AN ADAPTIVE VISCOELASTIC FLUID SOLVER: FORMULATION, VERIFICATION, AND APPLICATIONS TO FLUID-STRUCTURE INTERACTION

Aaron Barrett

A dissertation submitted to the faculty at the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Mathematics in the College of Arts and Sciences.

Chapel Hill
2019

Approved by:

Boyce E. Griffith

M. Gregory Forest

Ehssan Nazockdast

Robert D. Guy

Katherine Newhall

ABSTRACT

Aaron Barrett: An Adaptive Viscoelastic Fluid Solver: Formulation, Verification, and Applications to Fluid-Structure Interaction
(Under the direction of Boyce E. Griffith)

Many biological fluids exhibit non-Newtonian responses to stresses, and, as such, the Navier-Stokes equations are inadequate to properly model these fluids. To model the polymeric, visco-elastic fluids encountered in biological systems, we often turn to Oldroyd-B type models, where the immersed polymers are modeled using a linear Hooke's law for the restoring force. However, theory for how these models behave is severely lacking, and obtaining convergence in numerical simulations of these fluid models remains a significant challenge. These hyperbolic fluid models yield stress concentrations near large extensional flows. Sufficient spatial resolution is required to accurately resolve highly localized stress concentrations. However, away from such regions, lower computational resolution often suffices. This thesis presents a novel adaptive viscoelastic fluid solver that allows for the use of high spatial resolution where needed but permits the use of a coarser resolution away from areas of interest. The fluid solver is built into the open-source Immersed Boundary Adaptive Mesh Refinement (IBAMR) library to utilize recent adaptations of the immersed boundary method. We present applications of the fluid solver with an immersed interface method for complex fluids and a twirling flagellum in non-Newtonian fluids.

To my mother, who taught me perseverance by example.
To my father, who taught me the joy of exploring and learning.

ACKNOWLEDGEMENTS

Attempting to acknowledge everyone who helped would be a dissertation itself. While it downplays everyone's importance, and I will fail to acknowledge everyone, I will acknowledge a few groups. First, I would like to thank my family, for believing in me, providing constant support, and understanding and encouraging me despite my physical absence. Second, I thank my friends, in particular, Ben Roth, Mike Lee, Marie Nesfield, and Lee Barnett, for giving me plenty of distractions, as well as reminding me how to do simple (and not so simple) mathematics. Third, I thank all of my teachers, from kindergarten up to graduate school, for leading and inspiring me to learn. In particular, two of my first teachers Andrea Krueger and Susanne Church for putting up with my antics, and my advisors Boyce Griffith and Greg Forest for their patience and guidance. Fourth, science could not be done without funding, and I acknowledge and thank the NSF, in particular, their support through awards OAC 1450327 and DMS 1664645. Finally, I thank Sarah LaPotin, for her patience and understanding as I work a thousand miles away, and for being an absolute joy to be around.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Many biological systems involve flexible structures immersed in viscoelastic fluids, for example, sperm in the reproductive tract [3], bacterial motion in the stomach [4], and cilial transport of mucus in the lungs [5]. However, despite the ubiquity of viscoelastic fluids in real world applications, it remains difficult to accurately and efficiently resolve even the simplest continuum models. The models considered in this work are Maxwell type models [6] that are hyperbolic in nature. These types of models generate large stress gradients near extensional points, which are frequently found near fluid-structure interfaces. Recent work by Thomases *et al´* [7] have used asymptotic results in extensional flows to show the development of singular solutions for high elasticity. Despite work to develop high-fidelity finite element [2] and finite volume [8] numerical algorithms, relatively simple problems such as flow past a confined cylinder still remain unsolved. Under-resolved fluid flow in the wake of the cylinder causes oscillations in the drag coefficient of the cylinder, and it is computationally challenging to deploy the mesh resolution required to fully realize the stress gradients in the wake of the cylinder. In lid-driven cavity flow, boundary layers form near the corners of the box, causing recirculation regions in the fluid [9].

It is interesting to ask whether these issues are numerical in nature, a feature of the model, or physical in nature. In reality, it is likely a combination of all of the above. The molecular picture of viscoelastic fluids is complex. The polymers are frequently highly entangled with a spectrum of relaxation times. These polymers give rise to a number of different features, such as viscoelastic rod climbing, shear-thinning behavior, and the generation of normal stress differences. Many of the nonlinear observations can be understood through simple thought experiments. Consider a magnetic bead being dragged through a viscoelastic fluid. In physical experiments, we obtain shear thinning of the stresses as the bead approaches a steady-state velocity. This can be explained simply as the polymers breaking as the bead builds up stresses and stretches polymers to their breaking point. Normal stress differences can be explained by polymers naturally orienting themselves along fluid

streamlines while entropic forces attempt to return the polymer to its undisturbed configuration. While we can explain some of these nonlinear results, unphysical simplifying assumptions are commonly employed to produce tractable models. As an example, although real biological fluids will include constituents with a broad spectrum of relaxation times, the mathematical models that are most frequently studied only have a single relaxation time. While we can overcome this by having multiple stress modes, this introduces additional model parameters that must be fitted. Another drawback is that the simpler models assume dilute solvents; however, as we increase the elasticity parameters, we are inherently assuming the solute is becoming more and more prominent. This calls into question the validity of our modelling assumption. As will be demonstrated in later chapters, the nonlinear models give rise to high stress gradients. These high stress gradients are a feature of the models. However, under-resolving these features results in unstable simulations. The typical instabilities that appear in Newtonian fluid simulations are related to parcels of fluid moving more than a fixed fraction of a grid cell which are a result of using too large of a time step. Simulations with complex fluids can also have this kind of instability; however, in addition to the *time step* requirement, the fluid grid needs to be sufficiently fine for stability.

There are many unsolved problems with complex fluids. This work aims to specifically address the numerical issues found in simulating the most common viscoelastic models. We cannot answer all the open questions, but we wish to provide a framework for exploring the features of different models in future work. We attempt to overcome the issue of under-resolved simulations by building an adaptive fluid solver that provides additional spatial resolution near large stress gradients. While our discretizations are relatively simple, we show that by achieving this high level of spatial resolution, we can obtain similar results to other, more complicated numerical methods. We then demonstrate several methods of resolving simulations with fluid-structure interactions.

The remainder of this dissertation is divided into four parts. Chapter 2 gives an introduction to fluid mechanics and derives the Maxwell type models that we will be using for the rest of the work. We also demonstrate the sharp features that result from the Oldroyd-B model specifically. Chapter 3 provides numerical details on the fluid solver with a few examples demonstrating the accuracy and ability of the method. In chapters 4 and 5, we provide applications of the fluid solver to fluid-structure interaction problems. In chapter 4, we explore the difference between the immersed boundary and immersed interface methods, providing examples of each method. Finally, in chapter

5, we use the generalized immersed boundary method to simulate a twirling flagellum in both a Newtonian fluid and viscoelastic fluids of different elasticities.

CHAPTER 2

MATHEMATICAL MODELS OF NON-NEWTONIAN FLUIDS

In this chapter, we derive the equations of motion for Maxwell type models. While the discussion in this chapter will be complete, we omit several details. We point the interested reader to the books of Morosov and Spagnolie [6] and Larson [10] for more detailed discussions. We begin by deriving the conservation of mass and momentum.

2.1 Conservation Equations

2.1.1 Lagrangian and Eulerian Variables

We can describe our state variables such as density, velocity, and pressure in two different ways. The first is using a material or Lagrangian description. Here, we associate a particular parcel of the fluid with some label $\mathbf{X}$, usually chosen to be the initial position of the fluid parcel, as shown in Figure 2.1. The velocity of the portion of fluid at a particular time t can be written as $\mathbf{U}(\mathbf{X}, t)$. We can also describe the fluid using a spatial or Eulerian description. Here, we describe the fluid at a fixed position $\mathbf{x}$ in space. That is, the velocity at a particular position $\mathbf{x}$ and time t would be written as $\mathbf{u}(\mathbf{x}, t)$. We use the map $\boldsymbol{\chi}$ that relates a particular Lagrangian coordinate $\mathbf{X}$ to its current physical location $\mathbf{x}$ at some time t via $\mathbf{x} = \boldsymbol{\chi}(\mathbf{X}, t)$. If we define the velocity of the fluid at the point $\mathbf{x}$ as $\mathbf{u}(\mathbf{x}, t)$, then we can define the velocity of the material point $\mathbf{X}$ as

$$\left.\frac{d\mathbf{x}}{dt}\right|_{\mathbf{X}} = \frac{d\boldsymbol{\chi}(\mathbf{X}, t)}{dt} = \mathbf{u}(\boldsymbol{\chi}(\mathbf{X}, t), t). \tag{2.1}$$

Suppose we have a scalar Lagrangian variable F. The time rate of change given in Lagrangian coordinates is $\frac{\partial F(\mathbf{X},t)}{\partial t}$. If we wish to describe this variable in terms of the spatial variables $f(\boldsymbol{\chi}(\mathbf{X}, t), t) = F(\mathbf{X}, t)$, then we must account for the change of frame. Using the chain rule,

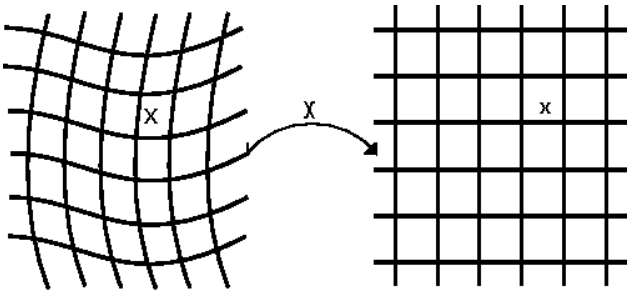

Figure 2.1: The reference configuration (left) can be mapped into the current configuration (right) by the mapping $\boldsymbol{\chi}(\mathbf{X}, t)$.

we obtain

$$\left.\frac{d}{dt} f(\boldsymbol{\chi}(\mathbf{X}, t), t)\right|_{\mathbf{X}} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial x_i}\frac{\partial x_i}{\partial t} \tag{2.2}$$

$$= \left(\frac{\partial}{\partial t} + \mathbf{u}\cdot\nabla\right) f = \frac{Df}{Dt}. \tag{2.3}$$

The operator $\frac{D}{Dt}$ is called the material derivative and represents the time derivative along material trajectories. Using this derivative, we can represent the acceleration of a parcel of fluid as

$$\frac{D\mathbf{u}(\mathbf{x}, t)}{Dt} = \frac{\partial \mathbf{u}(\mathbf{x}, t)}{\partial t} + \mathbf{u}(\mathbf{x}, t)\cdot\nabla\mathbf{u}(\mathbf{x}, t). \tag{2.4}$$

One important measure of deformation is the derivative of the map $\boldsymbol{\chi}(\mathbf{X}, t)$, called the deformation gradient tensor, given by $\mathbb{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}$. One can think of $\mathbb{F}$ as a mapping that sends a line element $d\mathbf{X}$ in a material configuration to a line element in $d\mathbf{x}$ in physical space as $d\mathbf{x} = \mathbb{F}\cdot d\mathbf{X}$. We can characterize the deformation by looking at the determinant of the deformation tensor $J = \det\mathbb{F}$, which satisfies the following relation

$$\frac{\partial J(\mathbf{X}, t), t}{\partial t} = (\nabla\cdot\mathbf{u}(\boldsymbol{\chi}(\mathbf{X}, t), t))\, J(\mathbf{X}, t, t). \tag{2.5}$$

The Jacobian gives a measure of area change from the deformation. An incompressible material is one for which $J(\mathbf{X}, t) \equiv 1$, which implies that $\nabla\cdot\mathbf{u} = 0$ for all $\mathbf{x}$ and t. From now on, we will assume that the fluid is incompressible.

