by allowing

$$\sigma(t) = \int_{-\infty}^{t} -\partial_{t'}\boldsymbol{B}(t,t')G(t,t')\,\mathrm{d}t', \tag{3.17}$$

where G is the so called *relaxation modulus*. This opens the possibility for more advanced models. The whole integral for σ can be interpreted as a history dependency. Every past timestep affects the current one. The relaxation modulus can be seen as a weight function in this context. We want that the more a past timestep lies in the past the less relevant it is for the current calculation.

3.3 Dimension reduction

After we have discussed the physical aspects in the previous sections, we will now focus on making these equations easier to handle. To provide better readability we will omit the arguments of the arising functions, where it is obvious. We start from the well-known NAVIER-STOKES equations

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \boldsymbol{f} + \nabla \cdot \sigma + \mu_s \Delta \boldsymbol{u} - \nabla p, \tag{3.18}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{3.19}$$

$$\boldsymbol{u} = \boldsymbol{u}_b \text{ on } \partial\hat{\Omega}, \tag{3.20}$$

$$\boldsymbol{u}(t=0) = \boldsymbol{u}_0, \tag{3.21}$$

where $t \in [0, T]$ is the current time and T the end time of our simulation. We set a domain $\hat{\Omega} \subset \mathbb{R}^3$ and an element $\hat{\boldsymbol{x}} \in \hat{\Omega}$. We also have $\boldsymbol{u}(t, \hat{\boldsymbol{x}})\colon \mathbb{R}^+ \times \hat{\Omega} \to \mathbb{R}^3$ as the velocity vector, $\boldsymbol{f}(t, \hat{\boldsymbol{x}})\colon \mathbb{R}^+ \times \hat{\Omega} \to \mathbb{R}^3$ a generic source or sink term, which is modeling the body forces, and $\sigma(t, \hat{\boldsymbol{x}})\colon \mathbb{R}^+ \times \hat{\Omega} \to \mathbb{R}^{3\times 3}$ the stress tensor. As usual $\mu_s \in \mathbb{R}_0^+$ denotes the solvent viscosity. We will use $p(t, \hat{\boldsymbol{x}})\colon \mathbb{R}^+ \times \hat{\Omega} \to \mathbb{R}$ as the pressure. Our boundary condition for $\boldsymbol{u}$ is denoted by $\boldsymbol{u}_b$. The difference between NEWTONIAN and non-NEWTONIAN fluids lies in the definition of σ. In the UCM or OLDROYD-B model the stress tensor is given by (see 3.2):

$$\sigma(t, \hat{\boldsymbol{x}}) = \int_{-\infty}^{t} -\partial_{t'} \boldsymbol{B}(t, t', \hat{\boldsymbol{x}}) G(t, t') \, \mathrm{d}t', \tag{3.22}$$

where $G\colon \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}$ is a weight function which decays fast in t'. In both physical models, which we have discussed in section 3.2, G is given by:

$$G(t, t') = \mu_p \cdot e^{-(t-t')/\lambda}, \tag{3.23}$$

where $\mu_p \in \mathbb{R}^+$ is the polymer viscosity and $\lambda \in \mathbb{R}$ the relaxation time. Both are fluid-dependent parameters. However, G can be more complex. For example, if G becomes space-dependent, this introduces many new problems. This is why we will stick to (3.23) for most of this work.

The term $\boldsymbol{B}\colon \mathbb{R}^+ \times \mathbb{R}^+ \times \hat{\Omega} \to \mathbb{R}^{3\times 3}$ in (3.22) is called the FINGER *tensor* and obeys the differential equation

$$\partial_t \boldsymbol{B} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{B} - (\nabla \boldsymbol{u})\boldsymbol{B} - \boldsymbol{B}(\nabla \boldsymbol{u})^T = \boldsymbol{0}. \tag{3.24}$$

We introduce the initial and boundary conditions

$$\boldsymbol{B}(t', t', \hat{\boldsymbol{x}}) \equiv \boldsymbol{1}, \tag{3.25}$$

$$\boldsymbol{B}(0, t', \hat{\boldsymbol{x}}) \equiv \boldsymbol{1}, \tag{3.26}$$

where both conditions imply the assumption that we start the computation stress free.

The variable t', which appears in the last equation, is called "history variable". We will discuss its impact later on.

For now, we will assume that $\boldsymbol{u}, \nabla p$ and $\boldsymbol{f}$ are only non-zero in one direction, that is,

$$\boldsymbol{u} = (0, 0, u)^T, \quad \nabla p = (0, 0, \partial_3 p), \quad \boldsymbol{f} = (0, 0, f), \tag{3.27}$$

where $u, f, p \colon \mathbb{R}^+ \times \hat{\Omega} \to \mathbb{R}$. Because $\nabla \cdot \boldsymbol{u} = 0$ (see (3.19)), we immediately obtain $\partial_3 u = 0$, which leads to

$$[(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}]_3 = u_1 \partial_1 u_3 + u_2 \partial_2 u_3 + u_3 \partial_3 u_3 = 0 \tag{3.28}$$

and obviously the other components of this term become zero as well. Therefore, we loose the advection term in the Navier-Stokes equations.
We will now examine the diffusion term

$$\Delta \boldsymbol{u} = \begin{pmatrix} \Delta u_1 \\ \Delta u_2 \\ \Delta u \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \partial_1^2 u + \partial_2^2 u + \partial_3^3 u \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \partial_1^2 u + \partial_2^2 u \end{pmatrix}. \tag{3.29}$$

If we take a close look at (3.18), we can see that in order to fulfill the equations for u_1 and u_2 ($0 = (\nabla \cdot \sigma)_l, \quad l \in \{1, 2\}$), the block with indexes $(i, j) : i, j \in \{1, 2\}$ of σ has to be constant in space. The last row and last column are the same because of symmetry of the tensor. We have to make sure that they are constant in the third spatial direction, which is obviously given by our assumptions.

After we treated the Navier-Stokes equations in the last paragraph, we will now focus on the governing equation for the Finger tensor. We already saw that only its last row and last column are of interest for us. Let us look at (3.24) in index notation, which reads

$$\partial_t \boldsymbol{B}_{i,j} + \sum_{k=1}^{3} \boldsymbol{u}_k \partial_k \boldsymbol{B}_{i,j} - \sum_{k=1}^{3} \partial_k \boldsymbol{u}_i \boldsymbol{B}_{k,j} - \sum_{k=1}^{3} \boldsymbol{B}_{i,k} \partial_k \boldsymbol{u}_j = 0 \tag{3.30}$$

We can now put our assumptions for $\boldsymbol{u}$ into it. Additionally, $\partial_3 \boldsymbol{B}_{3,j}$ should be 0. We will just focus on the last row for now. This leads to

$$\partial_t \boldsymbol{B}_{3,j} - \sum_{k=1}^{2} \partial_k u \boldsymbol{B}_{k,j} - \begin{Bmatrix} 0 & , j = 1, 2 \\ \sum_{k=1}^{2} \boldsymbol{B}_{3,k} \partial_k u & , j = 3 \end{Bmatrix} = 0. \tag{3.31}$$

If we revisit (3.30), we observe that

$$\partial_t \boldsymbol{B}_{i,j} = -u(\partial_3 \boldsymbol{B}_{i,j}) \; \forall (i, j) \in \{1, 2\}^2 \tag{3.32}$$

This shows that if we choose this block constant in the beginning, as required by our assumptions, it will not change over time. In our case we will choose the identity matrix for this. These observations also justify that we only care about the last row and ignore the other ones. If we define $\hat{\boldsymbol{b}}_j := \boldsymbol{B}_{3,j}$, we obtain the equation

$$\partial_t \hat{\boldsymbol{b}} = \begin{pmatrix} \partial_1 u \\ \partial_2 u \\ 2\left((\partial_1 u)\boldsymbol{b}_1 + (\partial_2 u)\boldsymbol{b}_2\right) \end{pmatrix}. \tag{3.33}$$

We have stated above that $\partial_3 \hat{\boldsymbol{b}}_3 = 0$. This directly leads to $\partial_3 \sigma_{3,3} = 0$. Now, taking a good look at (3.18), we see that $\sigma_{3,3}$ only contributes with its spatial derivative in x_3 direction. So it vanishes from the equations completely. We define

$$\partial_t \boldsymbol{b} = \begin{pmatrix} \partial_1 u \\ \partial_2 u \end{pmatrix}. \tag{3.34}$$

Furthermore, we define $\boldsymbol{s} := \sigma_{3,j}$. Consequentially, we obtain the governing equation

$$\boldsymbol{s}(t, \hat{\boldsymbol{x}}) = \int_{-\infty}^{t} -\partial_{t'} \boldsymbol{b}(t, t', \hat{\boldsymbol{x}}) G(t, t') \, \mathrm{d}t'. \tag{3.35}$$

We can now define our new system of equations

$$\partial_t u(t, \boldsymbol{x}) = -\partial_3 p + f + \nabla \cdot \boldsymbol{s} + \mu_s \Delta u, \tag{3.36}$$

$$\boldsymbol{s}(t, \boldsymbol{x}) = \int_{-\infty}^{t} -\partial_{t'} \boldsymbol{b}(t, t', \boldsymbol{x}) G(t, t') \, \mathrm{d}t', \tag{3.37}$$

$$\partial_t \boldsymbol{b}(t, t', \boldsymbol{x}) = \nabla u, \tag{3.38}$$

where $\Omega \subset \mathbb{R}^2$ is the domain with an element $\boldsymbol{x}$. We let $u(t, \boldsymbol{x}), p(t, \boldsymbol{x}), f(t, \boldsymbol{x}) \colon \mathbb{R}^+ \times \mathbb{R}^2 \to \mathbb{R}$ denote the third component of the velocity, the pressure and a generic source/sink term respectively. Δ stands for the two-dimensional LAPLACE-operator. To define the stress and finger tensor, we use $\boldsymbol{s}(t, \boldsymbol{x})$ and $\boldsymbol{b}(t, \boldsymbol{x}) \colon \mathbb{R}^+ \times \mathbb{R}^2 \to \mathbb{R}^2$ respectively.

By using our assumptions, we now deduced a true two-dimensional problem. This can be interpreted as a slice orthogonally to the fluid's flow direction. During this deduction we silently eliminated the nonlinearity in $\boldsymbol{b}_3$ because it is not relevant for the equations anymore. This makes work much easier, as we will see later on.

3.4 Transformation of the equation into the frequency domain

The next step, that is usually taken, is the introduction of an "age" variable $\tau = t - t'$. This is sensible because if we take a look at (3.37), we see that only the times t' close to t are relevant (recall that G is exponentially declining the larger the difference between t and t'). The further back in history t' gets, the less contribution it makes. By putting this transformation into our equations, we obtain

$$\boldsymbol{s}(t, \boldsymbol{x}) = \int_{0}^{\infty} \partial_\tau \boldsymbol{b}(t, t - \tau, \boldsymbol{x}) G(t, t - \tau) \, \mathrm{d}\tau. \tag{3.39}$$

This integral runs "backwards" in time. Using the chain rule the governing equation for the FINGER tensor transforms to

$$\partial_t \boldsymbol{b} + \partial_\tau \boldsymbol{b} = \nabla u, \tag{3.40}$$

with the corresponding boundary and initial conditions

$$\boldsymbol{b}(t,t) = \mathbf{0}, \tag{3.41}$$
$$\boldsymbol{b}(t,0) = \mathbf{0}. \tag{3.42}$$

At first glance this appears to have made our situation worse because we now have two time derivatives in the equation. Luckily, there is a tool that can help us with that: the LAPLACE transformation. This is explained extensively in section 2.4. We have

$$\mathcal{L}_\tau\{\boldsymbol{b}\}(t,\boldsymbol{x},s) = \int_0^\infty \boldsymbol{b}(t,t-\tau,\boldsymbol{x})e^{-s\tau}\,\mathrm{d}\tau \tag{3.43}$$

as the LAPLACE transform of $\boldsymbol{b}$, where $s \in \mathbb{C}$ is the frequency parameter. We can switch derivative and integral because we assume both limits to exist.

$$\partial_t\mathcal{L}_\tau\{\boldsymbol{b}\}(t,\boldsymbol{x},s) + \mathcal{L}_\tau\{\partial_\tau\boldsymbol{b}\}(t,\boldsymbol{x},s) = \frac{1}{s}\nabla u. \tag{3.44}$$

However, because ∂_τ is not independent of τ, we have to use integration by parts to transform it to

$$\mathcal{L}_\tau\partial_\tau\boldsymbol{b}(t,\boldsymbol{x},s) = \int_0^\infty \partial_\tau\boldsymbol{b}(t,t-\tau,\boldsymbol{x})e^{-s\tau}\,\mathrm{d}\tau \tag{3.45}$$
$$= \lim_{r\to\infty}\boldsymbol{b}(t,t-r,\boldsymbol{x})e^{-sr} - \boldsymbol{b}(t,t,\boldsymbol{x})e^{-s\cdot 0} + s\int_0^\infty \boldsymbol{b}(t,t-\tau)e^{-s\tau}\,\mathrm{d}\tau. \tag{3.46}$$

The first term vanishes because the exponential function dominates the behaviour of this term. This is a necessary condition for the LAPLACE transform to converge.

$$= -\boldsymbol{b}(t,t,\boldsymbol{x}) + s\mathcal{L}_\tau\{\boldsymbol{b}\}(t,\boldsymbol{x},s) \tag{3.47}$$
$$= s\mathcal{L}_\tau(t,\boldsymbol{x},s). \tag{3.48}$$

We have successfully eliminated the time derivative in τ direction. The governing equation for the LAPLACE transform of the finger tensor is given by

$$(\partial_t + s)\mathcal{L}_\tau\boldsymbol{b}(t,\boldsymbol{x},s) = \frac{1}{s}\nabla u \tag{3.49}$$

The corresponding initial condition is

$$\mathcal{L}_\tau\{\boldsymbol{b}\}(0,\boldsymbol{x},s) = \mathbf{0}. \tag{3.50}$$

As a next step, we will introduce the conformation tensor, which non-dimensionalizes $\mathcal{L}_\tau\{\boldsymbol{b}\}$. It is defined as

$$\boldsymbol{C}_s(t,\boldsymbol{x}) := s\mathcal{L}_\tau\{\boldsymbol{b}\}(t,\boldsymbol{x},s), \tag{3.51}$$

and has the governing equation

$$\partial_t\boldsymbol{C}_s(t,\boldsymbol{x}) + s\boldsymbol{C}_s(t,\boldsymbol{x}) = \nabla u \tag{3.52}$$