2.1.2 Conservation of Mass

The density of the fluid $\rho(\mathbf{x}, t)$ is defined as the mass per volume in an infinitesimal fluid parcel centered at $\mathbf{x}$. We now consider a volume of fluid in a deforming region $\Omega(t) \subset \mathbb{R}^3$ in the current configuration. The total mass of fluid in $\Omega(t)$ can be found by integrating the density over the volume, $M(t) = \int_{\Omega(t)} \rho(\mathbf{x}, t) dV_x$. If we assume that no mass is created or destroyed, then the time rate of change of the mass must be zero. We can write the conservation of mass in Eulerian form as

$$\frac{d}{dt} \int_{\Omega(t)} \rho(\mathbf{x}, t) dV_x = 0, \tag{2.6}$$

where dV_x is a volume infinitesimal in the Eulerian frame. We can form a differential form of mass conservation by changing coordinates to the Lagrangian format:

$$\begin{aligned}
0 &= \frac{d}{dt} \int_{\Omega(t)} \rho(\mathbf{x}, t) dV_x && (2.7)\\
&= \frac{d}{dt} \int_{\Omega(0)} \rho(\boldsymbol{\chi}(\mathbf{X}, t), t) J(\mathbf{X}, t)\, dV_a && (2.8)\\
&= \int_{\Omega(0)} \left(\frac{D\rho(\boldsymbol{\chi}(\mathbf{X}, t), t)}{Dt} + \rho(\boldsymbol{\chi}(\mathbf{X}, t), t) \left(\nabla \cdot \mathbf{u}\left(\boldsymbol{\chi}\mathbf{X}, t\right)\right) \right) J(\mathbf{X}, t)\, dV_a && (2.9)\\
&= \int_{\Omega(t)} \left(\frac{D\rho(\mathbf{x}, t)}{Dt} + \rho(\mathbf{x}, t) \left(\nabla \cdot \mathbf{u}\left(\mathbf{x}, t\right)\right) \right) dV_x. && (2.10)
\end{aligned}$$

Since this integral holds for any material volumes, we get that it must hold pointwise,

$$\frac{D\rho(\mathbf{x}, t)}{Dt} + \rho(\mathbf{x}, t) \left(\nabla \cdot \mathbf{u}\left(\mathbf{x}, t\right)\right) = 0. \tag{2.11}$$

Because our fluid is incompressible, this equation reduces to $\frac{D\rho}{Dt} = 0$, or that the density of a material point remains constant as it moves with the fluid. From here on out, we assume the density of the fluid is uniform, so the statement of conservation of mass reduces to

$$\nabla \cdot \mathbf{u}\left(\mathbf{x}, t\right) = 0. \tag{2.12}$$

2.1.3 Conservation of Momentum

We can define the momentum of a fluid in a given control volume $\Omega(t)$ can be written in terms of Eulerian coordinates as $\mathbf{P}(t) = \int_{\Omega(t)} \rho(\mathbf{x}, t)\mathbf{u}(\mathbf{x}, t) dV_x$. We can look at the rate of change of

momentum by following a similar technique in the previous section:

$$\frac{d}{dt}\mathbf{P}(t) = \frac{d}{dt}\int_{\Omega(t)} (\rho\mathbf{u})(\mathbf{x},t)dV_x \tag{2.13}$$

$$= \frac{d}{dt}\int_{\Omega(0)} (\rho\mathbf{u})(\boldsymbol{\chi}(\mathbf{X},t),t)\, JdV_a \tag{2.14}$$

$$= \int_{\Omega(0)} \left(\frac{D\rho}{Dt}\mathbf{u} + \rho\frac{D\mathbf{u}}{Dt} + \rho\mathbf{u}(\nabla\cdot\mathbf{u})\right) JdV_a \tag{2.15}$$

$$= \int_{\Omega(0)} \left(\rho\frac{D\mathbf{u}}{Dt}\right) JdV_a = \int_{\Omega(t)} \rho\frac{D\mathbf{u}}{Dt}(\mathbf{x},t)dV_x. \tag{2.16}$$

Newton's second law states that the rate of change of the momentum must be equal to the net forces acting on the fluid. These forces come from two sources, external body forces such as gravity denoted by $\mathbf{f}$, and surface forces such as viscous or elastic stresses denoted by the traction $\mathbf{t}$. We can describe the surface traction by $\mathbf{t} = \boldsymbol{\sigma}\cdot\mathbf{n}$, where $\mathbf{n}$ is the outward-pointing normal vector and $\boldsymbol{\sigma}$ is the Cauchy stress tensor. Balancing these forces with the time rate of change of the momentum leads to the following integral form of momentum conservation

$$\frac{d}{dt}\mathbf{P}(t) = \int_{\Omega(t)} \rho\frac{D\mathbf{u}(\mathbf{x},t)}{Dt}dV_x = \int_{\Omega(t)} \mathbf{f}(\mathbf{x},t)\,dV_x + \int_{\partial\Omega(t)} \mathbf{t}(\mathbf{x},t)\,dS_x, \tag{2.17}$$

where dS_x is an infinitesimal surface area element and $\partial\Omega(t)$ is the boundary of $\Omega(t)$. We can convert this integral to a volume integral using the divergence theorem:

$$\frac{d}{dt}\mathbf{P}(t) = \int_{\Omega(t)} \rho\frac{D\mathbf{u}(\mathbf{x},t)}{Dt}dV_x = \int_{\Omega(t)} \mathbf{f}(\mathbf{x},t)\,dV_x + \int_{\Omega(t)} \nabla\cdot\boldsymbol{\sigma}(\mathbf{x},t)\,dV_x. \tag{2.18}$$

Because this holds for any volume $\Omega(t)$, it follows that it must hold pointwise, which results in the differential form of conservation of momentum:

$$\rho\frac{D\mathbf{u}(\mathbf{x},t)}{Dt} = \mathbf{f}(\mathbf{x},t) + \nabla\cdot\boldsymbol{\sigma}(\mathbf{x},t). \tag{2.19}$$

2.1.4 The Cauchy Stress Tensor

Various fluid and material models reduce to a specification of the Cauchy stress tensor $\boldsymbol{\sigma}$. Since the traction on a surface is represented as $\mathbf{t} = \boldsymbol{\sigma}\mathbf{n}$ where $\mathbf{n}$ is the outward unit normal vector, σ_{ij} represents the traction in the j^{th}-direction on a surface that is perpendicular to the i^{th}-direction. The

net force $\mathbf{F}$ and torque $\mathbf{L}$ about a point $\mathbf{x}_0$ are given by integrating the traction over an immersed surface with boundary ∂S:

$$\mathbf{F} = \int_{\partial S} \boldsymbol{\sigma} \cdot \mathbf{n} da, \quad \mathbf{L} = \int_{\partial S} (\mathbf{x} - \mathbf{x}_0) \times (\boldsymbol{\sigma} \cdot \mathbf{n}) \, da. \tag{2.20}$$

If we assume that there are no body torques that give rise to internal angular momentum, then it can be shown that the stress tensor is symmetric $\boldsymbol{\sigma} = \boldsymbol{\sigma}^T$.

All that remains is to create a constitutive law that gives us the formulation of the Cauchy stress tensor. We first decompose the stress tensor $\boldsymbol{\sigma}$ into its volumetric and deviatoric stress components. Simply, we can write this as

$$\boldsymbol{\sigma} = -p\mathbb{I} + \boldsymbol{\tau}, \tag{2.21}$$

where p is the pressure, $\mathbb{I}$ is the identity tensor, and $\boldsymbol{\tau}$ is the deviatoric stress tensor. In a Newtonian fluid model, the deviatoric stress is related to the local strain rate. We can decompose the gradient of the velocity field into it's symmetric and antisymmetric parts $\nabla \mathbf{u} = \frac{1}{2}(\dot{\boldsymbol{\gamma}} + \boldsymbol{\omega})$, where

$$\dot{\boldsymbol{\gamma}} = \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) \text{ and } \quad \boldsymbol{\omega} = \left(\nabla \mathbf{u} - (\nabla \mathbf{u})^T\right). \tag{2.22}$$

Here, $\dot{\boldsymbol{\gamma}}$ is the rate of strain tensor and $\boldsymbol{\omega}$ is the vorticity tensor. We note that the rate of strain tensor corresponds to extensional flow, while the vorticity tensor corresponds to a rigid body rotation. A Newtonian fluid is defined so that the deviatoric stress is linearly related to the rate of strain and the fluid is isotropic. Because the stress is symmetric, the most general form of the deviatoric stress tensor can be written using a traceless part and an isotropic part:

$$\boldsymbol{\tau} = \mu \left(\dot{\boldsymbol{\gamma}} - \frac{2}{3}(\nabla \cdot \mathbf{u})\mathbb{I}\right) + \mu'(\nabla \cdot \mathbf{u})\mathbb{I}, \tag{2.23}$$

where μ is the viscosity and μ' is the dilational viscosity. In the case of an incompressible fluid, then the total stress tensor has the form

$$\boldsymbol{\sigma} = -p\mathbb{I} + \mu\dot{\boldsymbol{\gamma}}. \tag{2.24}$$

Inserting the stress into the momentum equation, we obtain the following incompressible Navier-

Stokes equations:

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{f} \tag{2.25}$$

$$\nabla \cdot \mathbf{u} = 0. \tag{2.26}$$

2.2 Viscoelastic Fluids

As discussed previously, a Newtonian fluid is a fluid in which the deviatoric stress is linearly related to the rate of strain. In particular, Newtonian fluids do not include stresses that depend on the flow history. Many fluids in biological settings contain microstructures, such as proteins and DNA, that push or pull on the fluid in response to fluid flow. These viscoelastic fluids are characterized by the presence of memory and stress anisotropy. Before we specify the constitutive equation for the Cauchy stress, we look at simple one-dimensional models designed to demonstrate elastic and viscous behavior.

2.2.1 Maxwell and Kelvin-Voigt Elements

Viscoelastic fluids are fluids that exhibit both elastic and viscous responses to deformation. We can consider a simple shear deformation which is characterized by its gradient γ. The stress in a simple elastic element under this deformation obeys Hooke's law

$$\sigma = E\gamma, \tag{2.27}$$

where E is the elastic modulus of the material. Meanwhile, the stress in a simple viscous element under this deformation is

$$\sigma = \eta\dot{\gamma}, \tag{2.28}$$

where η is the viscosity of the material and $\dot{\gamma}$ is the strain rate of the deformation. Note the strain rate tensor $\dot{\boldsymbol{\gamma}}$ is the three-dimensional analog of $\dot{\gamma}$. These elements can be simply thought of as springs and dashpots and are denoted as such. Figure 2.2 shows the various elements as well as a creep test in which one end of the element is displaced from time t_0 to time t_1, and the stress in the element is measured over time. Springs exhibit perfect elastic behavior and are often used as a simple model of an elastic solid. Dashpots, however, exhibit perfect viscous behavior and provide

the basis of Newtonian fluid models. The simplest viscoelastic materials consist of a spring and dashpot in either serial or parallel. The spring and dashpot in serial is referred to as a Maxwell element and has the stress-strain relationship

$$\sigma + \frac{\eta}{E}\dot{\sigma} = \eta\dot{\gamma}. \tag{2.29}$$

This element is an example of a viscoelastic *fluid* because at long time scales it behaves like a fluid. To see this, consider a Maxwell element being displaced on both ends. Both the spring and dashpot are stretched initially; however, when the spring relaxes, the resulting displacement can be shifted back onto the dashpot. This keeps the total displacement constant and results in the stress at long term behaving like a dashpot, $\sigma \sim \dot{\gamma}$. In fact, one can use the Maxwell element as a simple discrete model of viscoelasticity by immersing the elements inside a viscous Newtonian fluid [11].

In contrast, a spring and dashpot in parallel is referred to as a Kelvin-Voigt element and has a stress-strain relationship

$$\sigma = E\gamma + \eta\dot{\gamma}. \tag{2.30}$$

This element is an example of a viscoelastic *solid*. The long term behavior of Kelvin-Voigt elements differs because the dashpot is no longer able to absorb the relaxation of the spring. This relaxes the element back to its initial resting length, which results in the stress at long term behaving like a spring, $\sigma \sim \gamma$. Because our goal is to model polymeric fluids, we focus on the Maxwell element.

2.2.2 Maxwell Fluid Models

The linear Maxwell element combines viscous and elastic responses to deformations. Our first attempt in creating a continuous viscoelastic fluid model would be to replace the stress with the Cauchy stress, and the rate of strain with the rate of strain tensor

$$\boldsymbol{\sigma} + \frac{\eta}{E}\frac{\partial \boldsymbol{\sigma}}{\partial t} = \eta\dot{\boldsymbol{\gamma}}. \tag{2.31}$$

The issue is that this model is not *frame invariant*. To see this, consider two different frames of reference, a stationary frame, and a frame moving at a constant velocity of $\mathbf{u}_0$. Because the fluid is moving at a constant velocity, the stress should be constant as there is no strain applied to the fluid.

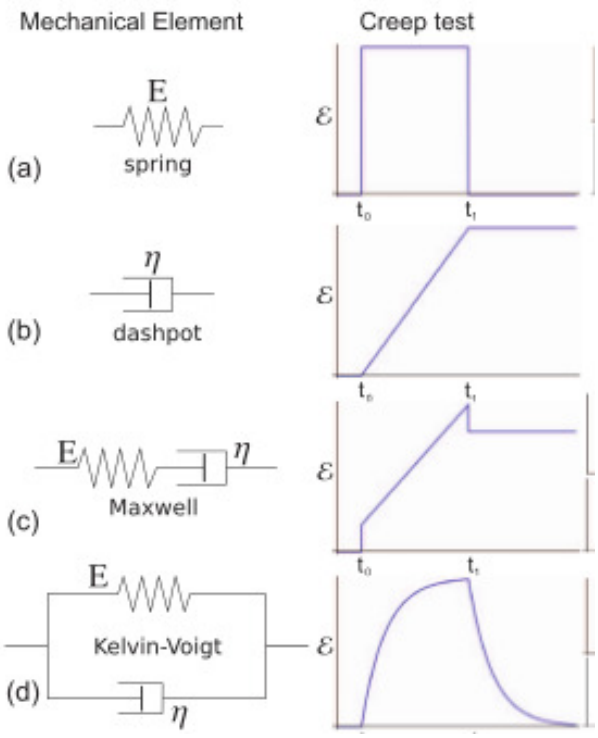

Figure 2.2: Creep tests for the different elements where a strain is applied to each element, and the resulting stress is measured. Note the long time difference between the Kelvin-Voigt and Maxwell elements.

However, the Cauchy stress for the moving frame can be written as

$$\frac{\partial}{\partial t}\boldsymbol{\sigma}\left(\mathbf{x}+\mathbf{u}_0 t\right)=\frac{\partial}{\partial t}\boldsymbol{\sigma}\left(\mathbf{x}\right)+\mathbf{u}_0\nabla\boldsymbol{\sigma}. \tag{2.32}$$

The time derivative of the stress differs by a term proportional to $\mathbf{u}_0$, indicating that the derivative $\frac{\partial}{\partial t}$ is not frame invariant. Similar to the material derivative $\frac{D}{Dt}=\frac{\partial}{\partial t}+\mathbf{u}\cdot\nabla$, we need a frame invariant derivative for tensors. For tensors, the material derivative is not sufficient because it does not allow for the rotation and stretching of the local basis of the tensor. Before we introduce frame invariant derivatives for tensors, we will demonstrate the importance of rotational frame invariance. Consider a simple material where the constitutive equation for the stress is

$$\boldsymbol{\sigma}=E\mathbb{F}, \tag{2.33}$$

where $\mathbb{F}$ is the deformation gradient. If the material then undergoes a simple rotation so that $\mathbb{F}=\mathbb{Q}$ where $\mathbb{Q}$ is a rotation tensor, then the resulting stress of the material is

$$\boldsymbol{\sigma}=E\mathbb{Q}. \tag{2.34}$$

However, since the material is under no strain, we would expect there to be no stress in the material. Clearly, the constitutive equation in equation (2.33) is not a valid relation since a solid body rotation induces a stress. Instead consider the right Cauchy-Green tensor

$$\mathbb{C}=\mathbb{F}^T\mathbb{F}. \tag{2.35}$$

The right Cauchy-Green tensor contains no rotational information from the resulting deformation. Consider the length of a vector $\mathbf{x}$ after deformation:

$$\mathbf{x}^T\mathbf{x}=\mathbf{X}^T\mathbb{F}^T\mathbb{F}\mathbf{X}=\mathbf{X}^T\mathbb{C}\mathbf{X}. \tag{2.36}$$

Therefore, $\mathbb{C}$ contains information about the stretch and shape change of vectors but is agnostic to their rotation. Now consider a material where the constitutive stress relationship is given by

$$\boldsymbol{\sigma} = E\mathbb{C}. \tag{2.37}$$

Now, under a solid body rotation where $\mathbb{F} = \mathbb{Q}$, the resulting stress is

$$\boldsymbol{\sigma} = E\mathbb{C} = E\mathbb{Q}^T\mathbb{Q} = E\mathbb{I}. \tag{2.38}$$

The material now is stress-free, only undergoing an isotropic pressure stress. The stress should not depend on the deformation gradient, but instead on a definition of deformation that is invariant under rotation, such as that provided by the right Cauchy-Green tensor. We can now look at how the right Cauchy-Green tensor evolves over time. First, the time derivative of the deformation gradient is

$$\frac{D\mathbb{F}}{Dt} = \frac{D}{Dt}\frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}\frac{\partial \frac{D\mathbf{x}}{Dt}}{\partial \mathbf{X}} = \mathbf{F}\nabla\mathbf{u}. \tag{2.39}$$

From equations (2.39) and (2.35), it follows that

$$\frac{D\mathbb{C}}{Dt} = \nabla\mathbf{u}\mathbb{C} + \mathbb{C}\nabla\mathbf{u}^T. \tag{2.40}$$

This time derivative of the right Cauchy-Green tensor defines our first frame invariant derivative. Two common forms of frame invariant derivatives for tensors are the lower- and upper-convective derivatives,

$$\overset{\triangle}{\boldsymbol{\sigma}} = \frac{\partial \boldsymbol{\sigma}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} + (\nabla\mathbf{u})^T\cdot\boldsymbol{\sigma} + \boldsymbol{\sigma}\cdot\nabla\mathbf{u}, \tag{2.41}$$

$$\overset{\triangledown}{\boldsymbol{\sigma}} = \frac{\partial \boldsymbol{\sigma}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} - \nabla\mathbf{u}\cdot\boldsymbol{\sigma} - \boldsymbol{\sigma}\cdot(\nabla\mathbf{u})^T. \tag{2.42}$$

We note that the upper-convective derivative is defined so that the derivative of the right Cauchy-Green tensor is zero, which follows from equation (2.40). Using these frame-invariant derivatives, we can finally define the Oldroyd models. We first separate the Cauchy stress into a Newtonian solvent

component and a polymeric component

$$\boldsymbol{\sigma} = \boldsymbol{\tau}_{\mathrm{n}} + \boldsymbol{\tau}_p, \quad \boldsymbol{\tau}_{\mathrm{n}} = -p\mathbb{I} + \mu_n \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right). \tag{2.43}$$

The polymeric stress $\boldsymbol{\tau}_p$ follows the Maxwell equation where the time derivative of the stress is replaced with a convective derivative. This introduces the Oldroyd-A model

$$\boldsymbol{\tau}_p + \lambda \overset{\triangle}{\boldsymbol{\tau}}_p = \mu_p \dot{\boldsymbol{\gamma}} \tag{2.44}$$

and the Oldroyd-B model

$$\boldsymbol{\tau}_p + \lambda \overset{\triangledown}{\boldsymbol{\tau}}_p = \mu_p \dot{\boldsymbol{\gamma}}, \tag{2.45}$$

where λ is the relaxation time of the polymers and μ_p is the polymer contribution to the viscosity. Typically, we discard the Oldroyd-A model since it yields results that disagree with experimental results. We note that the above models can be derived from a molecular argument where the immersed polymers are modelled with bead-spring chains. One major shortcoming of the Oldroyd-B model is that it allows for unbounded stress growth in the presence of extensional flows as will be demonstrated later. We can remedy this feature in the model by adding nonlinearities to the model. The Giesekus model adds in a quadratic nonlinearity governed by a dimensionless parameter α

$$\boldsymbol{\tau}_p + \lambda \overset{\triangledown}{\boldsymbol{\tau}}_p + \alpha \frac{\lambda}{\mu_p} \boldsymbol{\tau}_p^2 = \mu_p \dot{\boldsymbol{\gamma}}. \tag{2.46}$$

We note that the Giesekus model can be derived from the same molecular bead-spring model as the Oldroyd-B model, except that there is an anisotropic drag on the immersed polymers. The Giesekus model assumes the preferred direction for the drag is in the direction of the stress tensor, and α governs the amount of the anisotropic drag. Finally, the Rolie-Poly model is based on a detailed representation of the molecular structure of the polymers themselves. The constitutive relation for the model is

$$\boldsymbol{\tau}_p + \lambda \overset{\triangledown}{\boldsymbol{\tau}}_p = \mu_p \dot{\boldsymbol{\gamma}} - \frac{2}{3}\lambda \left(\boldsymbol{\tau}_p : \nabla \mathbf{u}\right) \left(\mathbb{I} + (1+\epsilon)\frac{\lambda}{\mu_p}\boldsymbol{\tau}_p\right). \tag{2.47}$$

While we will not visit this model again, we note that we believe this model to provide the most realistic results due to its derivation from a detailed molecular picture. The methods described in

this paper can be applied to all models that are based on the Oldroyd-B model. In the future, we will focus our applications on this model.