This transformation worked very well and eliminated many problems, but the question remains: how do we retrieve $\boldsymbol{s}$? Let's recall (3.39). Because we only have a finite history and start stress free, we can cut the integral off at t

$$\boldsymbol{s}(t, \boldsymbol{x}) = \int_0^t \partial_\tau \boldsymbol{b}(t, t-\tau, \boldsymbol{x}) G(t, t-\tau) \, \mathrm{d}\tau. \tag{3.53}$$

If we assume that the LAPLACE transform of G is well-defined and allow that G is space dependent, we can set

$$g(\boldsymbol{x}, t, s) = \int_0^\infty G(t, t', \boldsymbol{x}) e^{-st'} \, \mathrm{d}t'. \tag{3.54}$$

Then $\boldsymbol{s}$ is simply given by an inverse Laplace transform so specifically

$$\boldsymbol{s}(t, \boldsymbol{x}) = \mathcal{L}^{-1}\{\boldsymbol{C}_s(t, \boldsymbol{x}) g(t, \boldsymbol{x}, s)\}(t, \boldsymbol{x}). \tag{3.55}$$

At first glance one may be tempted to be satisfied with this. However, numerically it is unclear how this could result in an efficient algorithm. One of the nice properties of this approach is that if we choose

$$G(t, t-\tau) = \mu_p e^{-\tau/\lambda}, \quad s = \frac{1}{\lambda}, \tag{3.56}$$

where λ is the material specific relaxation time, we can retrieve the MAXWELL / OLDROYD-B model

$$\boldsymbol{s}(t, \boldsymbol{x}) = \mu_p \int_0^t \partial_\tau \boldsymbol{b}(t, t-\tau, \boldsymbol{x}) e^{-\tau/\lambda} \, \mathrm{d}\tau, \tag{3.57}$$

using integration by parts,

$$= \mu_p \boldsymbol{b}(t, 0) e^{-t/\lambda} - \mu_p \boldsymbol{b}(t, t) e^{-0/\lambda} + \frac{\mu_p}{\lambda} \int_0^t \boldsymbol{b}(t, t-\tau) e^{-\tau/\lambda} \, \mathrm{d}\tau. \tag{3.58}$$

If we use the boundary and initial conditions and extend the integral to ∞ (still equal because for $\tau > t$ $\boldsymbol{b}(t, \tau) = 0$), we obtain

$$= \frac{\mu_p}{\lambda} \int_0^\infty \boldsymbol{b}(t, t-\tau) e^{-\tau/\lambda} \, \mathrm{d}\tau \tag{3.59}$$

$$= \frac{\mu_p}{\lambda} L_{\boldsymbol{b}}(t, \boldsymbol{x}, 1/\lambda) = \mu_p \boldsymbol{C}_{1/\lambda}. \tag{3.60}$$

So using the assumptions on G and $\boldsymbol{u}$, we removed the integral term completely! In this example the stress tensor is just the scaled conformation tensor.

4 Simulation

This chapter will deal with all aspects of numerical considerations. At first, we will build the numerical scheme and discuss its convergence properties. After that we will take a look at some results that could be obtained.

4.1 Discretization and building the numerical scheme

The next step will be to build a numerical scheme. For this, we first need the weak formulation. Then we will discuss the discretizations in space and time. Our equations are given by

$$\partial_t u(t, \boldsymbol{x}) = -\partial_3 p + f + \nabla \cdot \boldsymbol{s} + \mu_s \Delta u, \tag{4.1}$$

$$\boldsymbol{s}(t, \boldsymbol{x}) = \mu_p \boldsymbol{C}_{1/\lambda}, \tag{4.2}$$

$$\partial_t \boldsymbol{C}_{1/\lambda}(t, \boldsymbol{x}) = -\frac{1}{\lambda} \boldsymbol{C}_{1/\lambda}(t, \boldsymbol{x}) + \nabla u \tag{4.3}$$

4.1.1 Weak formulation

To obtain the weak formulation for equations (4.1) to (4.3), we will use the standard approach as discussed in subsection 2.3.2. So let $\varphi = (\varphi_1, \varphi_2) \in \mathrm{C}_0^\infty(\Omega \times \mathbb{R})$ with values in $\mathbb{R}^2$ and $\psi \in \mathrm{C}_0^\infty(\Omega \times \mathbb{R})$ with values in $\mathbb{R}$ be test functions of compact support. We will discuss simulation cases, where the boundary conditions are not 0 later on, but for now we will use this restriction. By integrating in space, we can transform (4.3) to

$$\int_\Omega (\partial_t \boldsymbol{C}_{1/\lambda} + \frac{1}{\lambda} \boldsymbol{C}_{1/\lambda}) \cdot \varphi \, \mathrm{d}\Omega = \int_\Omega \nabla u \cdot \varphi \, \mathrm{d}\Omega. \tag{4.4}$$

We will not modify this equation further, as we do not have any knowledge about the boundary values of $\boldsymbol{C}$. The spatial derivatives of u however, are not a problem during the simulation as u is not time-dependent in this equation and therefore known. Equation (4.2) does not need to be handled, so what remains is equation (4.1), which is also the most complicated one. Integration over Ω and multiplication with the respective test function yields

$$\int_\Omega (\partial_t u + \partial_3 p - \nabla \cdot \boldsymbol{s} - \mu_s \Delta u) \psi \, \mathrm{d}\Omega = 0. \tag{4.5}$$

Using the linearity of the integral, we get

$$\int_\Omega (\partial_t u) \psi \, \mathrm{d}\Omega + \int_\Omega \partial_3 p \psi \, \mathrm{d}\Omega - \int_\Omega (\nabla \cdot \boldsymbol{s}) \psi \, \mathrm{d}\Omega - \mu_s \int_\Omega \psi \Delta u \, \mathrm{d}\Omega = 0. \tag{4.6}$$

The first two terms do not need additional work. So we will focus on the latter two. The last term can be rewritten using well-known GREEN's Theorem. So we obtain

$$\int_\Omega \psi \Delta u = \int_\Omega \nabla u \cdot \nabla \psi \, \mathrm{d}\Omega. \tag{4.7}$$

Applying basic calculus and integration by parts enables us to transform the remaining term

$$\int_\Omega \psi (\nabla \cdot \boldsymbol{s}) \, \mathrm{d}\Omega = \int_{\partial\Omega} \psi \boldsymbol{s} \, \mathrm{d}\partial\Omega - \int_\Omega \boldsymbol{s} \cdot \nabla \psi \, \mathrm{d}\Omega. \tag{4.8}$$

Because we chose $\psi = u = 0$ on $\partial\Omega$, the boundary integral becomes 0. Inserting both transformations into the original equation, yields

$$\int_\Omega (\partial_t u)\psi \, \mathrm{d}\Omega + \int_\Omega \partial_3 p \psi \, \mathrm{d}\Omega + \int_\Omega \boldsymbol{s} \cdot \nabla \psi \, \mathrm{d}\Omega + \mu_s \int_\Omega \nabla u \cdot \nabla \psi \, \mathrm{d}\Omega = 0. \tag{4.9}$$

In summary, we can give the following system of weak formulations for our governing equations

$$\int_\Omega (\partial_t u)\psi \, \mathrm{d}\Omega + \int_\Omega \partial_3 p \psi \, \mathrm{d}\Omega + \int_\Omega \boldsymbol{s} \cdot \nabla \psi \, \mathrm{d}\Omega + \mu_s \int_\Omega \nabla u \cdot \nabla \psi \, \mathrm{d}\Omega = 0, \tag{4.10}$$

$$\boldsymbol{s} = \mu_p \boldsymbol{C}_{1/\lambda}, \tag{4.11}$$

$$\int_\Omega (\partial_t \boldsymbol{C}_{1/\lambda} + \frac{1}{\lambda}\boldsymbol{C}_{1/\lambda}) \cdot \varphi \, \mathrm{d}\Omega = \int_\Omega \nabla u \cdot \varphi \, \mathrm{d}\Omega. \tag{4.12}$$

4.1.2 Time discretization

Our goal in this subsection is to discretize the time dimension in this equation. To achieve this goal, we will introduce the timesteps

$$0 = t_0, t_1, \ldots, t_N = T. \tag{4.13}$$

The time derivative will be approximated using the implicit EULER method. To be specific, we approximate

$$\partial_t (\cdot)^{n+1} \approx \frac{(\cdot)^{n+1} - (\cdot)^n}{\Delta t}, \tag{4.14}$$

where n refers to the n-th timestep, which we assume is already known, and Δt is the timestep width. If the argument, on which the differential operator operates, is linear then this results in a linear system to be solved in every timestep. In return for the work we put into every timestep, this method is stable for every timestep width. To learn more about the stability of RUNGE-KUTTA methods in general follow [13].

For ease of notation we set $(\cdot)^{n+1} := (\cdot)$. If we put (4.14) into the weak form (4.10)-(4.12) at $t = t_{n+1}$, we obtain

$$\int_\Omega (u - u^n)\psi \, \mathrm{d}\Omega + \Delta t \left(\int_\Omega \partial_3 p \psi \, \mathrm{d}\Omega + \int_\Omega \boldsymbol{s} \cdot \nabla \psi \, \mathrm{d}\Omega + \mu_s \int_\Omega \nabla u \cdot \nabla \psi \, \mathrm{d}\Omega \right) = 0, \tag{4.15}$$

$$\boldsymbol{s} = \mu_p \boldsymbol{C}_{1/\lambda}, \tag{4.16}$$

$$\int_\Omega (\boldsymbol{C}_{1/\lambda} - \boldsymbol{C}^n_{1/\lambda} + \frac{\Delta t}{\lambda}\boldsymbol{C}_{1/\lambda}) \cdot \varphi \, \mathrm{d}\Omega = \Delta t \int_\Omega \nabla u \cdot \varphi \, \mathrm{d}\Omega. \tag{4.17}$$

There are numerous possible means to solve the resulting linear system.

4.1.3 Spatial discretization

Now that we have transformed the equation in their weak forms, the next thing to consider is the spatial discretization. We will use the finite element method (FEM) for that. We will write the equations in the form $AU = b$. This is not necessary for solving it using our software framework FEniCS (more on FEniCS later on) but for understanding the problem it is nonetheless relevant.

At first we have to break the problem down into the bilinear form $a(U, \phi)$ and the linear form $F_{n+1}(\phi)$. We will define $U, \phi \colon \mathbb{R}^2 \to \mathbb{R}^3$ as

$$U = \begin{pmatrix} \boldsymbol{C}_{1/\lambda} \\ u \end{pmatrix} \text{ and } \phi = \begin{pmatrix} \varphi \\ \psi \end{pmatrix}. \tag{4.18}$$

Note that these vectors have three entries each. Because we choose the components of φ and ψ linear independent, we can write the whole system as a sum of (4.15)-(4.17). Specifically we obtain

$$\begin{aligned} &\int_\Omega [u\psi + (1 + \frac{\Delta t}{\lambda})\boldsymbol{C}_{1/\lambda} \cdot \varphi - u^n \psi - \boldsymbol{C}^n_{1/\lambda} \cdot \varphi \\ &+ \Delta t \left(\partial_3 p \psi - \nabla u \cdot \varphi + \mu_p \boldsymbol{C}_{1/\lambda} \cdot \nabla \psi + \mu_s \nabla u \cdot \nabla \psi\right)] \,\mathrm{d}\Omega = 0. \end{aligned} \tag{4.19}$$

Putting U and ϕ into (4.19) results in the equation

$$\begin{aligned} &\int_\Omega [U \cdot \phi + \Delta t((\mu_p U_{1:2} + \mu_s \nabla U_3) \cdot \nabla \phi_3) + \frac{\Delta t}{\lambda} U_{1:2} \cdot \phi_{1:2} - \Delta t \nabla U_3 \cdot \phi_{1:2}] \,\mathrm{d}\Omega \\ &= \int_\Omega [U^n \cdot \phi - \Delta t \partial_3 p \phi_3] \,\mathrm{d}\Omega. \end{aligned} \tag{4.20}$$

We denoted with 1:2 that the vector containing the first and second entry should be taken. We define $a(U, \phi)$ as the left hand side of the equation. That this is a bilinear form is trivially true due to the linearity of the integral and the dot product. However, it is not symmetric and therefore no inner product. That means, that our resulting matrix A is not symmetric, which will impact the choice of a suited linear solver.

The right hand side of (4.20) is defined as the linear form F, which only consists of the term from the time discretization and the derivative of the pressure. We will proof in 4.1.4 that this problem has a unique solution.

We will now discuss, which finite elements should be used. The software FEniCS provides a mesh generator, that uses triangles. We define the ansatz functions ϕ as

$$\phi \in P_1(\Omega) \times P_1(\Omega) \times P_1(\Omega), \tag{4.21}$$

where

$$P_1(\Omega) := \{\varphi \in \mathrm{C}^0(\Omega) \colon \varphi|_K \in P_1(K) \quad \forall K \in \mathcal{T}_h\}. \tag{4.22}$$

In this definition we use $\mathcal{T}_h$ and $P_1(K)$ as defined in subsection 2.3.2. This is the smallest order a finite element can have to guarantee continuity across the cell boundary. A higher order element is also not necessary because we only use a time integration scheme of order 1.

4.1.4 Existence and uniqueness of a solution

We now want to proof, with the help of Theorem 32 (Lax-Milgram), that there exists a unique solution for the problem (4.20). Firstly, we have to discuss the form of the solution space. Because $\boldsymbol{C}_{1/\lambda}$ has no spatial derivatives in the governing equations, we assume $\boldsymbol{C}_{1/\lambda} \in \mathrm{L}^2(\Omega) \times \mathrm{L}^2(\Omega)$. The velocity appears with a spatial derivative of order one, so we want $u \in \mathrm{H}^1(\Omega)$. This leads to

$$U \in \mathrm{L}^2(\Omega) \times \mathrm{L}^2(\Omega) \times \mathrm{H}^1(\Omega) =: X(\Omega). \tag{4.23}$$

The test function ϕ should be of the same regularity. We define a norm on this space via the inner product. The product space of Hilbert spaces has an easy inner product

$$\langle U, \phi \rangle_X = \langle U_1, \phi_1 \rangle_{\mathrm{L}^2} + \langle U_2, \phi_2 \rangle_{\mathrm{L}^2} + \langle U_3, \phi_3 \rangle_{\mathrm{H}^1}, \tag{4.24}$$

where we use the inner products on L^2 and H^1 as defined in section 2.2. This inner product induces a norm on X, which we will use to show the existence of a solution. It is given by

$$\|U\|_X^2 = \|U_{1:2}\|_{\mathrm{L}^2}^2 + \|U_3\|_{\mathrm{H}^1}^2 \tag{4.25}$$

Theorem 54. *Let $X = \mathrm{L}^2 \times \mathrm{L}^2 \times \mathrm{H}^1$ be a Hilbert space, $\mu_s > 0$ and $\mu_s(\frac{1}{\Delta t} + \frac{1}{\lambda}) > \frac{1}{4}(\mu_p - 1)^2$. Then the problem in (4.20) has a unique solution in X.*

Proof. We will use theorem 32 (Lax-Milgram) to show this theorem. Therefore we have to show $|a(U, \phi)| \leq \beta \|U\|_X \|\phi\|_X$ for all $U, \phi \in X$ and for $\beta > 0$ (continuity) . Secondly we have to show coercivity so $a(U, U) \geq \alpha \|U\|_X^2$ for all $U \in X$.