2.2.3 The Conformation Tensor

In this section, we derive the Oldroyd-B model using an alternative approach borrowing ideas from solid mechanics. Recall the deformation gradient $\mathbb{F} = \frac{\partial \boldsymbol{\chi}}{\partial \mathbf{X}}$ where $\boldsymbol{\chi}$ is the motion map that relates the reference coordinates with the physical coordinates and $\mathbf{X}$ is the reference configuration. We can define the left Cauchy-Green tensor

$$\mathbb{B} = \mathbb{F}\mathbb{F}^T \tag{2.48}$$

which is a strain measure of how lengths of line elements and angles between line elements change between configurations. The Cauchy stress in a neo-Hookean material can be described by

$$\boldsymbol{\tau} = G\left(\mathbb{B} - \mathbb{I}\right). \tag{2.49}$$

This is essentially the continuum analog of the one-dimensional elastic spring described earlier. A simple viscoelastic material consists of a superposition of relaxing neo-Hookean materials. Given some memory kernel $M\left(t\right)$, the stress can be found by integrating over all previous configurations

$$\boldsymbol{\tau} = \int_{-\infty}^{t} M\left(t - t'\right)\left(\mathbb{B}_{t'}\left(t\right) - \mathbb{I}\right) dt', \tag{2.50}$$

where $\mathbb{B}_{t'}\left(t\right)$ is the left Cauchy-Green tensor based on the deformation gradient where the reference configuration is given at time t'. If the memory kernel consists of a single relaxation time $M\left(t\right) = \frac{G}{\lambda} e^{-t/\lambda}$, then we can rewrite the model in a differential form and recover the Oldroyd-B model

$$\boldsymbol{\tau} + \lambda \overset{\nabla}{\boldsymbol{\tau}} = \mu \dot{\boldsymbol{\gamma}}. \tag{2.51}$$

The benefit of this viewpoint is it allows a simple explanation of the *conformation tensor* $\mathbb{C}$. When derived from an elastic dumbbell model of the immersed models, the conformation tensor represents an ensemble average of end to end points of the dumbbells. For the Oldroyd-B model, there is a

Model	$\mathbf{S}(\mathbb{C})$	$\mathbf{g}(\mathbb{C})$
Oldroyd-B	$\mathbb{C}-\mathbb{I}$	$-\frac{1}{\lambda}(\mathbb{C}-\mathbb{I})$
Oldroyd-B with S.D.	$\mathbb{C}-\mathbb{I}$	$-\frac{1}{\lambda}(\mathbb{C}-\mathbb{I})+D\nabla^2\mathbb{C}$
Giesekus	$\mathbb{C}-\mathbb{I}$	$-\frac{1}{\lambda}\left(\mathbb{C}-\mathbb{I}+\alpha(\mathbb{C}-\mathbb{I})^2\right)$
Rolie-Poly	$\mathbb{C}-\mathbb{I}$	$\frac{1}{\lambda_d}(\mathbb{C}-\mathbb{I})$ $-\frac{2}{\lambda_s}\left(1-\sqrt{\frac{3}{\mathrm{tr}(\mathbb{C})}}\right)\left(\mathbb{C}+\beta\left(\frac{\mathrm{tr}(\mathbb{C})}{3}\right)^{\delta}(\mathbb{C}-\mathbb{I})\right)$
FENE-P	$\frac{\mathbb{C}}{1-\mathrm{tr}(\mathbb{C})/l^2}-\frac{\mathbb{I}}{1-3/l^2}$	$\frac{-1}{\lambda}\left(\frac{\mathbb{C}}{1-\mathrm{tr}(\mathbb{C})/l^2}-\frac{\mathbb{I}}{1-3/l^2}\right)$

Table 2.1: A sample of different viscoelastic fluid models.

linear relationship between the conformation tensor and the stress

$$\boldsymbol{\tau} = G(\mathbb{C}-\mathbb{I}) = \mathbf{S}(\mathbb{C}). \tag{2.52}$$

In fact, this simple linear relationship is common to all the models that will be discussed in this work. When approaching from the solid mechanics viewpoint, the conformation tensor represents a superposition of relaxing left Cauchy-Green tensors

$$\mathbb{C}(t) = \frac{1}{\lambda}\int_{-\infty}^{t} e^{-(t-t')/\lambda}\mathbb{B}_{t'}(t)\,dt'. \tag{2.53}$$

Because the left Cauchy-Green tensor is symmetric positive definite, we conclude that the conformation tensor must also be symmetric positive definite. If at any point during a simulation the conformation tensor is *not* positive definite, then the simulation has lost physical relevance. Instead of evolving the stress tensor, we instead will evolve the conformation tensor. We can convert equation (2.53) into a differential equation which yields

$$\lambda\overset{\nabla}{\mathbb{C}} = \mathbf{g}(\mathbb{C}), \tag{2.54}$$

where $\mathbf{g}(\mathbb{C})$ is the relaxation function takes different forms for different fluid models. A collection of models is given in Table 2.1. We note that while equation (2.54) has no term enforcing the symmetric positive definiteness of the conformation tensor, it can be shown [12] that if the conformation tensor is initially symmetric positive definite, it will remain so for all time.

To conclude, we present a dimensional analysis of the complete model. For completeness, we

summarize the equations of motion:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\nabla\mathbf{u}\right)=\nabla\cdot\boldsymbol{\sigma}+\mathbf{f}\left(\mathbf{x},t\right), \tag{2.55}$$

$$\nabla\cdot\mathbf{u}=0, \tag{2.56}$$

$$\boldsymbol{\sigma}=-p\mathbb{I}+\mu_n\cdot\boldsymbol{\gamma}+\boldsymbol{\tau}_p, \tag{2.57}$$

$$\boldsymbol{\tau}_p=\frac{\mu_p}{\lambda}\mathbf{S}\left(\mathbb{C}\right), \tag{2.58}$$

$$\lambda\overset{\triangledown}{\mathbb{C}}=\mathbf{g}\left(\mathbb{C}\right). \tag{2.59}$$

The relevant dimensionless numbers are the Reynolds number $\mathrm{Re}=\frac{\rho UL}{\mu}$ where $\mu=\mu_p+\mu_n$ is the total viscosity, the Weissenberg number $\mathrm{Wi}=\frac{\lambda U}{L}$ and the Deborah number $\mathrm{De}=\frac{\lambda}{t_p}$ where t_p is a relevant time scale of observation. The Reynolds number is familiar from Newtonian fluid mechanics. It is the ratio of inertial to viscous forces. For fluids with very large viscosities or for very small length scales, the Reynolds number can be approximated as zero which reduces the equations to the well known Stokes equations,

$$0=\nabla\cdot\boldsymbol{\sigma}+\mathbf{f}, \tag{2.60}$$

$$0=\nabla\cdot\mathbf{u}. \tag{2.61}$$

The Weissenberg number gives the ratio of elastic forces to viscous forces. When $\mathrm{Wi}=0.0$, we are back in the regime of Newtonian fluid flow. When the Weissenberg number is large $\mathrm{Wi}\sim 1.0$, the fluid is strongly elastic and we see large concentrations of stress near regions of high strain rates. The Deborah number is defined as the ratio between the relaxation time and a relevant time scale. The Deborah number describes whether the fluid is acting more solid-like or fluid-like. While the Weissenberg and Deborah numbers have similar definitions and are frequently the same value, we stress that they describe different phenomenons. The Deborah number measures the *rate of change* of flow conditions. It is therefore related to the steadiness of flow conditions. The Weissenberg number, on the other hand, gives a notion of the steady elasticity of the fluid [13].

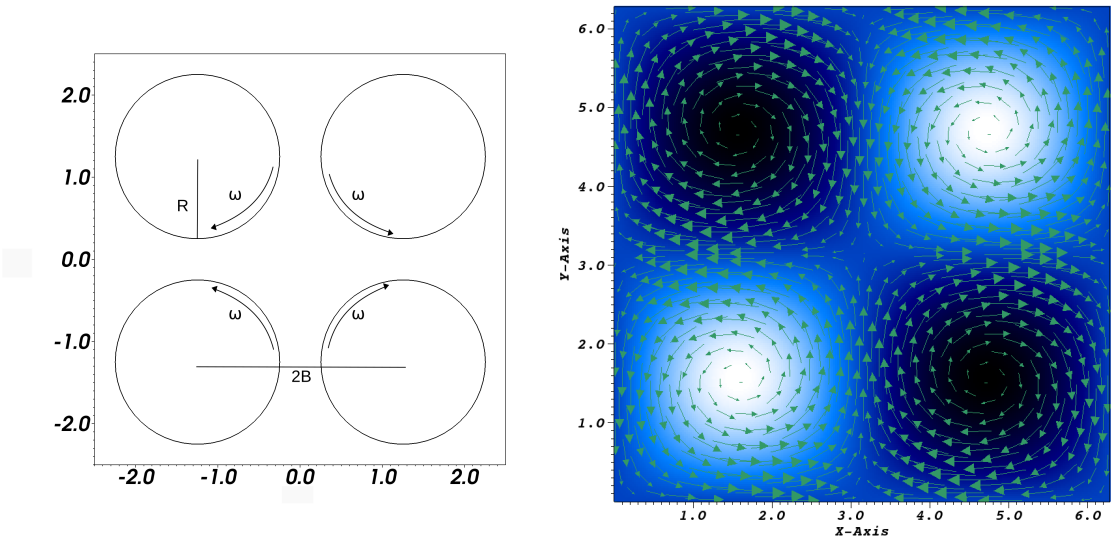

Figure 2.3: The four-roll mill physical setup (left) and the modelling approximation (right). The force vector field is shown in green arrows while the vorticity is background color map.

2.2.4 The Oldroyd-B Model

Here, we discuss the features of the Oldroyd-B model and provide a motivation for the contents of the remainder of the dissertation. As previously stated, the Oldroyd-B model allows for unbounded stress growth in the presence of extensional points. We can rationalize the unbounded growth by recalling the Oldroyd-B model models the polymers as Maxwell elements. If we pull on both sides of the elements hard enough and long enough, the element can get stretched arbitrarily long creating unbounded stress. To demonstrate this, we consider a four-roll mill flow. The four-roll mill experiment was devised by Taylor to generate various types of flows, from pure rotational flows to pure extensional flows by varying the rotation speed of the rollers. We model the four-roll mill geometry to create an artificial extensional point by applying a background force of $\mathbf{f}(\mathbf{x}, t) = (2\sin x \cos y, -2\cos x \sin y)^T$. This force creates a hyperbolic, stagnation point at the point $(0, 0)$. For a Newtonian fluid at the zero Reynolds number (Stokes flow), the velocity satisfies $\mathbf{u} = \frac{1}{2}\mathbf{f}$. Figure 2.3 shows the physical model as well as the computational setup. The force vector field is shown in green arrows while the background color map shows the vorticity of a Newtonian fluid under the forcing conditions. We fix the polymer viscosity to solvent viscosity $\frac{\mu_p}{\mu_n} = 0.5$ and vary the Weissenberg number.

Figure 2.4 shows the developments of the singularities that appear in the idealized four-roll mill geometry. For low Weissenberg numbers $\mathrm{Wi} \leq 0.5$, the conformation tensor remains smooth and

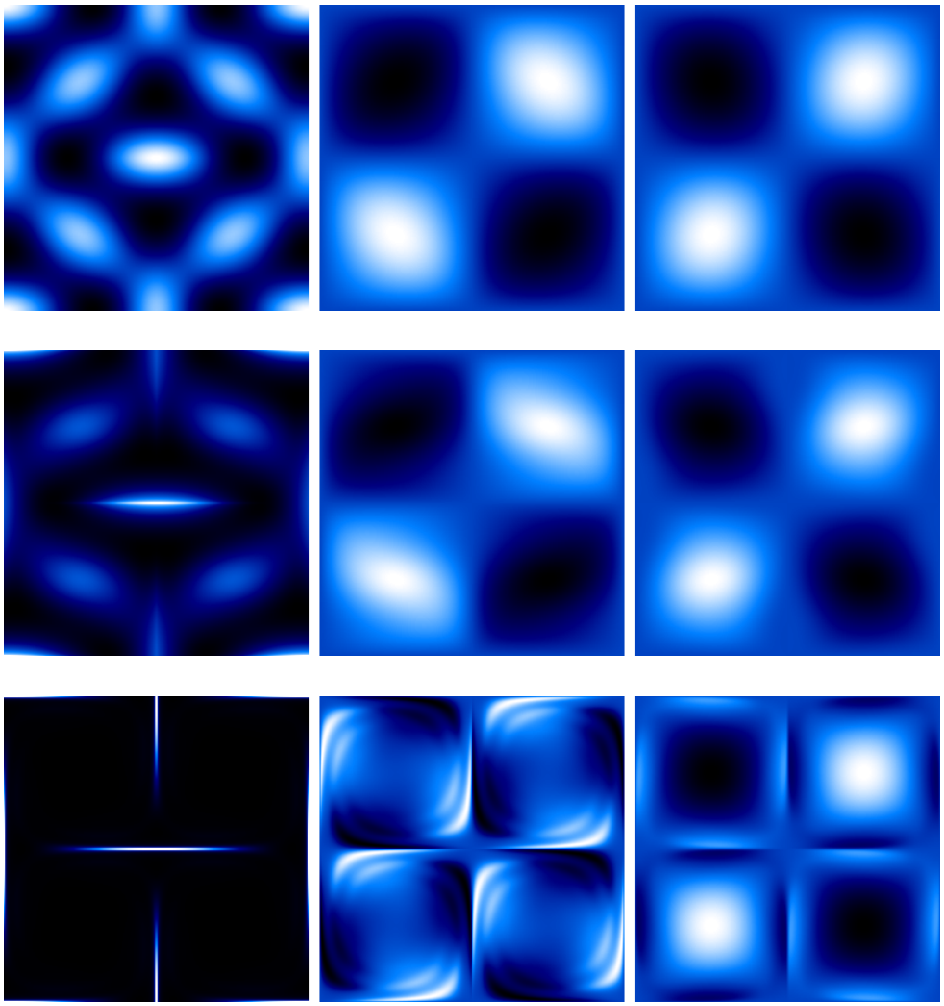

Figure 2.4: The trace and off-diagonal component of the conformation tensor, and the voriticity of the fluid for Weissenberg numbers 0.3 (top), 0.7 (middle), and 5.0 (bottom).

the vorticity remains slaved to the background force. As we increase the Weissenberg number to intermediate values $0.5 < \mathrm{Wi} \leq 0.9$, the conformation tensor forms a cusp near extensional points. However, the vorticity still remains slaved to the background force. As the Weissenberg number grows further $\mathrm{Wi} \geq 1.0$, the conformation tensor grows without bound, and the vorticity starts to show recirculation regions near the extensional points [7]. Figure 2.5 shows a slice of the trace of the conformation tensor along the line $x = \pi$. The cusps and exponential growth can be seen more readily.

This simple model is an excellent demonstration of the challenging features that exist with the Oldroyd-B model. The fine features shown in Figure 2.5 are common features in applications.

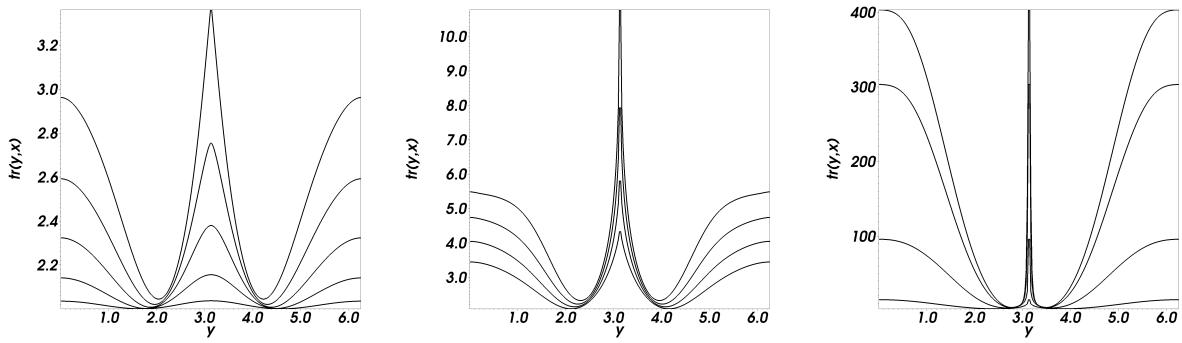

Figure 2.5: The trace of the conformation tensor along the line $x = \pi$ for Weissenberg numbers $0.1 - 0.5$, $0.6 - 0.9$, and $1.0 - 5.0$.

Extensional points occur frequently at fluid-structure interfaces because of the no-slip condition. This will generate large stress gradients that need to be resolved sufficiently to get accurate results. However, these fine features are localized to certain regions of the computational domain. Using a fine uniform discretization results in prohibitive computational costs, especially in three spatial dimensions. This motivates the remainder of the dissertation which involves using an adaptive algorithm that provides fine spatial resolution near regions of large stress gradients while using a coarse discretization where the flow is smooth.

CHAPTER 3

AN ADAPTIVE SOLVER FOR OLDROYD-B TYPE EQUATIONS

The localized features that are present in the complex fluid models discussed in the previous chapter need sufficient resolution to resolve. Using uniformly fine meshes throughout the computational domain, however, will quickly exceed the bounds of reasonable computational time, especially in three spatial dimensions. Adaptive mesh refinement (AMR) is a methodology for placing fine spatial resolution near key features while leaving a coarse description for the relatively smooth regions [14, 15]. Unstructured AMR methods are frequently used with finite element [16] or finite volume methods [17]. Unstructured grids typically use simplices to discretize the domain, which simplifies mesh generation as well as refinement; however, whenever a refine operation occurs, the local connectivity relations must be rebuilt. Unstructured grids are useful for complex geometries because they allow using a body-conforming grid for complex geometries. Structured AMR methods can only be used on domains that can be described using logically Cartesian parameterizations. However, they enable simple and efficient geometric multigrid level solvers to be used. They also allow for local time-stepping algorithms, although this concept will not be explored in this work. Although the major benchmark problems in the complex fluid literature are problems with complicated geometries, such as flow past a confined cylinder or flow through an abrupt contraction, we base our method on a structured AMR approach. In the next Chapter, we discuss how to model the complex geometries that form using the immersed boundary and immersed interface methods, and we discuss the ability of these methods to accurately resolve the dynamics near the boundary.

3.1 Discretization

We use a staggered-grid spatial discretization for the fluid velocity where the velocity field is discretized so that the components normal to cell faces (or, in three spatial dimensions, faces) are approximated at the centers of the edges (faces); see Figure 3.1. All other degrees of freedom (pressure and the components of the conformation tensor $\mathbb{C}$) are approximated at the cell centers. This type of discretization, originally called the marker and cell method (MAC), was developed by Harlow and

Welch [18] to solve the incompressible Navier-Stokes equations. We describe the discretization in two spatial dimensions as the extension to a third spatial dimension is straightforward. The equations are discretized on a hierarchical grid consisting of locally refined rectangular grid *patches* that are organized into a sequence of *patch levels* where all the patches in a given patch level use the same grid spacing. The computational domain consists of a rectangular box Ω such that the coarsest level is a uniform discretization of Ω using grid spacings of $\left(\Delta x_1^0, \Delta x_2^0\right)$. The different patch levels are defined so that patch level ℓ uses grid spacings $\left(\Delta x_1^\ell = \frac{\Delta x_1^0}{r^\ell}, \Delta x_2^\ell = \frac{\Delta x_2^0}{r^\ell}\right)$ where r^ℓ is the refinement ratio between levels. Although we do not denote it as such, there is no requirement that r remains constant in different dimensions or across levels. We denote the cell index of a given level ℓ to be $\mathbf{c}_{i,j}^\ell$ where the cell center is defined as $\mathbf{x}_{i,j} = \left((i+\frac{1}{2})\Delta x_0^\ell, (j+\frac{1}{2})\Delta x_1^\ell\right)$. The union of all grid patches that compose level ℓ form a region Ω^ℓ is defined so that it is completely contained within the coarser level, i.e. $\Omega^\ell \subset \Omega^{\ell-1}$.

Each patch level Ω^ℓ is padded with a layer of ghost cells to allow appropriate definitions of the needed stencils. There are three different types of ghost cells for a given level ℓ of the domain. (1) A ghost cell $\mathbf{c}_{i,j}^\ell$ could overlap the interior of another patch on the same level so that $\mathbf{c}_{i,j}^\ell \in \Omega^\ell$. For these ghost cells, we simply copy the corresponding value into the ghost cell. (2) A ghost cell $\mathbf{c}_{i,j}^\ell$ could overlap with the interior of patch level $\ell - 1$ so that $\mathbf{c}_{i,j}^\ell \in \Omega^{\ell-1}$. At these coarse-fine interfaces, the ghost cell value is interpolated using a specialized quadratic interpolation procedure. (3) A ghost cell $\mathbf{c}_{i,j}^\ell$ could be exterior to the domain so that $\mathbf{c}_{i,j}^\ell \notin \Omega$. In this case, we determine cell values by the physical boundary conditions for the problem. The ghost filling operations are described in detail in Griffith *et al˙* [19]. We note here that the boundary conditions for the conformation tensor are applied to each component separately only at inflow boundary conditions. At outflow boundaries, we use linear extrapolation on the components of the conformation tensor. Figure 3.2 shows ghost cells for (2) coarse-fine interfaces in blue and (3) physical boundaries in red. Case (1) typically occurs when different patches on the same level are distributed across processors and are thus not shown.

We use second-order finite differences for the gradient of the pressure and Laplacian and divergence of the velocity. The nonlinear term $\mathbf{u} \cdot \nabla \mathbf{u}$ is computed using a dimension-by-dimension PPM [20] approach based on the xsPPM7 scheme [21]. The nonlinear term $\mathbf{u} \cdot \nabla \mathbb{C}$ is computed using a wave propagation algorithm to be described in a later section. The velocity gradients in the stretching term $\left(\nabla \mathbf{u}\mathbb{C} + \mathbb{C}\nabla \mathbf{u}^T\right)$ are computed using second-order finite differences. Because the discretization

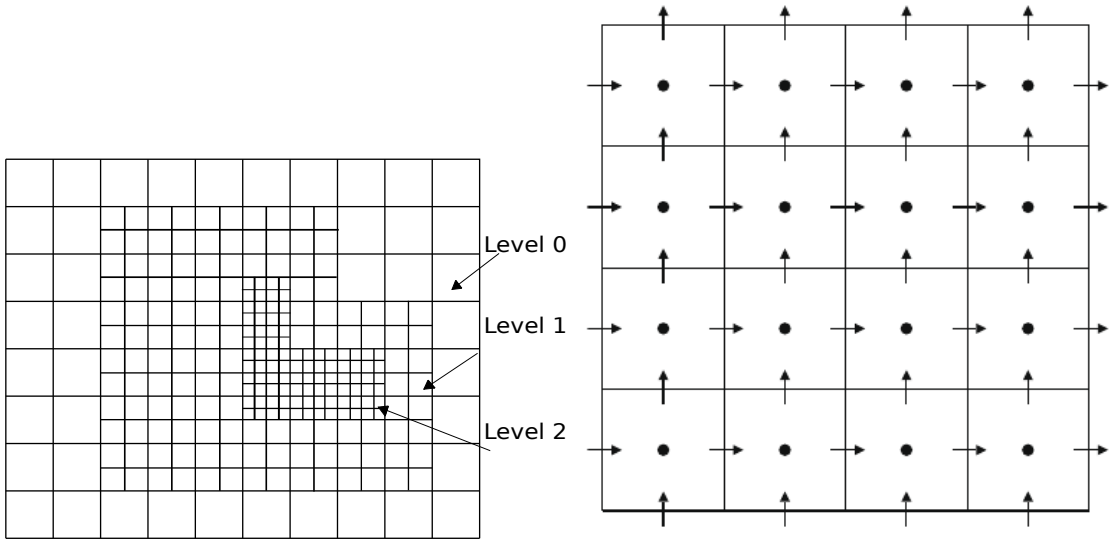

Figure 3.1: The patch hierarchy with a locally refined Cartesian grid. The refinement ratio here is $r = 2$. A sample patch in the patch hierarchy showing the MAC grid is to the right. The normal components of $\mathbf{u}$ are approximated at cell edges (faces in three spatial dimensions) and the components of $\mathbb{C}$ and the pressure are approximated at cell centers.