Let us consider the continuity inequality first. Because a is a bilinear form it is sufficient to show

$$a(U, \phi) \leq \beta \|U\|_X \|\phi\|_X . \tag{4.26}$$

The bilinear form a is given by

$$a(U, \phi) = \int U \cdot \phi + \Delta t(\nabla \phi_3(\mu_p U_{1:2} + \mu_s \nabla U_3)) + \frac{\Delta t}{\lambda} U_{1:2}\phi_{1:2} - \Delta t \nabla U_3 \cdot \phi_{1:2} \, \mathrm{d}\Omega. \tag{4.27}$$

By expanding all terms we obtain

$$a(U, \phi) = \int U \cdot \phi + \Delta t \mu_p U_{1:2} \nabla \phi_3 + \Delta t \mu_s \nabla \phi_3 \cdot \nabla U_3 + \frac{\Delta t}{\lambda} U_{1:2}\phi_{1:2} - \Delta t \nabla U_3 \cdot \phi_{1:2} \, \mathrm{d}\Omega. \tag{4.28}$$

Using the linearity of the integral and the positivity of a norm, it yields

$$a(U, \phi) \leq \left(1 + \frac{\Delta t}{\lambda}\right) \|U_{1:2}\|_{\mathrm{L}^2} \|\phi_{1:2}\|_{\mathrm{L}^2} + \Delta t \mu_p \|U_{1:2}\|_{\mathrm{L}^2} \|\phi_{1:2}\|_{\mathrm{L}^2} \tag{4.29}$$

$$+ \Delta t \mu_s \|\nabla \phi_3\|_{\mathrm{L}^2} \|\nabla U_3\|_{\mathrm{L}^2} + \|U_3\|_{\mathrm{L}^2} \|\phi_3\|_{\mathrm{L}^2} + \Delta t \|\nabla U_3\|_{\mathrm{L}^2} \|\phi_{1:2}\|_{\mathrm{L}^2} . \tag{4.30}$$

To deal with the constant factors, we introduce

$$\beta = 2 \max\left(1 + \frac{\Delta t}{\lambda}, \Delta t \mu_p, \Delta t \mu_s, \Delta t\right). \tag{4.31}$$

That way we obtain

$$a(U,\phi) \leq \frac{\beta}{2}\big(\|U_{1:2}\|_{\mathrm{L}^2} \|\phi_{1:2}\|_{\mathrm{L}^2} + \|U_{1:2}\|_{\mathrm{L}^2} \|\nabla\phi_3\|_{\mathrm{L}^2} + \|U_3\|_{\mathrm{L}^2} \|\phi_3\|_{\mathrm{L}^2} \tag{4.32}$$

$$+ \|\nabla U_3\|_{\mathrm{L}^2} \|\nabla\phi_3\|_{\mathrm{L}^2} + \|\nabla U_3\|_{\mathrm{L}^2} \|\phi_{1:2}\|_{\mathrm{L}^2} \big). \tag{4.33}$$

One can observe that the last equation can be interpreted as an inner product of the $\mathbb{R}^5$. This reinterpretation yields

$$a(U,\phi) \leq \frac{\beta}{2} \left(\begin{pmatrix} \|U_{1:2}\|_{\mathrm{L}^2} \\ \|U_{1:2}\|_{\mathrm{L}^2} \\ \|U_3\|_{\mathrm{L}^2} \\ \|\nabla U_3\|_{\mathrm{L}^2} \\ \|\nabla U_3\|_{\mathrm{L}^2} \end{pmatrix} \cdot \begin{pmatrix} \|\phi_{1:2}\|_{\mathrm{L}^2} \\ \|\nabla\phi_3\|_{\mathrm{L}^2} \\ \|\phi_3\|_{\mathrm{L}^2} \\ \|\nabla\phi_3\|_{\mathrm{L}^2} \\ \|\phi_{1:2}\|_{\mathrm{L}^2} \end{pmatrix} \right). \tag{4.34}$$

Using Cauchy-Schwarz (theorem 18) we obtain

$$a(U,\phi) \leq \frac{\beta}{2} \left(2\|U_{1:2}\|_{\mathrm{L}^2}^2 + \|U_3\|_{\mathrm{L}^2}^2 + 2\|\nabla U_3\|_{\mathrm{L}^2}^2 \right)^{1/2} \left(2\|\phi_{1:2}\|_{\mathrm{L}^2}^2 + \|\phi_3\|_{\mathrm{L}^2}^2 + 2\|\nabla\phi_3\|_{\mathrm{L}^2}^2 \right)^{1/2}. \tag{4.35}$$

Because the square root is monotonous, we can also write

$$a(U,\phi) \leq \beta \left(\|U_{1:2}\|_{\mathrm{L}^2}^2 + \|U_3\|_{\mathrm{L}^2}^2 + \|\nabla U_3\|_{\mathrm{L}^2}^2 \right)^{1/2} \left(\|\phi_{1:2}\|_{\mathrm{L}^2}^2 + \|\phi_3\|_{\mathrm{L}^2}^2 + \|\nabla\phi_3\|_{\mathrm{L}^2}^2 \right)^{1/2} \tag{4.36}$$

$$= \beta \|U\|_X \|\phi\|_X , \tag{4.37}$$

which is exactly the continuity constraint.

The next thing to show is the coercivity. At the beginning will use Young's inequality [30]

$$a \cdot b \leq \|a\| \|b\| \leq \frac{\gamma}{2} \|a\|^2 + \frac{1}{2\gamma} \|b\|^2 , \tag{4.38}$$

where a, b are in some normed space and $\gamma \in \mathbb{R}^+$. Using the Cauchy-Schwarz inequality and taking care regarding the sign of $\mu_p - 1$ yields

$$\Delta t(\mu_p - 1) U_{1:2} \cdot \nabla U_3 \geq -\Delta t \, |\mu_p - 1| \, \|U_{1:2}\|_{\mathrm{L}^2} \|\nabla U_3\|_{\mathrm{L}^2} . \tag{4.39}$$

We can now use the above mentioned Young's inequality to obtain

$$\Delta t(\mu_p - 1) U_{1:2} \cdot \nabla U_3 \geq -\Delta t \, |\mu_p - 1| \, \big(\frac{\gamma}{2} \|U_{1:2}\|_{\mathrm{L}^2}^2 + \frac{1}{2\gamma} \|\nabla U_3\|_{\mathrm{L}^2}^2 \big), \tag{4.40}$$

with some $\gamma \in \mathbb{R}^+$. Plugging this inequality into the formula for a results in

$$a(U,U) \geq \left(1 + \frac{\Delta t}{\lambda} \right) \|U_{1:2}\|_{\mathrm{L}^2}^2 + \|U_3\|_{\mathrm{L}^2}^2 + \Delta t \mu_s \|\nabla U_3\|_{\mathrm{L}^2}^2 \tag{4.41}$$

$$- \Delta t \, |\mu_p - 1| \left(\frac{\gamma}{2} \|U_{1:2}\|_{\mathrm{L}^2} + \frac{1}{2\gamma} \|\nabla U_3\|_{\mathrm{L}^2}^2 \right). \tag{4.42}$$

If we collect the terms with the same factors we obtain

$$a(U,U) \geq \left(1 + \frac{\Delta t}{\lambda} - \Delta t \left|\mu_p - 1\right| \frac{\gamma}{2}\right) \|U_{1:2}\|^2_{\mathrm{L}^2} + \|U_3\|^2_{\mathrm{L}^2} + \left(\Delta t \mu_s - \frac{\Delta t \left|\mu_p - 1\right|}{2\gamma}\right) \|\nabla U_3\|^2_{\mathrm{L}^2}. \tag{4.43}$$

We want to choose γ such that

$$1 + \frac{\Delta t}{\lambda} - \Delta t \left|\mu_p - 1\right| \frac{\gamma}{2} > 0 \quad \text{and} \quad \Delta t \mu_s - \frac{\Delta t \left|\mu_p - 1\right|}{2\gamma} > 0. \tag{4.44}$$

It is also possible to express the conditions on γ as

$$\frac{\left|\mu_p - 1\right|}{2\mu_s} < \gamma < 2 \frac{1 + \frac{\Delta t}{\lambda}}{\Delta t \left|\mu_p - 1\right|}. \tag{4.45}$$

To know if such a γ exists we check if

$$\frac{\left|\mu_p - 1\right|}{2\mu_s} < 2 \frac{1 + \frac{\Delta t}{\lambda}}{\Delta t \left|\mu_p - 1\right|} \tag{4.46}$$

holds. This can also be written as

$$\mu_s \left(\frac{1}{\Delta t} + \frac{1}{\lambda}\right) > \frac{1}{4}(\mu_p - 1)^2. \tag{4.47}$$

But this is exactly one of the requirements of the theorem. We can define

$$\alpha := \min(1 + \frac{\Delta t}{\lambda} - \Delta t \left|\mu_p - 1\right| \frac{\gamma}{2}, 1, \Delta t \mu_s - \frac{\Delta t \left|\mu_p - 1\right|}{2\gamma}). \tag{4.48}$$

Using the new variable yields

$$a(U,U) \geq \alpha(\|U\|^2_{\mathrm{L}^2} + \|\nabla U_3\|^2_{\mathrm{L}^2}) = \alpha \|U\|_X. \tag{4.49}$$

Continuity and coercitivity implies the statement of the theorem as mentioned at the beginning of the proof. □

The boundary conditions were not necessary in the proof because we have a unique continuation from the previous time step for a given initial condition. The condition

$$\mu_s \left(\frac{1}{\Delta t} + \frac{1}{\lambda}\right) > \frac{1}{4}(\mu_p - 1)^2. \tag{4.50}$$

needs some discussion. Does it not impose a strong restriction? Consider a given set of λ, μ_s and μ_p. We can always find a Δt small enough that the inequality holds. To put it in terms of numerics, we find that we can only guarantee the existence and uniqueness of the solution if the time step size is small enough. Such a constraint reminds one of the CFL condition for Finite Volume schemes. However in this case the CFL condition dictates a relationship between spatial and time discretization for an explicit time stepping scheme. In the case of the last theorem it is not dependent on the spatial discretization and we use an implicit scheme. That we obtain some sort of condition on Δt however, should be expected because the Navier-Stokes equations have a hyperbolic part, which needs some consideration for Finite Element Methods. More information on the CFL condition can be found in the original German paper [7].

4.2 Numerical results

Now we have everything we need to put our equations into a simulation. We will first introduce the framework, which we use to handle most of the calculations performed for this work. After that we will perform an empirical convergence study and discuss some test cases and limitations of this approach.

4.2.1 Software

Most of the simulations conducted for this work used the FEniCS framework ([2], [19]). Other components of this project are the C++ backend DOLFIN ([21], [22]), the compiler for the variational form FFC ([17], [20], [24]), the language definition for variational forms UFL([5], [1]) and the corresponding code generator UFC ([4], [3]).

The advantages of using a framework at all are obvious. The less time one spends coding, the more one can assign to theoretical tasks. Another nice property is that the whole debugging process is shortened by a large bit. The software is also throughly tested so bugs are much less frequent. The responsibility of the user is to implement all properties of the finite element approach, one wants to use. This includes the finite element spaces, the weak formulation and the time discretization. Literature on how to get started with FEniCS is easy to find. A nice tutorial is given by the founders of FEniCS in [18]. The code, that has been written for this work, can be found on the CD, which accompanies this.

4.2.2 Simulation cases

The next subsections will deal with simulations and their results. This subsection should give a quick overview of all the relevant cases studied.

Two domains were used throughout the simulations. They can be seen in figures 4.1 and 4.2. The boundary of the unit disc is approximated using a polygonal chain. It is interesting to observe that even for the square the mesh generator returns an unstructured mesh. We used the unit disc for the following simulation cases:

- Convergence of our implementation
- Startup flow (fluid at rest in the beginning but perpendicular pressure)

The square had applications in these cases:

- Startup flow
- “ideal” rheometer cross-section

The reason why we simulated the startup flow case twice will be obvious once we compare the results. Videos of these testcases will be available on the CD that accompanies this work.