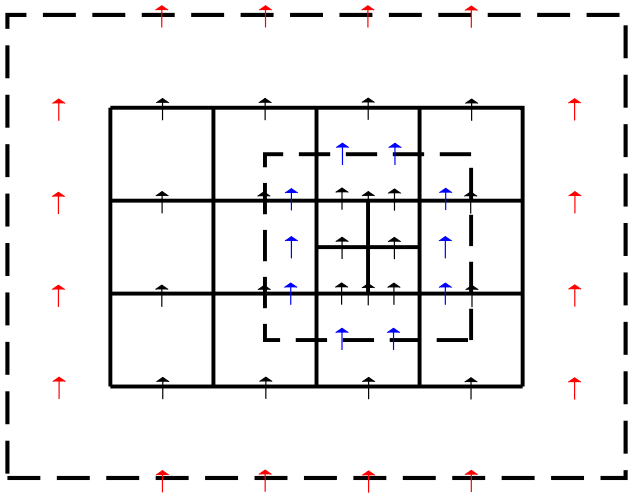

Figure 3.2: The ghost cells for the vertical component of the MAC grid. Two kinds of ghost cells are shown. The red arrows demonstrate physical boundaries that are filled by interpolation. The blue arrows demonstrate an interior ghost cell at a coarse-fine interface.

on any given patch level is the same, we will drop the superscript ℓ used to denote the level from now on.

3.1.1 Discrete Operators

Here we describe the discrete divergence of the conformation tensor, as well as the nonlinear terms $\mathbf{N}_2(\mathbf{u}, \mathbb{C}) \approx \mathbf{u} \cdot \nabla \mathbb{C}$ and $\mathbf{N}_3(\mathbf{u}, \mathbb{C}) \approx \mathbb{C}\nabla\mathbf{u}^T + \nabla\mathbf{u}\mathbb{C}$. For a description of the discretization of $\mathbf{N}_1(\mathbf{u}, \mathbf{u}) \approx \mathbf{u} \cdot \nabla \mathbf{u}$ and the divergence and gradients of the velocity, see Griffith [22].

Discrete Divergence We use second-order finite differences to compute $\nabla_h \cdot \mathbb{C}$. Note that the discrete operator must evaluate the normal components of the divergence of the conformation tensor on cell edges. We first interpolate the off-diagonal components of the conformation tensor onto cell edges (faces in three spatial dimensions)

$$(C_{xy})_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{4}\left((C_{xy})_{i.j} + (C_{xy})_{i+1,j} + (C_{xy})_{i,j+1} + (C_{xy})_{i+1,j+1}\right). \tag{3.1}$$

We can now approximate the divergence of the conformation tensor at cell centers via

$$((\nabla_h \cdot \mathbb{C})_x)_{i+\frac{1}{2},j} = \frac{(C_{xx})_{i+1,j} - (C_{xx})_{i,j}}{\Delta x} + \frac{(C_{xy})_{i+\frac{1}{2},j+1} - (C_{xy})_{i+\frac{1}{2},j}}{\Delta y}, \tag{3.2}$$

$$\left((\nabla_h \cdot \mathbb{C})_y\right)_{i,j+\frac{1}{2}} = \frac{(C_{xy})_{i+1,j+\frac{1}{2}} - (C_{xy})_{i,j+\frac{1}{2}}}{\Delta x} + \frac{(C_{yy})_{i,j+1} - (C_{yy})_{i,j}}{\Delta y}. \tag{3.3}$$

Advection We use a second-order wave propagation algorithm to compute $\mathbf{u} \cdot \nabla \mathbb{C}$. We first note that our time discretization described in a later section allows us to treat the velocity $\mathbf{u}$ as a given parameter instead of a state variable. This simplifies our solution to the Riemann problem needed to use the wave propagation algorithm. For details on the wave propagation algorithm see Ketcheson *et al.* [23]. We briefly highlight the process here. First, consider the hyperbolic linear initial value problem in one spatial dimension,

$$q_t + Aq_x = 0, \quad q(x,0) = \dot{q}(x), \tag{3.4}$$

where $A = R\Lambda R^{-1}$ is an $m \times m$ diagonalizable matrix with real eigenvalues. Then we can decouple the linear system by substituting $w = R^{-1}q$ into equation (3.4) to get

$$w_t + \Lambda w_x = 0, \quad w(x,0) = \dot{w}(x). \tag{3.5}$$

We can now solve the p^{th} equation to get $w^p(x,t) = w^p(x - \lambda^p t, 0)$ where λ^p is the p^{th} eigenvalue of A. Finally, we can recover the original solution

$$q(x,t) = Rw(x,t) = \sum_{p=1}^{m} \dot{w}^p(x - \lambda^p t) r^p, \tag{3.6}$$

where r^p is the p^{th} eigenvector of A. We can, therefore, view $q(x,t)$ as a superposition of waves each propagating at a separate speed given by the eigenvalues of A. The *Riemann problem* for a hyperbolic problem consists of equation (3.4) with initial data that consists of a single jump

$$\dot{q}(x) = \begin{cases} q_l & \text{if } x < 0 \\ q_r & \text{if } x \geq 0. \end{cases} \tag{3.7}$$

For this problem, each wave is the jump discontinuity along the ray $x = \lambda^p t$. Now consider the one spatial dimension equation

$$q_t + A(x) q_x = 0 \tag{3.8}$$

with $A(x)$ a constant function within each cell. Integrating equation (3.8) over a cell $C_i = \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right]$ and a small time step Δt yields

$$Q_i\left(t_0 + \Delta t\right) - Q_i\left(t_0\right) = -\frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} A_i \frac{\partial}{\partial x} q\left(x, t_0 + \Delta t\right) dx, \tag{3.9}$$

where $Q_i = \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} q(x,t)$ is the cell average over cell C_i. Now suppose in each cell we can diagonalize $A(x_i) = A_i = R_i \Lambda_i R_i^{-1}$. We denote the minimum of the wave speeds $s^L = \min_p \lambda^p$ and the maximum wave speeds $s^R = \max_p \lambda^p$. Note that $s^L \Delta t$ is the furthest a traveling wave can move to in the negative x direction while $s^R \Delta t$ is the furthest a wave can move in the positive x direction.

If we approximate $q(x,t) = \tilde{q}(x,t)$ with a piecewise constant function, such that

$$\tilde{q}(x,t) = Q_i \text{ for } x \in [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}], \tag{3.10}$$

then the linear system in equation (3.8) consists of a series of local Riemann problems at each cell interface, which we can express as a set of waves

$$Q_i - Q_{i-1} = \sum_p \lambda^p_{i-\frac{1}{2}} W^p_{i-\frac{1}{2}}, \tag{3.11}$$

where $W^p_{i-\frac{1}{2}}$ is an appropriately defined wave at the cell interface $x_{i-\frac{1}{2}}$. This allows us to decompose the integral in equation (3.9) into three separate integrals

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} A_i \frac{\partial}{\partial x}\tilde{q}\,(x, t_0 + \Delta t)\,dx = \int_{x_{i-\frac{1}{2}}}^{x_{i-\frac{1}{2}}+s^R\Delta t} A_i \frac{\partial}{\partial x}\tilde{q}\,(x, t_0 + \Delta t)\,dx \tag{3.12}$$

$$+ \int_{x_{i+\frac{1}{2}}+s^L\Delta t}^{x_{i+\frac{1}{2}}} A_i \frac{\partial}{\partial x}\tilde{q}\,(x, t_0 + \Delta t)\,dx \tag{3.13}$$

$$+ \int_{x_{i-\frac{1}{2}}+s^R\Delta t}^{x_{i+\frac{1}{2}}+s^L\Delta t} A_i \frac{\partial}{\partial x}\tilde{q}\,(x, t_0 + \Delta t)\,dx. \tag{3.14}$$

We can evaluate these integrals by using the solution to the Riemann problem from equation (3.11).

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} A_i \frac{\partial}{\partial x}\tilde{q}\,(x, t_0 + \Delta t)\,dx = \Delta t \sum_{p=1}^{m} \left(\lambda_{i-\frac{1}{2}}\right)^+ W^p_{i-\frac{1}{2}} \tag{3.15}$$

$$+ \Delta t \sum_{p=1}^{m} \left(\lambda_{i+\frac{1}{2}}\right)^- W^p_{i+\frac{1}{2}} \tag{3.16}$$

$$+ 0, \tag{3.17}$$

where $(\lambda)^+ = \max(0, \lambda)$ and $(\lambda)^- = \min(0, \lambda)$ and the last integral is 0 because $\tilde{q}$ is constant within that fraction of the cell. Taking the limit as $\Delta t \to 0$, we get the semi-discrete system

$$\frac{\partial Q_i}{\partial t} = \left(\Delta t \sum_{p=1}^{m} \left(\lambda_{i-\frac{1}{2}}\right)^+ W^p_{i-\frac{1}{2}} + \Delta t \sum_{p=1}^{m} \left(\lambda_{i+\frac{1}{2}}\right)^- W^p_{i+\frac{1}{2}} \right). \tag{3.18}$$

If we replace the piecewise constant approximation with a p^{th} order polynomial reconstruction, then the integral in equation (3.14) is no longer zero. In general, this integral must be evaluated using a quadrature rule. The final wave propagation scheme is given as

$$\frac{\partial Q_i}{\partial t} = -\frac{1}{\Delta x}\left(A^+\Delta q_{i-\frac{1}{2}} + A^-\Delta q_{i+\frac{1}{2}} + \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} A_i\tilde{q}_x dx\right). \tag{3.19}$$

Because we can solve the Riemann problem exactly, the last integral in equation (3.19) reduces to $A\Delta q_i = A^+\Delta q_i + A^-\Delta q_i$, a fictitious 'internal' Riemann problem for Cartesian grid cell i. In two spatial dimensions, we proceed in a dimension by dimension manner. For the equation

$$q_t + A(x,y)q_x + B(x,y)q_y = 0, \tag{3.20}$$

the algorithm becomes

$$\frac{\partial Q_{ij}}{\partial t} = -\frac{1}{\Delta x\Delta y}\left(A^-\Delta q_{i+\frac{1}{2},j} + A^+\Delta q_{i-\frac{1}{2},j} + A\Delta q_{i,j} + B^-\Delta q_{i,j+\frac{1}{2}} + B^+\Delta q_{i,j-\frac{1}{2}}\right). \tag{3.21}$$

We note that for higher than second-order accuracy, the fluctuations would require integrals over cell edges.

To reconstruct the values of $\mathbb{C}$ at cell edges, we use a fifth-order WENO reconstruction algorithm [24], which we overview in the appendix. We note that although the above algorithm and original wave propagation paper advances the cell averages Q_{ij}, we solve for the values at the cell centers, which is a second-order approximation to the cell average. Therefore the overall order of accuracy remains second order.

Stretching The last term we have to discretize is the stretching term $\mathbf{N}_3(\mathbf{u},\mathbb{C}) \approx \mathbb{C}\nabla\mathbf{u}^T + \nabla\mathbf{u}\mathbb{C}$. Here, we use second-order finite differences.

The normal velocity terms are standard centered differences:

$$\left(\frac{\partial u}{\partial x}\right)_{i,j} = \frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x}, \tag{3.22}$$

$$\left(\frac{\partial v}{\partial y}\right)_{i,j} = \frac{v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{\Delta y}. \tag{3.23}$$

For the other terms, we need to use a wider stencil, so the derivatives become

$$\left(\frac{\partial u}{\partial y}\right)_{i,j} = \frac{u_{i+\frac{1}{2},j+1} + u_{i-\frac{1}{2},j+1} - u_{i+\frac{1}{2},j-1} - u_{i-\frac{1}{2},j-1}}{4\Delta x}, \tag{3.24}$$

$$\left(\frac{\partial v}{\partial x}\right)_{i,j} = \frac{v_{i+1,j+\frac{1}{2}} + v_{i+1,j-\frac{1}{2}} - v_{i-1,j+\frac{1}{2}} - v_{i-1,j-\frac{1}{2}}}{4\Delta y}. \tag{3.25}$$

Refinement Criteria At a specified interval, we undergo a regridding process. During the regridding operation, we tag cells that need to be refined. Each level has a threshold based on the absolute or relative magnitude of the divergence of the conformation tensor, above which a cell is tagged for refinement. To ensure that enough cells exist in a level to accurately capture the features, we use a large tag buffer to tag adjacent cells. For higher Reynolds number simulations, we also tag cells based on vorticity thresholds.

3.1.2 Time Discretizations

We use an explicit time discretization that is motivated by the implicit midpoint rule:

$$\rho\left(\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \mathbf{N}_1\right) = -\nabla_h p^{n+\frac{1}{2}} + \mu_s \nabla_h^2 \mathbf{u}^{n+\frac{1}{2}} + \frac{\mu_p}{\lambda}\nabla_h \cdot \mathbb{C}^{n+\frac{1}{2}} + \mathbf{f}^{n+\frac{1}{2}}, \tag{3.26}$$

$$\nabla_h \cdot \mathbf{u}^{n+1} = 0, \tag{3.27}$$

$$\frac{\mathbb{C}^{n+1} - \mathbb{C}^n}{\Delta t} + \mathbf{N}_2 - \mathbf{N}_3 = \mathbf{S}\left(\mathbb{C}^{n+\frac{1}{2}}\right), \tag{3.28}$$

where $\mathbf{u}^{n+\frac{1}{2}} = \frac{1}{2}\left(\mathbf{u}^n + \mathbf{u}^{n+1}\right)$ and $\mathbf{N}_1, \mathbf{N}_2$, and $\mathbf{N}_3$ are discretizations of the nonlinear terms described previously.

Instead of solving the fully coupled problem, we solve two separate equations and cycle between solving the two. The first problem is

$$\rho\left(\frac{\mathbf{u}^{n+1,k+1} - \mathbf{u}^n}{\Delta t} + \mathbf{N}_1\left(\mathbf{u}^{n+\frac{1}{2},k}\right)\right) = -\nabla_h p^{n+\frac{1}{2},k+1} + \mu_s \nabla_h^2 \mathbf{u}^{n+\frac{1}{2},k+1} + \frac{\mu_p}{\lambda}\nabla_h \cdot \mathbb{C}^{n+\frac{1}{2},j} + \mathbf{f}^{n+\frac{1}{2}} \tag{3.29}$$

$$\nabla_h \cdot \mathbf{u}^{n+1,k+1} = 0, \tag{3.30}$$

where $\mathbb{C}^{n+\frac{1}{2},j}$ is the most recent approximation to $\mathbb{C}^{n+\frac{1}{2}}$. The second problem is

$$\frac{\mathbb{C}^{n+1,j+1}-\mathbb{C}^{n}}{\Delta t}+\mathbf{N}_2\left(\mathbb{C}^{n+\frac{1}{2},j}\right)-\mathbf{N}_3\left(\mathbf{u}^{n+\frac{1}{2},k},\mathbb{C}^{n+\frac{1}{2},j}\right)=\mathbf{S}\left(\mathbb{C}^{n+\frac{1}{2},j}\right), \tag{3.31}$$

where $\mathbf{u}^{n+\frac{1}{2},k}$ is the most recent approximation to $\mathbf{u}^{n+\frac{1}{2}}$. Note that when the relaxation function $\mathbf{f}$ is chosen to include stress diffusion, we treat the resulting Laplacian implicitly. At the end of each iteration, we project $\mathbb{C}$ to the closest symmetric non-negative definite matrix in the two-norm [25]. Let $\mathbb{C}$ have the diagonalization $\mathbb{C}=R\Lambda R^{-1}$. Then we claim that the closest symmetric non negative-definite matrix is $\mathbb{C}^{+}=R\Lambda^{+}R^{-1}$ where Λ^{+} is the matrix where the positive eigenvalues are left alone, and the negative eigenvalues are replaced by zero. Suppose that $\tilde{\mathbb{C}}$ is a symmetric non negative-definite matrix that is closer to $\mathbb{C}$ in the two-norm than $\mathbb{C}^{+}$. Then

$$\left|\mathbb{C}-\tilde{\mathbb{C}}\right|_2\leq\left|\mathbb{C}-\mathbb{C}^{+}\right|=\left|\mathbb{C}^{-}\right|=\left|\lambda^{-}\right|, \tag{3.32}$$

where $\mathbb{C}^{-}=\mathbb{C}-\mathbb{C}^{+}$ and λ^{-} is the largest negative eigenvalue in absolute value of $\mathbb{C}$. Let $\mathbf{v}$ be the corresponding normalized eigenvector such that

$$\mathbf{v}^{T}\mathbb{C}\mathbf{v}=\lambda^{-}. \tag{3.33}$$

Then we can bound the quadratic form $\left|\mathbf{v}^{T}\left(\mathbb{C}-\tilde{\mathbb{C}}\right)\mathbf{v}\right|$ from below by

$$\left|\mathbf{v}^{T}\left(\mathbb{C}-\tilde{\mathbb{C}}\right)\mathbf{v}\right|=\left|\lambda^{-}-\mu\right|,\quad \mu>0 \tag{3.34}$$

$$>\left|\lambda^{-}\right|. \tag{3.35}$$

We can also bound the same quadratic form from above by

$$\left|\mathbf{v}^{T}\left(\mathbb{C}-\tilde{\mathbb{C}}\right)\mathbf{v}\right|\leq|\mathbf{v}|\left|\left(\mathbb{C}-\tilde{\mathbb{C}}\right)\mathbf{v}\right| \tag{3.36}$$

$$\leq\left|\mathbb{C}-\tilde{\mathbb{C}}\right| \tag{3.37}$$

$$\leq\left|\lambda^{-}\right|, \tag{3.38}$$

which is a contradiction of the previous inequality. Therefore, $\mathbb{C}^{+}$ is the closest symmetric non-

negative definite matrix to $\mathbb{C}$ in the two-norm.

We use a fixed number of fixed-point iterations n_{ns} for the first problem and n_{av} for the second. In general, we can use a different number of iterations for each equation. After iterating, the final values for the variables at the new time step are $\mathbf{u}^{n+1} = \mathbf{u}^{n+1,n_{\text{ns}}}$, $p^{n+\frac{1}{2}} = p^{n+\frac{1}{2},n_{\text{ns}}}$, and $\mathbb{C}^{n+1} = \mathbb{C}^{n+1,n_{\text{av}}}$.

The resulting linear systems for the conservation of mass and momentum equations are solved using a flexible GMRES algorithm with a projection method as a preconditioner [22].

3.1.3 Implementation

The solver is built into IBAMR [19], an open-source adaptive and distributed memory parallel implementation of the immersed boundary method. IBAMR is built on top of SAMRAI [26] which provides infrastructure for structured adaptive mesh refinement. The resulting linear systems are solved using implementations provided by PETSc [27].

3.2 Results

3.2.1 Manufactured Solutions

We test the implementation using the method of manufactured solutions. We add an extra forcing term to the original equation for the conformation tensor:

$$\overset{\triangledown}{\mathbb{C}} = \mathbf{S}(\mathbb{C}) + \mathbf{h}(\mathbf{x}, t). \tag{3.39}$$

We now choose the forcing terms $\mathbf{f}(\mathbf{x}, t)$ and $\mathbf{h}(\mathbf{x}, t)$ to drive the solution to a predetermined function. We choose as the exact solution for the velocity and pressure to be the well known Taylor vortices, which are exact solutions to the viscous Navier-Stokes equations

$$u = 1 - 2\cos(2\pi(x-t))\sin(2\pi(y-t))e^{-8\pi^2\mu t} \tag{3.40}$$

$$v = 1 + 2\sin(2\pi(x-t))\cos(2\pi(y-t))e^{-8\pi^2\mu t} \tag{3.41}$$

$$p = -(\cos(4\pi(x-t)) + \cos(4\pi(y-t))e^{-16\pi^2\mu t}. \tag{3.42}$$

For the extra stress, we choose

$$\mathbb{C} = (\sin(2\pi(x-t))\sin(2\pi(y-t)) + 2)\mathbb{I}, \tag{3.43}$$

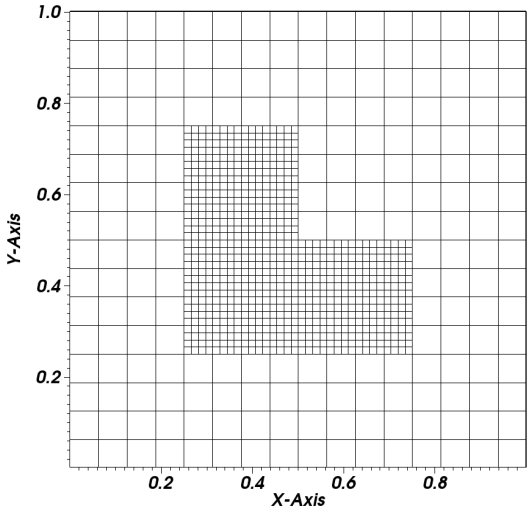

Figure 3.3: Static grid used for manufactured solutions.

where $\mathbb{I}$ is the identity tensor. We set the constants $\rho = 1$, $\mu = \mu_p = 0.01$, and $\lambda = 0.1$. This corresponds to a Reynolds number of $\mathrm{Re} \approx 100$ and a Weissenberg number of $\mathrm{Wi} \approx 0.1$. The domain is the unit square with periodic boundary conditions. We run the simulation with a locally refined L-shaped region as shown in Figure 3.3. The refined region has a refinement ratio of 4 relative to the coarsest level. The simulations are run such that the coarsest grid uses a $N \times N$ grid with $N = 16$, 32, 64, and 128. The timestep is chosen so that simulations are run at an advective CFL number of approximately 0.3. Figure 3.4 shows the convergence rates for the method of manufactured solutions. We achieve second-order accuracy for the velocity and components of the conformation tensor.