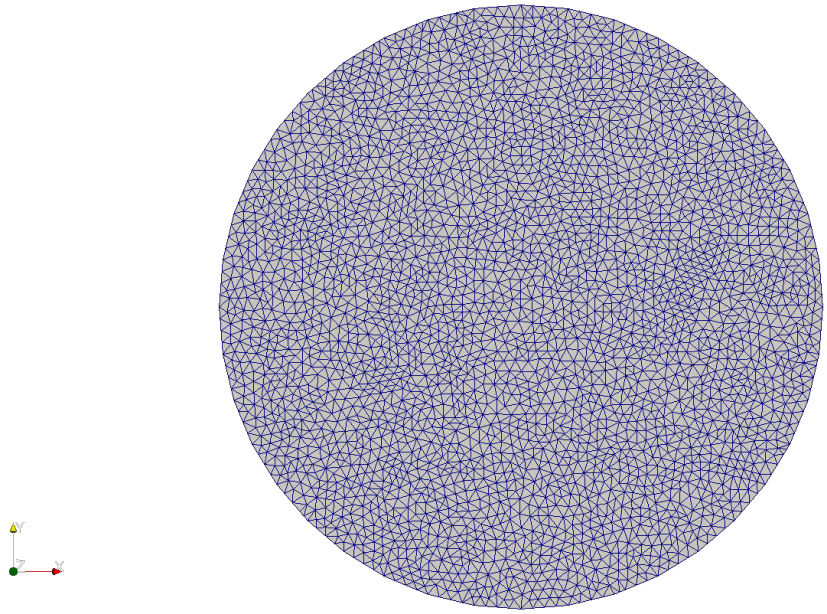

Figure 4.1: Finest mesh of the unit disc

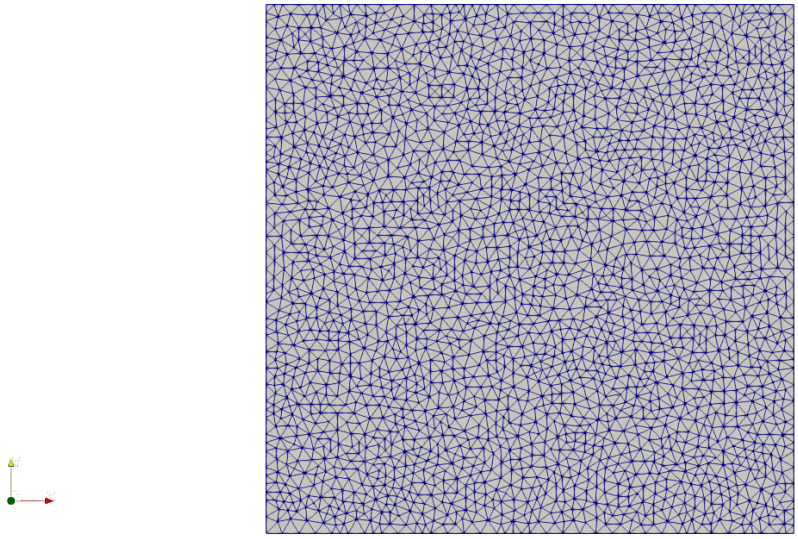

Figure 4.2: Mesh of a square with sidelength 2

4.2.3 Empirical convergence

In this section we will use a setup to study the empirical convergence of the implementation and its order. We will use the unit disc as a domain that is

$$\Omega = \{x \in \mathbb{R}^2 | \, |x| \leq 1\}. \tag{4.51}$$

As a boundary condition, we require

$$u = 0 \quad \forall \boldsymbol{x} \in \partial\Omega. \tag{4.52}$$

We want to study empirical convergence for the MAXWELL model ($\mu_s = 0$), because it is more challenging due to the missing dissipation term. We set the initial conditions

$$u(t=0) = e^{-\|x\|_2 - 1} - 1, \quad \boldsymbol{C}(t=0) = -2\boldsymbol{x}(e^{-(\|x\|_2^2 - 1)} - 1) \tag{4.53}$$

where $\boldsymbol{x} \in \Omega$. This initial condition is obviously continuous at the boundary. Because the exact solution of this equation is not known, we will use the concept of *manufactured solution*. The first step in the process of creating a manufactured solution is to set a function, that we want the exact solution to be. For simplicity, we want that the solution stays constant in time. We could have introduced a time dependent solution here, but it only makes calculations more complex. Next, we plug the solution functions into our equations and add a residual $\boldsymbol{R}\colon \Omega \to \mathbb{R}^3$ on the right side. It does not matter if we use the classic or weak formulation of our problem. By using the classic formulation we obtain

$$0 = -\partial_3 p - 2\mu_p \nabla \cdot [\boldsymbol{x}(e^{-(\|x\|_2^2 - 1)} - 1)] + \boldsymbol{R}_1, \tag{4.54}$$

$$-\frac{2}{\lambda}\boldsymbol{x}(e^{-(\|x\|_2^2 - 1)} - 1) = \nabla(e^{-(\|x\|_2^2 - 1)} - 1) + \boldsymbol{R}_{2:3}. \tag{4.55}$$

Note that the time derivatives evaluates to zero in both cases. Next, we solve this set of equations for $\boldsymbol{R}$. To do that we expand the divergence and the gradient operators to obtain

$$\boldsymbol{R}_1 = \partial_3 p + 4\mu_p(e^{-(\|x\|_2^2 - 1)}(1 - \|x\|_2^2) - 1), \tag{4.56}$$

$$\boldsymbol{R}_{2:3} = 2\boldsymbol{x}\left(e^{-(\|x\|_2^2 - 1)}\left(1 - \frac{1}{\lambda}\right) + \frac{1}{\lambda}\right). \tag{4.57}$$

To solve the equations, we have to add the residual to the weak formulation. This results in

$$\int_\Omega (u - u^n)\psi \, \mathrm{d}\Omega + \Delta t \left(\int_\Omega \partial_3 p \psi \, \mathrm{d}\Omega + \int_\Omega \boldsymbol{s} \cdot \nabla\psi \, \mathrm{d}\Omega - \int_\Omega \boldsymbol{R}_1 \psi \, \mathrm{d}\Omega\right) = 0, \tag{4.58}$$

$$\boldsymbol{s} = \mu_p \boldsymbol{C}_{1/\lambda}, \tag{4.59}$$

$$\int_\Omega (\boldsymbol{C}_{1/\lambda} - \boldsymbol{C}_{1/\lambda}^n + \frac{\Delta t}{\lambda}\boldsymbol{C}_{1/\lambda}) \cdot \varphi \, \mathrm{d}\Omega = \Delta t \int_\Omega \nabla u \cdot \varphi \, \mathrm{d}\Omega + \Delta t \int_\Omega \boldsymbol{R}_{2:3} \cdot \varphi \, \mathrm{d}\Omega. \tag{4.60}$$

≈ # Elements per diameter	L^2-error	EOC in L^2	L^∞-error	EOC in L^∞
20	0.0797652	-	0.877305	-
40	0.0233543	1.77207	0.195846	2.16336
80	0.00615334	1.92425	0.100548	0.96184
160	0.00163421	1.91278	0.0369229	1.44529
320	0.000433982	1.91288	0.0198643	0.89434

Table 4.1: Convergence of the MAXWELL model

The solution of this system is exactly our initial condition. This way we can easily calculate approximation errors.

We used $\Delta t = 0.1$ and calculated until $T = 1$. As the polymer viscosity we used 1. Table 4.1 shows the L^2 and L^∞ error and the experimental order of convergence for both errors. We can observe that the L^2 error is well behaved and results in an EOC, which approaches 2. Unfortunately the error in the L^∞ does not return something similar. The important thing to note is that the L^∞ error is monotonously decreasing. It is well-known that it is very difficult to obtain a smooth convergence in L^∞. It is typical for these kind of simulations that the L^∞ error is worse than L^2. Also, remember that the boundary of the domain has been approximated by a polygonal chain, which increases proportionally with the number of cells. The order of the geometry approximation influences the order of convergence just like the spatial or time discretization. This might also explain, why in the beginning the EOC is a bit far from the expected 2. As we have 320 elements per diameter we also approximate the circle with 320 segments in the polygonal chain which, to a naked eye, is indistinguishable from a true circle. The time integration order of one is also negatively influencing the EOC.

4.2.4 Startup flow on unit disc

We will now take a look at a more physically relevant example. Consider a fluid at rest that is subject to a perpendicular pressure gradient. This is basically the setup for the next two simulation cases. However, we will choose different domains and show that this leads to very different results. The setup is shown in figure 4.3. We chose the no slip boundary condition $u = 0$ on the boundary of the circle. We assumed a pressure gradient of $\partial_3 p = -5$ and a relaxation time of $\lambda = 1$. Figure 4.4 shows the velocity at the center of the circle. We used the MAXWELL model this time with $\mu_p = 1$. The typical non-NEWTONIAN oscillations can be seen. We calculated until $T = 12$ with $\Delta t = 0.001$. The graph for a NEWTONIAN fluid would be monotonously increasing to the steady state velocity.

4.2.5 Startup flow on square

Now we want to repeat this calculation but with a different domain. We will see that this will impact the result. The setup is shown in figure 4.5. A related problem is solved in [8]. Therefore, we choose periodic boundary conditions on the left and right sides

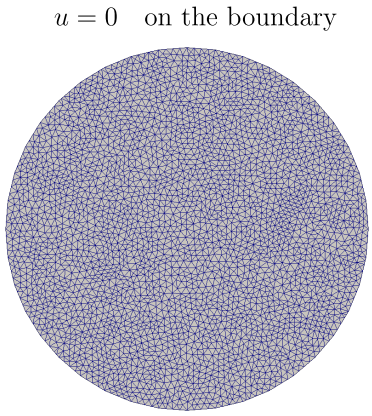

Figure 4.3: Setup startup flow unit disc

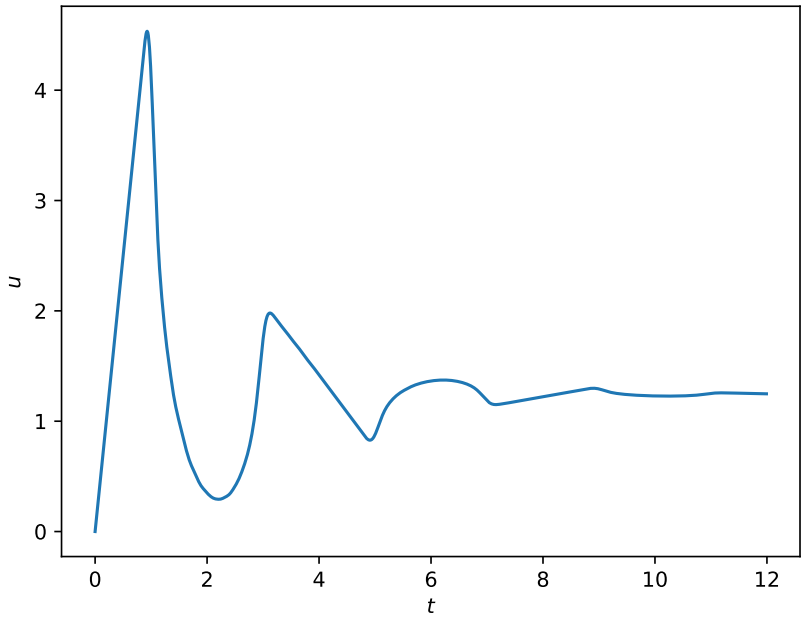

Figure 4.4: Centerline velocity on the unit disc domain with $\mu_s = 0$, $\mu_p = 1$, $\partial_3 p = -5$ and $\lambda = 1$

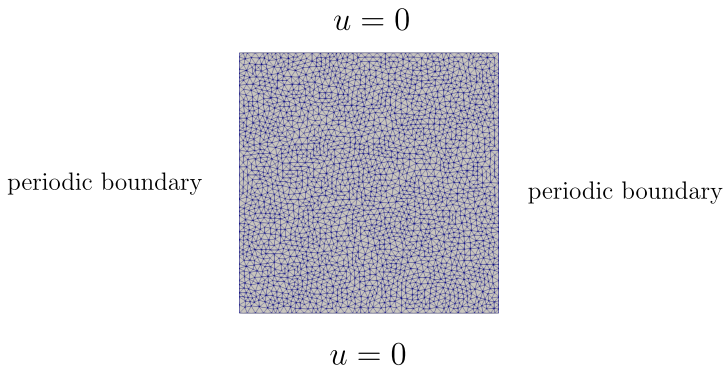

Figure 4.5: Setup of the startup flow simulation on the square domain

and a no slip boundary condition $u = 0$ at the top and bottom. We chose $\partial_3 p = -3$ to obtain the results from [8]. However, the magnitude of the pressure only affects the amplitude and not the behavior of the fluid. As one can see in figure 4.6 we obtain a very different behavior by changing the domain. This is also a difference to a NEWTONIAN fluid, where we would have gotten the same graph for both domains. It also seems like every extrema can be found at a multiple of the relaxation time and that these extrema are not differentiable. If we interpret the relaxation time of a NEWTONIAN fluid to be 0 we can explain the absence of maximal and minimal points in their velocity graph. Shocks cannot be found because we solve a linear system of equations where these cannot appear.

4.2.6 "Ideal" rheometer

A rheometer is a laboratory device used by rheologists to measure the behaviour of a material to applied forces. It contains a container for the material and a whisk which mixes through it. Properties like viscosity and relaxation times can be studied. To obtain a two-dimensional problem, this case represents a slice from top to bottom of an infinitively high rheometer. The setup for this case is shown in figure 4.7. The left side of the square has the no slip boundary condition $u = 0$. The top and bottom side are coupled with a periodic boundary condition. The right side has the DIRICHLET boundary condition $u = 10$.

The centerline velocity of this simulation can be seen in figure 4.8. We chose the OLDROYD-B model with $\mu_s = 0.1$ and $\mu_p = 0.9$. Our relaxation time was set to $\lambda = 2$ and we calculated until $T = 4$ with a step size of $\Delta t = 0.001$. Like before one can clearly see that we do not have a fluid that is showing NEWTONIAN behavior. This is the first case, where we have set $\mu_s > 0$. This can be clearly observed in figure 4.8 because the resulting graph looks like a differentiable function. This is typical for advective problems with dissipation from a second order derivative term.

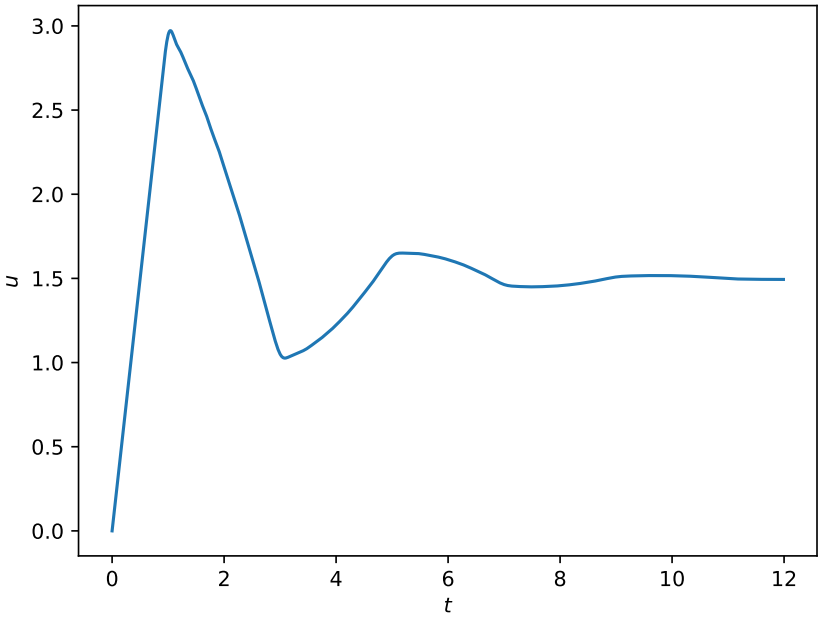

Figure 4.6: Centerline velocity of the start-up flow on the square domain $\mu_s = 0$, $\mu_p = 1$, $\partial_3 p = -3$ and $\lambda = 1$

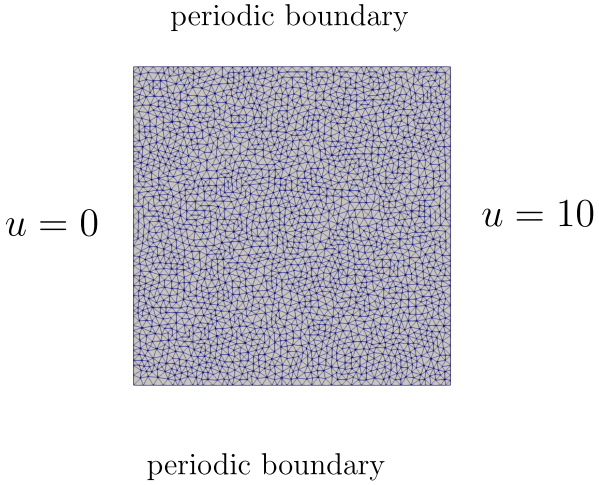

Figure 4.7: Setup for the rheometer simulation case

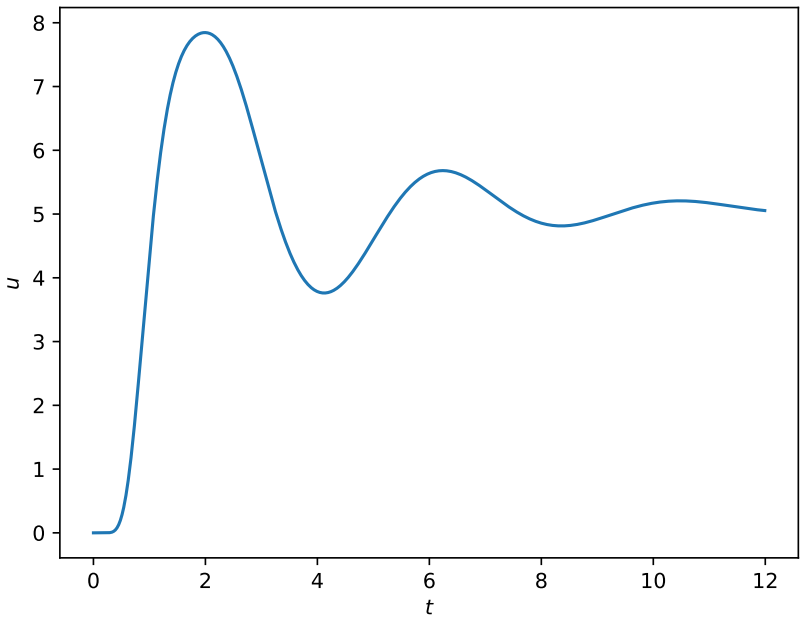

Figure 4.8: Centerline velocity in the rheometer with $\mu_s = 0.1$, $\mu_p = 0.9$, $\lambda = 2$

5 Different approaches for the integral equation

In chapter 3 we could eliminate the integral from equation (3.22) completely by taking sensible assumptions on the memory function G. However, this is a very specific case and the more complex physical models do not satisfy these assumptions. In this chapter we will discuss possibilities for a more general approach.

5.1 Combination of relaxation times

In physics, even for a complex relaxation modulus G it is sufficient most of the time to model the fluid's behavior using a combination of relaxation times. So for $N \in \mathbb{N}$ we set

$$G(\tau) = \sum_{n=1}^{N} \mu_p^{(n)} e^{-\tau/\lambda_n}, \tag{5.1}$$

where $\mu_p^{(n)}$ and λ_n are polymer viscosities and relaxation times, respectively. One can think of this process as descretizing the continuous spectrum of the fluid's relaxation times dictated by G. We will now deduce a new formula for $\boldsymbol{s}$ by inserting this assumption. This yields

$$\boldsymbol{s}(t) = \int_0^t \partial_\tau \boldsymbol{b}(t, t-\tau) \sum_{n=1}^{N} \mu_p^{(n)} e^{-\tau/\lambda_n} \, \mathrm{d}\tau. \tag{5.2}$$

Because all $\mu_p^{(n)}$ are constant and the integral is linear, we can rewrite this to read

$$\boldsymbol{s}(t) = \sum_{n=1}^{N} \mu_p^{(n)} \int_0^t \partial_\tau \boldsymbol{b}(t, t-\tau) e^{-\tau/\lambda_n} \, \mathrm{d}\tau. \tag{5.3}$$

We will now follow a similar pattern as we did in chapter 3. We will use integration by parts. It results in

$$\boldsymbol{s}(t) = \sum_{n=1}^{N} \left(\mu_p^{(n)} \boldsymbol{b}(t,0) e^{-t/\lambda_n} - \mu_p^{(n)} \boldsymbol{b}(t,t) + \frac{\mu_p^{(n)}}{\lambda_n} \int_0^t \boldsymbol{b}(t, t-\tau) e^{-\tau/\lambda_n} \, \mathrm{d}\tau \right). \tag{5.4}$$

We can now use the initial condition for $\boldsymbol{b}$ to eliminate some terms. The integral can also be extended to ∞ without changing the value. We obtain

$$\boldsymbol{s}(t) = \sum_{n=1}^{N} \frac{\mu_p^{(n)}}{\lambda_n} \int_0^\infty \boldsymbol{b}(t, t-\tau) e^{-\tau/\lambda_n} \, \mathrm{d}\tau. \tag{5.5}$$

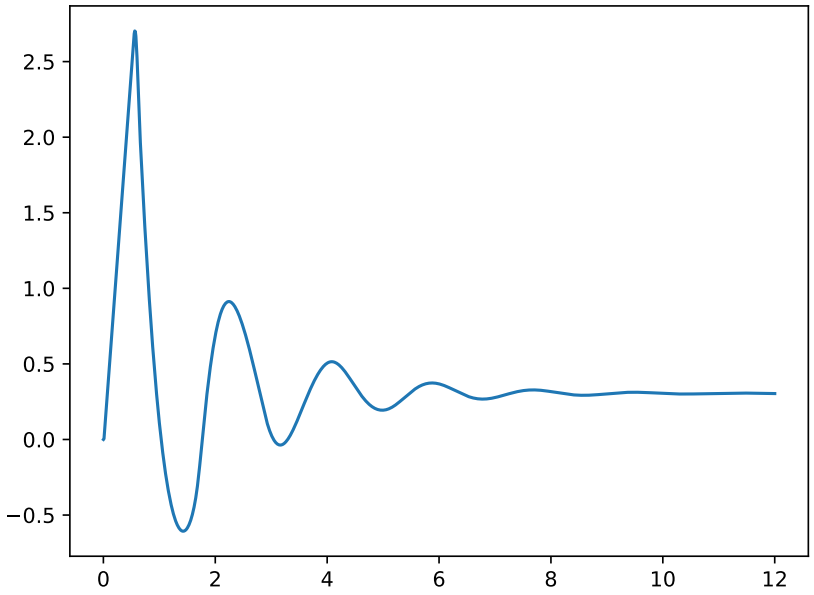

Figure 5.1: Centerline velocity graph for the unit disc domain and relaxation times $0.1, 1$ and 3 with $\mu_p^{(n)} \equiv 1$

This integral is exactly the LAPLACE transform of $\boldsymbol{b}$. Therefore, the equation for $\boldsymbol{s}$ becomes

$$\boldsymbol{s}(t) = \sum_{n=1}^{N} \frac{\mu_p^{(n)}}{\lambda_n} L_{\boldsymbol{b}}(t, 1/\lambda_n) = \sum_{n=1}^{N} \mu_p^{(n)} \boldsymbol{C}_{1/\lambda_n}. \tag{5.6}$$

If we look closely at the equation above one can see that we now have N equations for $\boldsymbol{C}$ to simulate. So to calculate $\boldsymbol{s}$ we have to simulate

$$\partial_t \boldsymbol{C}_{1/\lambda_n} + \frac{1}{\lambda_n} \boldsymbol{C}_{1/\lambda_n} = \nabla u \tag{5.7}$$

for all n. Because G is just a linear combination, all proofs and statements we made about the simpler equation remain true for this one. The discretization can be handled as before. However, we obtain n additional equations and $2n$ additional variables for n relaxation times . This is not ideal from an efficiency point of view. A simulation on the before mentioned unit disc mesh with $\mu_p^{(n)} \equiv 1$ and $\lambda_1 = 0.1, \quad \lambda_2 = 1, \quad \lambda_3 = 3$ resulted in 5.1. This figure shows the centerline velocity over time for a startup flow as in the previous examples. For the pressure we used $\partial_3 p = -5$. The computation time was significantly longer than in the earlier examples. This highlights the necessity for a performance discussion, which is out of scope for this work. One can also observe that the extrema have now moved to approximately multiples of the geometric mean of the relaxation times.

5.2 A reduced integral equation

In this section we will take a look at a related integral equation. Later on the solution of this equation can be interpreted as the square root of the relaxation modulus G. More information on the relationship of this equation and non-NEWTONIAN fluids can be found in [11]. The new equation has the same complexity regarding the integral as the ones we were looking at earlier, but the complexity regarding dimension and number of equations is much reduced. This makes it easier to concentrate on the main problem and to gather ideas, how one may approach the problem of an integro-dfferential equation in general. We use

$$\dot{\phi}(t) = -\phi(t) - \int_0^t m(t-\tau)\dot{\phi}(\tau)\,\mathrm{d}\tau, \tag{5.8}$$

$$m(t) = v_1\phi(t) + v_2\phi(t)^2, \tag{5.9}$$

where we use the dot notation for the derivative and v_1 and v_2 are given constants. This equation is also studied in [12]. The integral is the convolution of m and $\dot{\phi}$. The convolution is symmetric and because $m(\tau)$ is independent of t the derivative can be applied to the whole product instead of just ϕ. This is allowed, because if ϕ is differentiable then, by definition, m is as well. Using these ideas we get

$$\dot{\phi}(t) = -\phi(t) - \int_0^t \frac{\mathrm{d}}{\mathrm{d}t}(m(\tau)\phi(t-\tau))\,\mathrm{d}\tau. \tag{5.10}$$

If we apply the inverse Leibniz rule for parameter integrals, we can swap the integral and differential to obtain

$$\dot{\phi}(t) = -\phi(t) - \frac{\mathrm{d}}{\mathrm{d}t}\int_0^t m(\tau)\phi(t-\tau)\,\mathrm{d}\tau + m(t)\phi(0) \tag{5.11}$$

We can now integrate over t to eliminate all derivatives. This leads to

$$\phi(t) = \phi(0) - \int_0^t m(t')\phi(t-t')\,\mathrm{d}t' + \int_0^t m(t')\phi(0)\,\mathrm{d}t' - \int_0^t \phi(t')\,\mathrm{d}t', \tag{5.12}$$

where we renamed the integration variable to t' to be able to shorten the expression. We obtain

$$\phi(t) = \phi(0) + \int_0^t [m(t')(\phi(0) - \phi(t-t')) - \phi(t')]\,\mathrm{d}t'. \tag{5.13}$$

Now we have a pure integral equation. The next step will be to discretize the integral and t. We will use timesteps $0 = t_0, \ldots, t_N = T$, where T is the end time. The timestep length is given by $h_j = t_{j+1} - t_j$, where h_j does not have to be constant. For the integral we will use

$$\int_0^{t_n} f(t)\,\mathrm{d}t = \sum_{j=0}^{n-1}\int_{t_j}^{t_{j+1}} f(t)\,\mathrm{d}t \tag{5.14}$$

for a generic f. We will denote $f_{n+1} := f(t_{n+1})$. Now, we have to choose a method to calculate the integral in (5.14). By using a simple quadrature rule and $h_j \equiv \Delta t$, we obtain

$$\phi_{n+1} = \phi_0 + \sum_{j=0}^{n} \left(\Delta t \, m_j \left(\phi_0 - \phi_{n+1-j}\right) - \phi_j\right). \tag{5.15}$$

We call this algorithm (A1). One can observe that to calculate timestep t_n from t_{n-1} we need computational effort $\mathcal{O}(n)$. So to calculate t_N from the start t_0 results in effort

$$\sum_{n=1}^{N} n = \frac{N(N+1)}{2}. \tag{5.16}$$

So it results that the calculation of t_N from the initial condition results in $\mathcal{O}(N^2)$ effort. But that is very inefficient so we will try to find a better method. However, we implemented this brute force approach to obtain a reference for the next algorithms. The results for (A1) are shown e.g in figure 5.2.

5.2.1 Exponentially increasing time intervals

We try to employ, that the resulting solution ϕ is exponentially decaying. Therefore it makes sense that we use intervals that increase their size exponentially. We hope that this approach is successful because a similar trick is done in calculating the Fast-Fourier-Transformation. For more on this topic see [6]. So we set $t_j = h2^j$ and choose

$$\sum_{j=0}^{n-1} \int_{t_j}^{t_{j+1}} f(t)\,\mathrm{d}t \approx \sum_{j=0}^{n-1} (t_{j+1} - t_j) f_{j+1}. \tag{5.17}$$

So we approximate the integral with the rectangle formed by the value on the right side and the interval length. This underestimates the true value because the function is monotonously decreasing. Replacing the integral in (5.13) yields

$$\phi_n = \phi_0 + \sum_{j=0}^{n-1} h_j [m_{j+1}(\phi_0 - \phi(t_n - t_{j+1})) - \phi_{j+1}] \tag{5.18}$$

If we take the same equation for ϕ_{n+1} and subtract (5.18) we obtain

$$\phi_{n+1} = \phi_n - h_n \phi_{n+1} + \sum_{j=0}^{n-1} h_j m_{j+1} (\phi(t_n - t_{j+1}) - \phi(t_{n+1} - t_{j+1})) \tag{5.19}$$

We run into a problem here because of the terms $\phi(t_n - t_{j+1})$ and $\phi(t_{n+1} - t_{j+1})$. Because the difference in the argument does not necessarily correspond to a point in time we already calculated, we have to interpolate this value. To choose sensible interpolation points, we check in which interval the differences would lie in. It yields

$$t_n = h2^n \geq t_n - t_{j+1} = h(2^n - 2^{j+1}) \geq h(2^n - 2^{n-1}) = h2^{n-1} = t_{n-1}, \tag{5.20}$$

where $0 \leq j \leq n-2$. The case where $j = n-1$ is trivial and we will consider this term separately later on. This means that the first term lies completely in the already known interval from the previous timestep. We use $\phi(t_n - t_{j+1}) \approx \phi_n$ as an interpolation for all $0 \leq j \leq n-2$. If we conduct the same calculations for $t_{n+1} - t_{j+1}$, we get

$$t_{n+1} = h2^{n+1} \geq t_{n+1} - t_{j+1} \geq h(2^{n+1} - 2^{n-1}) = h\frac{3}{2}2^n = \frac{3}{2}t_n, \tag{5.21}$$

where $0 \leq j \leq n-2$ as before. This however presents the challenge, that we have to interpolate in the yet unknown interval $[t_{n+1}, 3/2t_n]$. As we want to obtain an implicit method we use t_{n+1}. Putting the assumptions for the interpolation into (5.19) we obtain

$$\phi_{n+1}(1 + h_n + \sum_{j=0}^{n-2} h_j m_{j+1}) = \phi_n(1 + \sum_{j=0}^{n-2} h_j m_{j+1}) + h_{n-1}m_n(\phi_0 - \phi_n). \tag{5.22}$$

If we now define $s_n := \sum_{j=0}^{n-2} h_j m_{j+1}$, we can rewrite everything as

$$\phi_{n+1}(1 + h_n + s_n) = \phi_n(1 + s_n - h_{n-1}m_n) + h_{n-1}m_n\phi_0, \tag{5.23}$$

$$s_{n+1} = s_n + h_{n-1}m_n. \tag{5.24}$$

We define $s_0 = 0$ as the initial condition of s. To obtain greater accuracy we should avoid the division of large numbers. Therefore we multiply the equation with 2^{-n}. This results in

$$\phi_{n+1}(2^{-n} + h + 2^{-n}s_n) = \phi_n(2^{-n} + 2^{-n}s_n - \frac{h}{2}m_n) + \frac{h}{2}m_n\phi_0, \tag{5.25}$$

$$s_n = s_n + h_{n-1}m_n. \tag{5.26}$$

We introduce $\beta_n = 2^{-n}s_n$. We obtain

$$\phi_{n+1}(2^{-n} + h + \beta_n) = \phi_n(2^{-n} + \beta - \frac{h}{2}m_n) + \frac{h}{2}m_n\phi_0, \tag{5.27}$$

$$\beta_{n+1} = 2^{-n-1}s_{n+1} = \frac{1}{2}(\beta_n + \frac{h}{2}m_n). \tag{5.28}$$

This is now an explicit algorithm, called (A2) with $\mathcal{O}(N)$ to solve the equation for ϕ. However as one can see in figure 5.2, it does not converge to the correct solution. Moreover it is unclear if for $h \to 0$ the error would become better as it would lead to $T - t_{N-1} \to T$. This means that this method is not suitable for our problem and we need to find a new approach.

5.2.2 Transformation of the argument

The reason that the previous method did not work was probably due to the fact that we eliminated the effect of the history completely. So we need an algorithm with a runtime between $\mathcal{O}(n)$ and $\mathcal{O}(n^2)$. The next idea one could follow is to transform $t = f(u)$ and

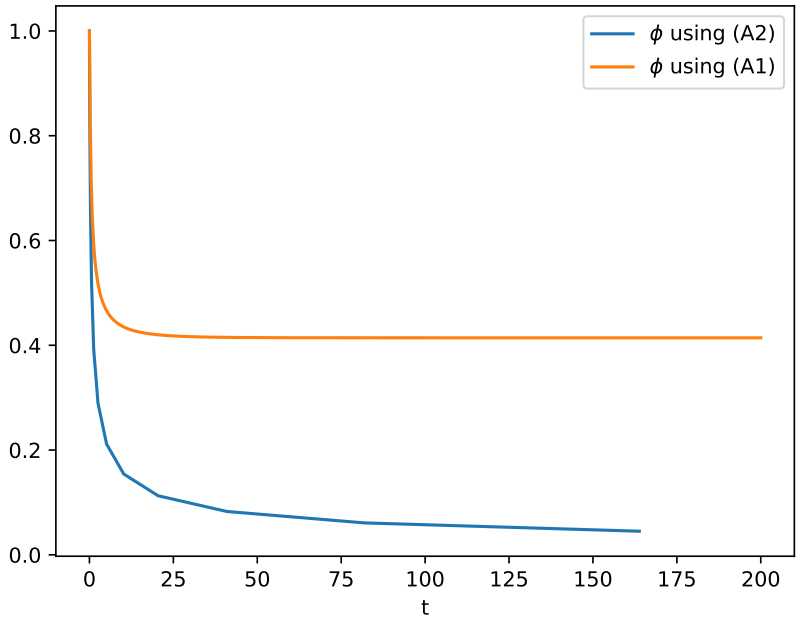

Figure 5.2: Both ϕ with $h = \Delta t = 0.01$ and $v_1 = 1.5,\ v_2 = 0.5$

$\tilde{\phi}(u) = \phi(f(u))$ (and m accordingly), where f is a diffeomorphism (bijective function with differentiable inverse). Let $u_{-1} = f^{-1}(0)$, which is unique because f is especially injective. If we recall (5.13) and insert the transformation for t, we obtain

$$\phi(f(u)) = \phi(f(u_{-1})) + \int_{u_{-1}}^{u} f'(u')[m(f(u'))(\phi(f(u_{-1})) - \phi(f(u) - f(u'))) - \phi(f(u'))]\,\mathrm{d}u'. \tag{5.29}$$

Now we can replace ϕ by $\tilde{\phi}$. The transformed equation reads

$$\tilde{\phi}(u) = \tilde{\phi}(u_{-1}) + \int_{u_{-1}}^{u} f'(u')[\tilde{m}(u')(\tilde{\phi}(u_{-1}) - \tilde{\phi}(f^{-1}(f(u) - f(u')))) - \tilde{\phi}(u')]\,\mathrm{d}u'. \tag{5.30}$$

The goal of this transformation is that we can use equally spaced points in u and u' but do not have effort $\mathcal{O}(N^2)$. The important term is $\tilde{\phi}(f^{-1}(f(u) - f(u')))$ because this is the only time that old timesteps effect the current one. Ideally

$$F(u, u') := f^{-1}(f(u) - f(u')) \approx u, \quad u' \in (u_{-1}, u] \tag{5.31}$$

almost everywhere. This would again eliminate the history completely like in the previous example. In figure 5.3 $F(u')$ is shown for a fixed u and two choices of f. The points w_i^3 will be introduced later. We can see that for both functions f we have a long interval, in which $u = F(u, u')$ is a good approximation.

Now let $f(u) = e^u$. We choose this because $F(u, u')$ is qualitatively the same function for every u just translated. For $f(u) = \exp(u)$ we have

$$F(u, u') = \log(\exp(u) - \exp(u')). \tag{5.32}$$

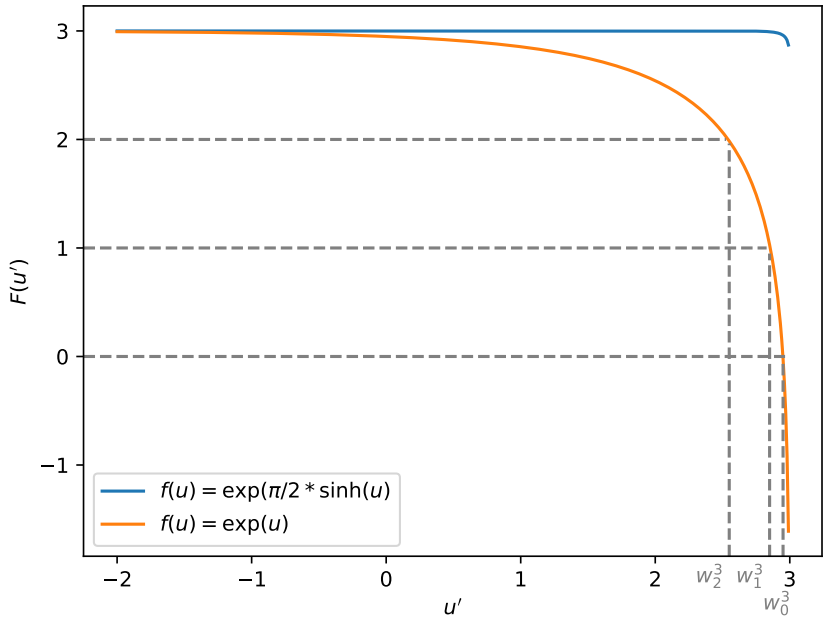

Figure 5.3: History dependency for different f with $u = 3$

This will help us immensely later on. It is true that $f(u) = \exp(\frac{\pi}{2}\sinh(u))$ is decreasing even faster, but because $F(u, u')$ is then qualitatively dependent on u it would introduce new effort to each timestep. We then have $u_{-1} = -\infty$. Obviously we cannot start our calculation at $-\infty$ so we introduce an arbitrary u_0 where $|f(u_0) - f(-\infty)|$ is negligible. This is possible, because $\exp(u) \to 0$ as $u \to -\infty$. From there we introduce a discretization for u as

$$u_i = u_0 + ih, \quad \forall i = 0, \dots, N \tag{5.33}$$

where $h = \frac{\log(T)-u_0}{N}$ is a fixed interval length and we stop at $u_N = \log(T)$, where T is the end time and N the given number of steps. Our goal now is to calculate values for $\tilde{\phi}(u_i)$ for all $i = 1, \dots, N$. We assume that for all $i = 0 \dots, N$ we have

$$\tilde{\phi}(u) \equiv \tilde{\phi}(u_i), \quad \forall u \in (u_{i-1}, u_i]. \tag{5.34}$$

This is where we introduce the discretization error. The evaluation of the integral will be exact under this assumption. Let us now take a look back at (5.30) and look at the integral part by part. Let us assume we know the values for each $\tilde{\phi}(u_i)$ for $i \leq n$ and we want to calculate $\tilde{\phi}(u_{n+1})$. For the first part we get

$$I_1 := \int_{-\infty}^{u_{n+1}} f'(u')\tilde{m}(u')\tilde{\phi}(-\infty)\,\mathrm{d}u' = \tilde{\phi}(-\infty)\sum_{i=0}^{n+1}\int_{u_{i-1}}^{u_i} f'(u')\tilde{m}(u')\,\mathrm{d}u' \tag{5.35}$$

Assuming $\tilde{\phi}$ to be piecewise constant on the integration interval, this also holds for $\tilde{m}$.

As such, jumps only occur on the integral boundary and we obtain

$$I_1 = \tilde{\phi}(-\infty) \sum_{i=0}^{n+1} \tilde{m}(u_i) \int_{u_{i-1}}^{u_i} f'(u') \, \mathrm{d}u' = \tilde{\phi}(-\infty) \sum_{i=0}^{n+1} \tilde{m}(u_i)[f(u_i) - f(u_{i-1})]. \tag{5.36}$$

We set

$$I_1 = \tilde{\phi}(-\infty)[S(n) + \tilde{m}(u_{n+1})[f(u_{n+1}) - f(u_n)]]. \tag{5.37}$$

This way we can perform an update for $S(n+1) = S(n) + \tilde{m}(u_{n+1})[f(u_{n+1}) - f(u_n)]$ and only use linear effort for this part. The summand for $n+1$ will have to be regarded separately because it is implicitly depending on the value of $\tilde{\phi}$ we are trying to calculate. We will actually discover later on that we can at worst obtain a cubic problem in $\tilde{\phi}(u_{n+1})$, which we will solve using NEWTON'S method.

After we now handled the first part of the integral successfully we will take a look at the third part because the second will be the most challenging. Consider

$$I_3 := -\int_{u_{-1}}^{u_{n+1}} f'(u') \tilde{\phi}(u') \, \mathrm{d}u' = -\sum_{i=0}^{n+1} \int_{u_{i-1}}^{u_i} f'(u') \tilde{\phi}(u') \, \mathrm{d}u'. \tag{5.38}$$

Using the same arguments as for I_1 we obtain

$$I_3 = -\sum_{i=0}^{n+1} \tilde{\phi}(u_i)[f(u_i) - f(u_{i-1})] = -M(n) - \tilde{\phi}(u_{n+1})[f(u_{n+1}) - f(u_n)], \tag{5.39}$$

with the update rule

$$M(n+1) = M(n) + \tilde{\phi}(u_{n+1})[f(u_{n+1}) - f(u_n)]. \tag{5.40}$$

To understand the second part, we will first rename the integration variable to avoid confusion and arrive at

$$I_2 := \int_{u_{-1}}^{u_{n+1}} f'(v) \tilde{m}(v) \tilde{\phi}(F(u_{n+1}, v)) \, \mathrm{d}v. \tag{5.41}$$

Now let us recall the behavior of $F(u, v)$ in figure 5.3. Heuristically, we can see that in the beginning we can set $\tilde{\phi}(F(u_{n+1}, v)) = \tilde{\phi}(u_{n+1})$. But what do we do with the remaining interval? To answer this question we observe that F is its own inverse in the second argument. Let $x = F(u, v)$. It yields

$$f(x) = f(u) - f(v) \quad \Rightarrow f(v) = f(u) - f(x) \quad \Rightarrow v = F(u, x). \tag{5.42}$$

The remaining discretization will be built on this fact. Specifically, we set $w_i^n = F(u_n, u_i)$. Some of these points are shown in figure 5.3. This ensures that

$$\tilde{\phi}(F(u_{n+1}, w_i^{n+1})) = \tilde{\phi}(u_j), \tag{5.43}$$

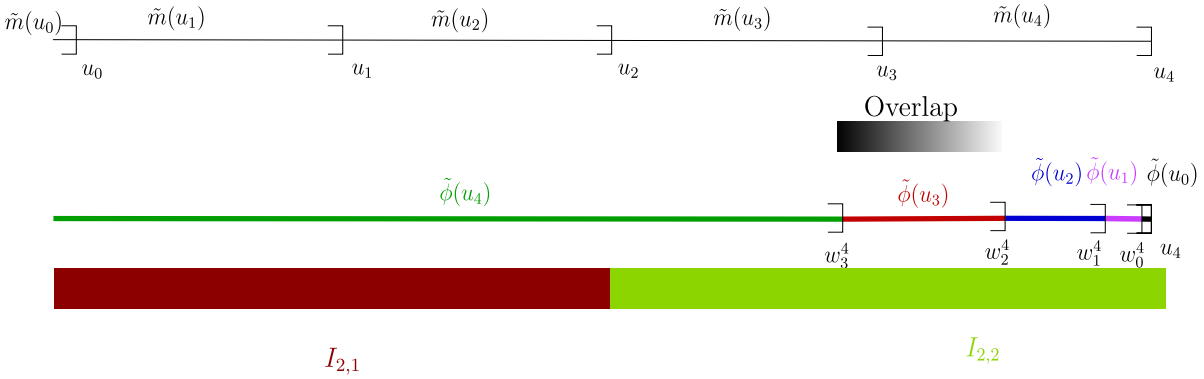

Figure 5.4: Visualization of the discretization for I_2, where $n = 3$

for some $j = 0, \ldots, n$. Note that the w_i^n are decreasing as i increases for a fixed n. The lowest value of w_i^n will be w_{n-1}^n. Our discretization I_2 will be

$$\{v_i\}_i = \text{sort}(\{u_i\}_i \cup \{w_i^{n+1}\}_i). \tag{5.44}$$

In other word, the sequence $\{v_i\}$ is the union of the sequences on the right hand side but sorted. The sequence $\{v_i\}$ will differ in each timestep. One might be tempted to think that this will at least result in an $\mathcal{O}(N^2)$ algorithm because we have to sort in every timestep, but we will solve this issue later on.

An important thing to know during the simulation is the first index j so that $v_j \in \{w_i^{n+1}\}_i$ for each n. Because we know that $v_j = w_n^{n+1}$ we can calculate this directly. We introduce the index

$$l^* := \left\lceil \frac{u_n - w_n^{n+1}}{h} \right\rceil. \tag{5.45}$$

This is constant in n because of our choice for f and equals the number of intervals, over which the w_i^{n+1} spread. Therefore we know that all intervals until u_{n-l^*+1} do not contain any points w_i^{n+1}. Figure 5.4 shows the discretization of I_2, where $n = 3$. Both lines represent $[u_0, u_4]$. The top line shows the equidistant discretization used for $\tilde{m}$ and the bottom line shows the discretization for $\tilde{\phi}(F(u_4, v))$ The value below the respective lines denote the discretization points. Above the respective lines one can see the values of the factors $\tilde{m}$ and $\tilde{\phi}(F(u_4, v))$ for that specific interval. In the case shown above we have $l^* = 2$. As a next step, we will divide the integral I_2 to become

$$I_2 = \int_{-\infty}^{u_{n-l^*+1}} f'(v)\tilde{m}(v)\tilde{\phi}(F(u_{n+1}, v))\,\mathrm{d}v + \int_{u_{n-l^*+1}}^{u_{n+1}} f'(v)\tilde{m}(v)\tilde{\phi}(F(u_{n+1}, v))\,\mathrm{d}v. \tag{5.46}$$

We see that

$$\tilde{\phi}(F(u_{n+1}, v)) = \tilde{\phi}(u_{n+1}) \tag{5.47}$$

in the first part of this integral. So we get

$$I_2 = \tilde{\phi}(u_{n+1}) \int_{-\infty}^{u_{n-l^*+1}} f'(v)\tilde{m}(v)\,\mathrm{d}v + \int_{u_{n-l^*+1}}^{u_{n+1}} f'(v)\tilde{m}(v)\tilde{\phi}(F(u_{n+1}, v))\,\mathrm{d}v. \tag{5.48}$$

By using similar arguments like we used to modify I_1 and I_3, we can simplify this to

$$I_2 = \tilde{\phi}(u_{n+1}) \sum_{i=0}^{n+1-l^*} \tilde{m}(u_i)[f(u_i) - f(u_{i-1})] + \int_{u_{n-l^*+1}}^{u_{n+1}} f'(v)\tilde{m}(v)\tilde{\phi}(F(u_{n+1}, v))\,\mathrm{d}v. \quad (5.49)$$

We will denote the remaining integral by $I_{2,2}$. Let us introduce j^* as the index of w_i^n so that $w_{l^*-i}^{n+1} \in [u_n, u_{n+1}], \quad \forall i \geq j^*$. This can also be calculated in the preprocessing of the calculation by using the problem: Find j^* such that

$$\frac{u_{l^*} - w_{l^*-j^*}^{l^*}}{h} \leq 1 \quad \text{and} \quad \frac{u_{l^*} - w_{l^*-j^*+1}^{l^*}}{h} > 1. \quad (5.50)$$

This is a well-defined problem because such j^* always exists at timestep l^*. We can now divide $I_{2,2}$ to obtain

$$I_{2,2} = \int_{u_{n-l^*+1}}^{w_{n-j^*}^{n+1}} f'(v)\tilde{m}(v)\tilde{\phi}(F(u_{n+1}, v))\,\mathrm{d}v + \int_{w_{n-j^*}^{n+1}}^{u_{n+1}} f'(v)\tilde{m}(v)\tilde{\phi}(F(u_{n+1}, v))\,\mathrm{d}v. \quad (5.51)$$

However, because the second integral lies completely within $[u_n, u_{n+1}]$ we get

$$I_{2,2} = \int_{u_{n-l^*+1}}^{w_{n-j^*}^{n+1}} f'(v)\tilde{m}(v)\tilde{\phi}(F(u_{n+1}, v))\,\mathrm{d}v \quad (5.52)$$

$$+ \tilde{m}(u_{n+1}) \sum_{i=0}^{n-j^*} \int_{w_{n-j^*-i}^{n+1}}^{w_{n-j^*-i-1}^{n+1}} f'(v)\tilde{\phi}(F(u_{n+1}, v))\,\mathrm{d}v. \quad (5.53)$$

We assumed that $\tilde{\phi}$ is constant on each interval. Therefore we want $\tilde{\phi}(F(u_{n+1}, v))$ to be $\tilde{\phi}(u_{n-j^*-i})$. We then obtain

$$I_{2,2} = \int_{u_{n-l^*+1}}^{w_{n-j^*}^{n+1}} f'(v)\tilde{m}(v)\tilde{\phi}(F(u_{n+1}, v))\,\mathrm{d}v \quad (5.54)$$

$$+ \tilde{m}(u_{n+1}) \sum_{i=0}^{n-j^*} \tilde{\phi}(u_{n-j^*-i})[f(w_{n-j^*-i-1}^{n+1}) - f(w_{n-j^*-i}^{n+1})] \quad (5.55)$$

If we recall the definition of w we can simplify the expression to

$$I_{2,2} = \int_{u_{n-l^*+1}}^{w_{n-j^*}^{n+1}} f'(v)\tilde{m}(v)\tilde{\phi}(F(u_{n+1}, v))\,\mathrm{d}v \quad (5.56)$$

$$+ \tilde{m}(u_{n+1}) \sum_{i=0}^{n-j^*} \tilde{\phi}(u_{n-j^*-i})[f(u_{n-j^*-i}) - f(u_{n-j^*-i-1})]. \quad (5.57)$$

By reordering the sum we obtain

$$I_{2,2} = \int_{u_{n-l^*+1}}^{w_{n-j^*}^{n+1}} f'(v)\tilde{m}(v)\tilde{\phi}(F(u_{n+1}, v))\,\mathrm{d}v + \tilde{m}(u_{n+1}) \sum_{i=0}^{n-j^*} \tilde{\phi}(u_i)[f(u_i) - f(u_{i-1})]. \quad (5.58)$$

We can directly identify how the integral over the interval $[u_{n+1-l^*}, w_n^{n+1}]$ should be calculated. We obtain

$$\int_{u_{n-l^*+1}}^{w_n^{n+1}} f'(v)\tilde{m}(v)\tilde{\phi}(F(u_{n+1},v))\,\mathrm{d}v = \tilde{\phi}(u_{n+1})\tilde{m}(u_{n-l^*+2})[f(w_n^{n+1}) - f(u_{n-l^*+1})]. \quad (5.59)$$

But it holds

$$f(w_n^{n+1}) = f(u_{n+1}) - f(u_n). \quad (5.60)$$

So it yields

$$\int_{u_{n-l^*+1}}^{w_n^{n+1}} f'(v)\tilde{m}(v)\tilde{\phi}(F(u_{n+1},v))\,\mathrm{d}v = \tilde{\phi}(u_{n+1})\tilde{m}(u_{n-l^*+2})[f(u_{n+1}) - f(u_n) - f(u_{n-l^*+1})]. \quad (5.61)$$

So what remains of the integral is the part, where the two discretizations overlap. We also have to discuss how we can eliminate the sorting process from the main loop as this would result in $\mathcal{O}(N^2\log(N))$ effort. We know that after calculating l^* time steps we have the first appearance of a w which is larger than u_0 and therefore needs consideration. But this is independent of n and can be calculated in advance as a result. The part of w which overlaps with the equidistant discretization is exactly the part that is not entirely in the interval $[u_n, u_{n+1}]$. So it is given by

$$w_{l^*-i}^{l^*}, \quad \forall i = 1,\dots,j^*. \quad (5.62)$$

Because the u_i are equally spaced, we know that

$$u_{n+1} = u_{l^*} + u_{n+1-l^*}. \quad (5.63)$$

We can therefore add u_{n+1-l^*} each timestep and obtain the current position of these ws. For each n we know that the points $u_{n+1-l^*},\dots,u_n$ overlap with the other discretization. By applying the shift backwards we find that the w as above overlap with

$$u_i, \quad \forall i = 1,\dots,l^*-1. \quad (5.64)$$

This way we can define the overlapping points

$$\tilde{u} = \bigcup_{i=1}^{l^*-1} u_i \cup \bigcup_{j=1}^{j^*} w_{l^*-j}^{l^*} \quad (5.65)$$

We can sort the vector $\tilde{u}$ in the preprocessing. If we keep the permutation array that resulted during the sorting process, we can later recall, from which discretization each point came. We introduce two new variables

$$q_i := \left|\{u_j \colon u_j \in \tilde{u} \text{ and } u_j \le \tilde{u}_i\}\right|, \quad y_i := \left|\{w_j^{l^*} : w_j^{l^*} \in \tilde{u} \text{ and } w_j^{l^*} \le \tilde{u}_i\}\right|. \quad (5.66)$$

Now we gathered all the information we need in the preprocessing, we can get back to $I_{2,2}$. With the new defined variables inserted, we obtain

$$I_{2,2} = \sum_{i=1}^{l^*+j^*-1} \tilde{m}(u_{n+2-l^*+q_{i-1}})\tilde{\phi}(u_{n+1-y_{i-1}})[f(u_{n+1-l^*} + \tilde{u_i}) - f(u_{n+1-l^*} + \tilde{u}_{i-1})] \quad (5.67)$$

$$+ \tilde{m}(u_{n+1}) \sum_{i=0}^{n-j^*} \tilde{\phi}(u_{i-1})[f(u_i) - f(u_{i-1})] \quad (5.68)$$

$$+ \tilde{\phi}(u_{n+1})\tilde{m}(u_{n-l^*+2})[f(u_{n+1}) - f(u_n) - f(u_{n-l^*+1})]. \quad (5.69)$$

Every term of (5.30) has been discretized. The whole discretization is obtained by calculating

$$\tilde{\phi}(u_{n+1}) = \tilde{\phi}(-\infty) + I_1 - I_2 + I_3. \quad (5.70)$$

This computation yields

$$\tilde{\phi}(u_{n+1}) = \tilde{\phi}(-\infty) + \tilde{\phi}(-\infty)[S(n) + \tilde{m}(u_{n+1})[f(u_{n+1}) - f(u_n)]] \quad (5.71)$$

$$- M(n) - \tilde{\phi}(u_{n+1})[f(u_{n+1}) - f(u_n)] - \tilde{\phi}(u_{n+1}) \sum_{i=0}^{n+1-l^*} \tilde{m}(u_i)[f(u_i) - f(u_{i-1})] \quad (5.72)$$

$$- \tilde{\phi}(u_{n+1})\tilde{m}(u_{n-l^*+2})[f(u_{n+1}) - f(u_n) - f(u_{n-l^*+1})] \quad (5.73)$$

$$- \sum_{i=1}^{l^*+j^*-1} \tilde{m}(u_{n+2-l^*+q_{i-1}})\tilde{\phi}(u_{n+1-y_{i-1}})[f(u_{n+1-l^*} + \tilde{u_i} - u_0) - f(u_{n+1-l^*} + \tilde{u}_{i-1} - u_0)] \quad (5.74)$$

$$- \tilde{m}(u_{n+1}) \sum_{i=0}^{n-j^*} \tilde{\phi}(u_i)[f(u_i) - f(u_{i-1})], \quad (5.75)$$

where we have

$$S(n+1) = S(n) + \tilde{m}(u_{n+1})[f(u_{n+1}) - f(u_n)] \quad (5.76)$$

and

$$M(n+1) = M(n) + \tilde{\phi}(u_{n+1})[f(u_{n+1}) - f(u_n)]. \quad (5.77)$$

As the initial condition we set $S(-1) = M(-1) = 0$. However due to the sums in (5.72) and (5.75), we still have quadratic effort. But this can be avoided by introducing

$$O(n) := \sum_{i=0}^{n+1-l^*} \tilde{m}(u_i)[f(u_i) - f(u_{i-1})] \quad (5.78)$$

and

$$Q(n) := \sum_{i=j^*}^{n-j^*} \tilde{\phi}(u_i)[f(u_i) - f(u_{i-1})] + \tilde{\phi}(f(u_0 - h(j^*+1))) \quad (5.79)$$

with the update rules

$$O(n+1) = O(n) + \tilde{m}(u_{n+2-l^*})[f(u_{n+2-l^*}) - f(u_{n+1-l^*})] \tag{5.80}$$

and

$$Q(n+1) = Q(n) + \tilde{\phi}(u_{n+1-j^*})[f(u_{n+1-j^*}) - f(u_{n-j^*})]. \tag{5.81}$$

The variables O and Q are well defined because l^* and j^* are at least 1. In this deduction we always assumed $n+1-l^* > 0$. But this is false for the first steps in the simulation. Therefore we will define

$$O(n) = O(n-1) + \tilde{m}(-\infty)[f(u_0 + (n+1-l^*)h) - f(u_0 + (n-l^*)h] \tag{5.82}$$

and

$$Q(n) = Q(n-1) + \tilde{\phi}(-\infty)[f(u_0 + (n-j^*)h) - f(u_0 + (n-1-j^*)h)] \tag{5.83}$$

for $n < l^* - 1$ and $n < j^*$ respectively. As the initial values we use

$$O(-1) = \tilde{m}(-\infty) f(u_0 - hl^*) \tag{5.84}$$

and

$$Q(-1) = \tilde{\phi}(-\infty) f(u_0 - h(j^* + 1)) \tag{5.85}$$

to obtain the correct results. Putting O and Q into (5.71) to (5.75) yields

$$\tilde{\phi}(u_{n+1}) = \tilde{\phi}(-\infty) + \tilde{\phi}(-\infty)[S(n) + \tilde{m}(u_{n+1})[f(u_{n+1}) - f(u_n)]] \tag{5.86}$$

$$- M(n) - \tilde{\phi}(u_{n+1})[f(u_{n+1}) - f(u_n)] - \tilde{\phi}(u_{n+1})O(n) - \tilde{m}(u_{n+1})Q(n) \tag{5.87}$$

$$- \tilde{\phi}(u_{n+1})\tilde{m}(u_{n-l^*+2})[f(u_{n+1}) - f(u_n) - f(u_{n-l^*+1})] \tag{5.88}$$

$$- \sum_{i=1}^{l^*+j^*-1} \tilde{m}(u_{n+2-l^*+q_{i-1}})\tilde{\phi}(u_{n+1-y_{i-1}})[f(u_{n+1-l^*} + \tilde{u}_i - u_0) - f(u_{n+1-l^*} + \tilde{u}_{i-1} - u_0)]. \tag{5.89}$$

The sum in the last equation above will yield a quadratic and linear term in $\tilde{\phi}(u_{n+1})$ so we will split it up. This way we obtain

$$\tilde{\phi}(u_{n+1}) = \tilde{\phi}(-\infty) + \tilde{\phi}(-\infty)[S(n) + \tilde{m}(u_{n+1})[f(u_{n+1}) - f(u_n)]] \tag{5.90}$$

$$- M(n) - \tilde{\phi}(u_{n+1})[f(u_{n+1}) - f(u_n)] - \tilde{\phi}(u_{n+1})O(n) - \tilde{m}(u_{n+1})Q(n) \tag{5.91}$$

$$- \tilde{\phi}(u_{n+1})\tilde{m}(u_{n-l^*+2})[f(u_{n+1}) - f(u_n) - f(u_{n-l^*+1})] \tag{5.92}$$

$$- \sum_{i=1}^{l^*+j^*-2} \tilde{m}(u_{n+2-l^*+q_{i-1}})\tilde{\phi}(u_{n+1-y_{i-1}})[f(u_{n+1-l^*} + \tilde{u}_i - u_0) - f(u_{n+1-l^*} + \tilde{u}_{i-1} - u_0)] \tag{5.93}$$

$$- \tilde{m}(u_{n+1})\tilde{\phi}(u_{n+1-j^*})[f(w^{n+1}_{n+1-j^*}) - f(u_n)]. \tag{5.94}$$

# Elements	L^2-error	EOC in L^2	L^∞-error	EOC in L^∞
50	1.29927	-	0.0250247	-
100	0.450672	1.5275475676992607	0.0101168	1.3066064275748301
200	0.153403	1.5547536813705434	0.00407696	1.311182153107891
400	0.0558441	1.4578457386762156	0.0017411	1.2274971804453685
800	0.0222726	1.3261379803853823	0.00147244	0.2417923432251396

Table 5.1: Convergence of the new algorithm for the simpler integral equation

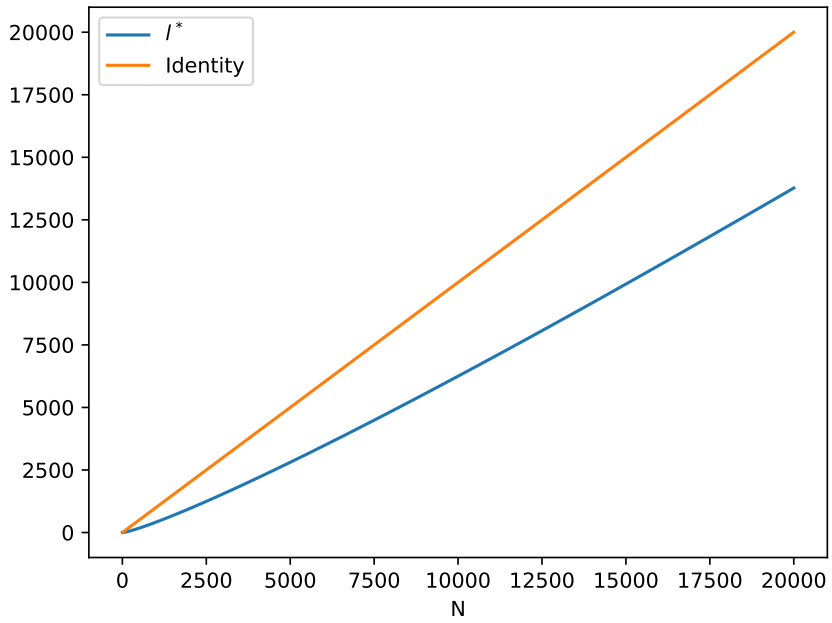

Figure 5.5: Relationship of l^* and N

One can observe that $l^* = 1$ is a special case because then we obtain a cubic problem in $\tilde{\phi}$. Otherwise the problem becomes quadratic. In the code this problem will be solved using NEWTON'S method. If we compare the results that this algorithms (A3) produces with (A1), we obtain table 5.1. One can clearly observe that convergence is obtained in both norms. Because we used constant ansatz functions we would expect an order of 1. Possibly, because we approximated the history integral with a much higher order, we obtain a slightly better convergence. This effect becomes less relevant the more elements we use. To understand the values for the L^∞ norm we have to recall that we do not compare our results to a known exact solution but we rather compare it to another approximation by an algorithm we trust. This can explain the drop in order of convergence for the L^∞ norm.

The last thing to consider is the computational effort of this method. One might be tempted to think of linear runtime. However, l^* and j^* increase as h decreases and h

decreases only as N increases for a fixed end time T. A priori we cannot give a formula for this so we will empirically take a look at this relationship in figure 5.5. This clearly shows a linear relationship in l^*. This leads to a $\mathcal{O}(N^2)$ algorithm. However, every time step within one discretization takes the same amount of time. The brute force algorithm on the other hand slows down from time step to time step. In the outlook given in chapter 6, we will discuss possibilities, which could improve the newly developed algorithm.

6 Conclusion and Outlook

In this chapter we will wrap up all results, we obtained throughout this work and give an idea for future research in this topic.

In the introduction we formulated the goal to be the development of an efficient method to simulate the governing equations for non-NEWTONIAN fluids in two spatial dimensions. This was achieved using the LAPLACE transform to eliminate the bothering integral from the governing equations completely. The resulting algorithm is as efficient as a normal CFD simulation. We could simulate the cross section of a rheometer, which has practical value. Theoretical existence and uniqueness of the solution to the governing equation could be shown for a sensible set of parameters. Even though this could not be shown for $\mu_s = 0$ we saw empirical convergence in the numerical experiments.

The second thing we looked at was the possibility to solve the integral from the governing equations without the LAPLACE transform to be able to loosen some of the necessary assumptions. One approach we took was to allow a combination of multiple relaxation times instead of one. This had the advantage that all methods and proofs, which we developed in this work, hold for the new system of equations. However, for each additional relaxation time we obtain two variables and one equation more, which is not ideal performance-wise but it returns the expected results. For other ideas, we studied the equation from [11]. Three approaches have been considered:

- brute force
- exponentially increasing time intervals
- transformation of the argument

However, all of them were not perfectly suited. The first and third approach have a computational effort of $\mathcal{O}(N^2)$ and would therefore not be appropriate for an extension to the five-dimensional system of equations with a sensible number of degrees of freedom. The second approach did unfortunately not lead to a correct solution as the calculation error was to great.

For future research it might be worth to take a look at our newly developed algorithm from the third approach again. In it current state in timestep $n \to n+1$ n history points are considered. Maybe one could carefully choose a subset of them and still obtain a good approximation? Another point to consider could be the difficulties that arise by introducing nonlinearity into the equations by either increasing the spatial dimension or allowing more flow directions. The theoretical existence of a solution might be interesting to study for this case. Another possibility to enhance the newly developed algorithm

could lie in the choice of f. We saw in figure 5.3 that $f(u) = \exp(\pi/2 \sinh(u))$ has better properties regarding the history dependency. However, $F(u, u')$ then becomes heavily dependent on u and introduces new effort each timestep.
One could also take a look back at the combination of relaxation times. It turns out that in Physics one often just needs a sufficient number of relaxation times to correctly model a fluids behavior. Therefore, it could be interesting to study the resulting system of equations with respect to the linear algebra. The only nonlinear component, which could be introduced into the system of equations, would be in the velocity equation. Therefore, one could discuss the opportunities to optimize performance for the solver.

All in all, one can say that for a spatial two-dimensional problem with one flow direction and a memory function of special form we could present an efficient algorithm. For a related equation, we studied some approaches that can be a good starting point for further studies.

List of Figures

List of Tables

References

[1] Martin S. Alnæs, *UFL: a finite element form language*, ch. 17, Springer, 2012.

[2] Martin S. Alnæs, Jan Blechta, Johan Hake, August Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie E. Rognes, and Garth N. Wells, *The FEniCS project version 1.5*, Archive of Numerical Software **3** (2015), no. 100.

[3] Martin S. Alnæs, Anders Logg, and Kent-Andre Mardal, *UFC: a finite element code generation interface*, ch. 16, Springer, 2012.

[4] Martin S. Alnæs, Anders Logg, Kent-Andre Mardal, Ola Skavhaug, and Hans Petter Langtangen, *Unified framework for finite element assembly*, International Journal of Computational Science and Engineering **4** (2009), no. 4, 231–244.

[5] Martin S. Alnæs, Anders Logg, Kristian B. Ølgaard, Marie E. Rognes, and Garth N. Wells, *Unified form language: A domain-specific language for weak formulations of partial differential equations*, ACM Transactions on Mathematical Software **40** (2014), no. 2.

[6] James W. Cooley and John W. Tukey, *An algorithm for the machine calculation of complex* Fourier *series*, Mathematics of Computation **19** (1965), 297–301.

[7] R Courant, K Friedrichs, and H. Lewy, *Über die partiellen Differenzengleichungen der mathematischen Physik*, Mathematische Annalen **100** (1928), 32–74, In German.

[8] A.S.R Duarte, A.I.P. Miranda, and P.J. Oliveira, *Numerical and analytical modeling of unsteady viscoelastic flows: The start-up and pulsating test case problems*, Journal of Non-Newtonian Fluid Mechanics **154** (2008), 153–169.

[9] Sashikumaar Ganesan and Lutz Tobiska, *Finite elements: Theory and algorithms*, Cambridge IISc Series, Cambridge University Press, 2017.

[10] Vivette Girault and Pierre-Arnaud Raviart, *Finite element methods for navier-stokes equations: theory and algorithms*, Springer series in computational mathematics, Springer, 1986.

[11] M. V. Gnann and Th. Voigtmann, *Asymptotic analysis of mode-coupling theory of active nonlinear microrheology*, Physical Review E **86** (2012), 011406.

[12] W. Götze and L. Sjögren, *General properties of certain non-linear integro-differential equations*, Journal of Mathematical Analysis and Applications (1995), no. 195, 230–250.

[13] E. Hairer, *A- and b-stability for runge-kutta methods - characterizations and equivalence*, Numerische Mathematik **48** (1986), 383–389.

[14] M.A. et al. Hulsen, *A new approach to the deformation fields method for solving complex flows using integral constitutive equations*, Journal of Non-Newtonian Fluid Mechanics **98** (2001), 201–221.

[15] Martien A. Hulsen and Patrick D. Anderson, *The deformation fields method revisited: Stable simulation fo instationary viscoelastic fluid flow using integral models*, Journal of Non-Newtonian Fluid Mechanics **262** (2018), 68–78.

[16] Jürgen Jost, *Partial Differential Equations*, 3 ed., Springer, 2013, Chapter 11.

[17] Robert C. Kirby and Anders Logg, *A compiler for variational forms*, ACM Transactions on Mathematical Software **32** (2006), no. 3.

[18] Hans Petter Langtangen and Anders Logg, *Solving PDEs in Python -the FEniCS tutorial volume 1*, September 2017.

[19] Anders Logg, Kent-Andre Mardal, Garth N. Wells, et al., *Automated solution of differential equations by the finite element method*, Springer, 2012.

[20] Anders Logg, Kristian B. Ølgaard, Marie E. Rognes, and Garth N. Wells, *FFC: the FEniCS form compiler*, ch. 11, Springer, 2012.

[21] Anders Logg and Garth N. Wells, *DOLFIN: Automated finite element computing*, ACM Transactions on Mathematical Software **37** (2010), no. 2.

[22] Anders Logg, Garth N. Wells, and Johan Hake, *DOLFIN: a C++/Python finite element library*, ch. 10, Springer, 2012.

[23] Grzegorz Łukaszewicz and Piotr Kalita, *Navier-stokes equations - an introduction with applications*, vol. 1, Springer, 2016.

[24] Kristian B. Ølgaard and Garth N. Wells, *Optimisations for quadrature representations of finite element tensors through automated code generation*, ACM Transactions on Mathematical Software **37** (2010).

[25] Jennifer Ouellette, *An-ti-ci-pa-tion: The physics of dripping honey*, article online at https://blogs.scientificamerican.com/cocktail-party-physics/an-ti-ci-pa-tion-the-physics-of-dripping-honey/, April 2013, accessed: June 11, 2019.

[26] R. G. Owens and T. N. Phillips, *Computational rheology*, Imperial College Press, 2002, chapter 1.

[27] Pekaje, *Maxwell model for viscoelastic materials*, https://commons.wikimedia.org/w/index.php?curid=2280260 accessed: June 11, 2019.

[28] Frank M. White, *Viscous fluid flow*, 3 ed., McGraw Hill, 2006.

[29] D. V. Widder, *What is the* Laplace *transform?*, The American Mathematical Monthly **52** (1945), no. 8, 419–425.

[30] W. H. Young, *On class of summable functions and their* Fourier *series*, Proceedings of the Royal Society A **87** (1912), 225–229.

Eigenständigkeitserklärung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne die Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden. Ich versichere, dass die eingereichte elektronische Fassung der eingereichten Druckfassung vollständig entspricht.

Köln, 15. Oktober 2019

Nils Dornbusch