3.2.2 Pipe flow

There exists an analytic solution for Poiseuille flow. We set up a computational domain of $\Omega = [0, 3] \times [0, 1]$. We set the initial conditions to be at rest $\mathbf{u} = 0, \mathbb{C} = \mathbb{I}$, and at time $t = 0$, we

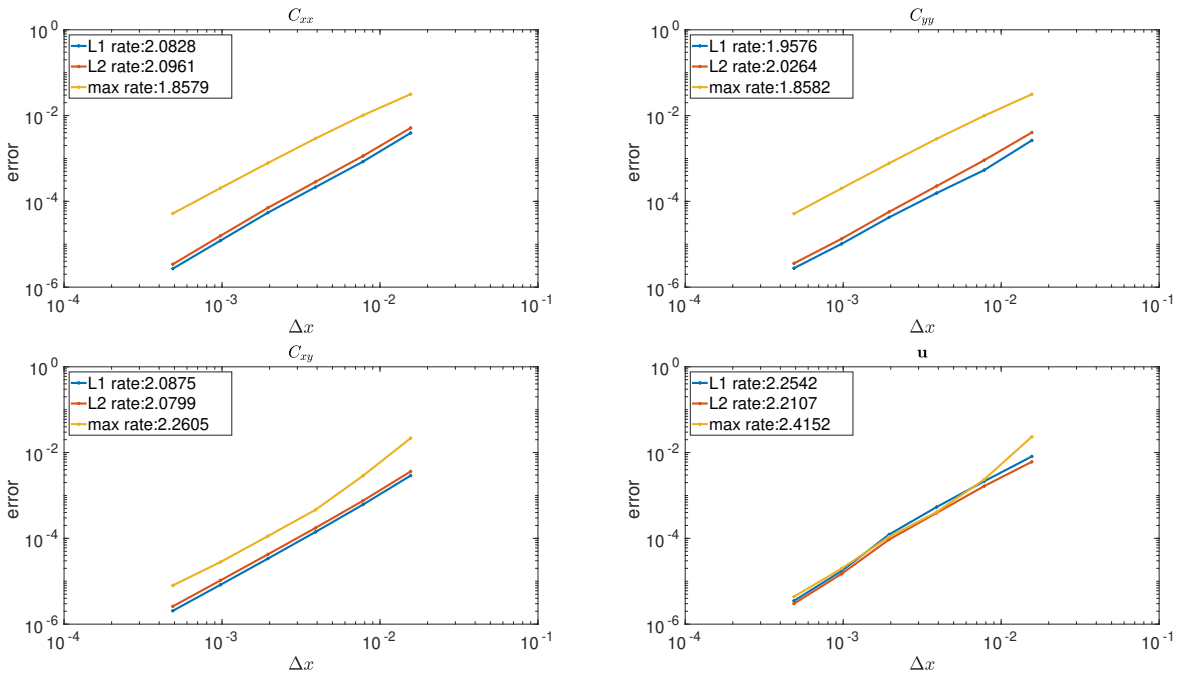

Figure 3.4: Convergence plots for method of manufactured solutions. Here, Δx refers to the finest grid spacing.

specify the boundary conditions to be

$$u = \frac{\Delta p}{2\left(\mu + \mu_p\right)} y(1-y) \frac{\tanh\left(\frac{t-t_h}{\tau}\right) + \tanh\left(\frac{t_h}{\tau}\right)}{1 + \tanh\left(\frac{t_h}{\tau}\right)}, \tag{3.44}$$

$$v = 0, \tag{3.45}$$

$$C_{xx} = 1 + 2\left(\lambda \frac{\partial u}{\partial y}\right)^2, \tag{3.46}$$

$$C_{yy} = 1, \tag{3.47}$$

$$C_{xy} = \lambda \frac{\partial u}{\partial y}, \tag{3.48}$$

where Δp is a given pressure differential and t_h and τ are parameters to control the rate at which the velocity inflow is prescribed. Here, we set $t_h = 0.5$ and $\tau = 0.125$. The above initial conditions approach the exact solution for Poiseuille flow as $t \to \infty$. We note that at the outflow boundary we use linear extrapolation on the components of the conformation tensor. The simulation is run with $\mu = \mu_p = 0.5$, $\rho = 1.0$, and $\lambda = 0.1$ which corresponds to the Reynolds number $\mathrm{Re} = 2$ and Weissenberg number $\mathrm{Wi} = 0.1$. We use a grid with a locally refined region with refinement ratio 2

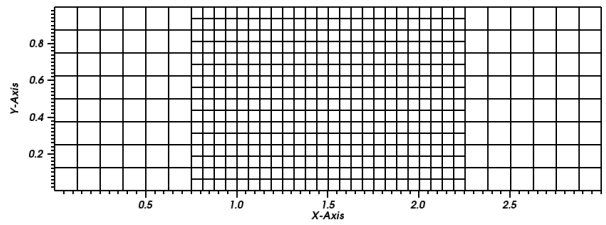

Figure 3.5: Static grid used for Poiseuille flow.

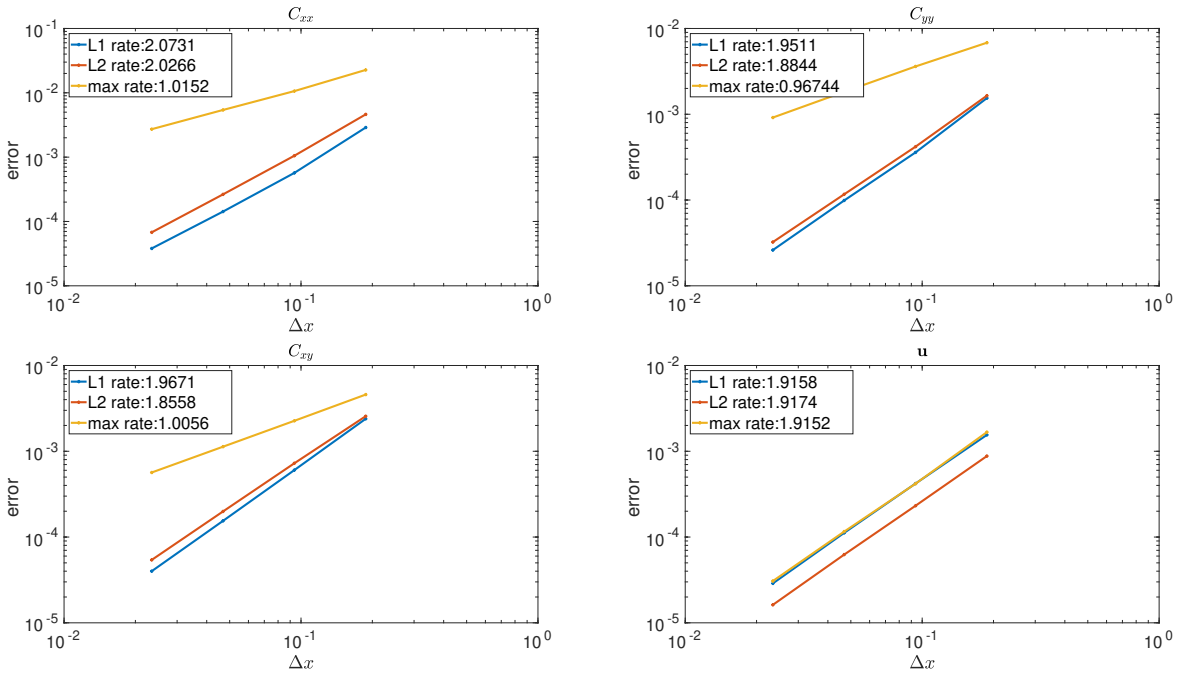

Figure 3.6: Convergence rates for Poiseuille flow.

located in the center of the domain, shown in Figure 3.5.

Convergence rates are presented in Figure 3.6. We see second-order convergence rates in all norms except the max norm for the components of the conformation tensor. We note that the reduction to first-order accuracy lies at the physical corners of the computational domain. In order to restore second-order accuracy at these points, we need to use a more careful treatment of the boundary conditions. Due to the hyperbolic nature of the equations, the error advects into the domain, causing first-order accuracy in the interior of the domain as well. However, these points do not spoil the overall convergence rates for the L^1 and L^2 norms.

3.2.3 Lid driven cavity

We now turn to the classical problem of lid-driven cavity flow. In this geometry, fluid motion is induced in a box by the translation of one of the walls. Under Stokes flow conditions, where $\mathrm{Re} \to 0$, a main recirculation region appears in the middle of the domain that is symmetric across the center line of the domain. Two smaller corner eddies are induced at the bottom corners of the box. As the aspect ratio $\Lambda = \frac{H}{L}$ grows, these corner eddies grow and eventually become a second major recirculation region. Under viscoelastic fluid conditions, the left-right symmetry condition is broken, and the central recirculation region moves closer to the top right, as demonstrated experimentally [9]. Computationally, a major issue is the corner singularities where the fluid velocity jumps from no-slip conditions to a constant value of U. This yields infinite shear rates at the corners that, for Oldroyd-B type viscoelastic models, will yield stress singularities. The effective Deborah number $\mathrm{De} = \frac{\lambda}{\tau}$ where τ is a characteristic residence time is infinite near the corner singularity. A common modelling fix to this issue is to use a quartic polynomial approximation to the wall velocity [28].

Here, we restrict ourselves to the square lid-driven cavity where $\Lambda = 1$. At the three stationary walls, we use zero slip boundary conditions $\mathbf{u} = 0$ and zero flux boundaries for the components of the extra stress. Following Sousa *et al*˙ [29], at the moving wall, we specify,

$$\mathbf{u}(x,1) = \begin{cases} \frac{1}{0.2^2 0.8^2} U x^2 (1-x)^2 & \text{if } 0 \le x \le 0.2 \\ U & \text{if } 0.2 < x < 0.8 \\ \frac{1}{0.2^2 0.8^2} U x^2 (1-x)^2 & \text{if } 0.8 \le x \le 1. \end{cases} \tag{3.49}$$

This form of the inlet velocity maintains the lid velocity of U across 60% of the lid. While the velocity and velocity gradient vanish at the corners, the velocity gradient is not continuous at the points of change between the polynomial and constant velocity profile. We adopt the standard definition of the Weissenberg number to use the *peak* velocity. However, we note that when using the *mean* velocity across the domain, our Weissenberg number is approximately 1.4 times higher than corresponding literature values that use the quartic inflow condition. The simulations are initialized with a coarse grid of 64 points in each dimension but are run with a maximum of four levels each with a refinement ratio of two. If the entire domain were refined to this level, this would result in a grid with 512 points in each dimension. Every eight timesteps, we undergo a regridding operation

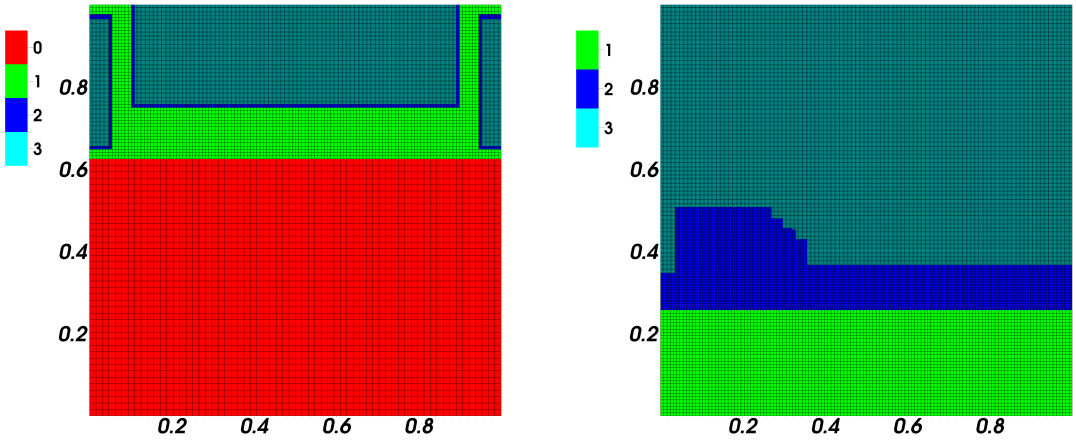

Figure 3.7: The locally refined Cartesian grid showing the different levels and grid resolution at the final time, for the Newtonian case (left) and the case with an Oldroyd-B fluid with Weissenberg number $\mathrm{Wi} = 1$ (right).

where cells are tagged for refinement based on one of two criteria. We also tag a cell for refinement if the magnitude of the divergence of the polymeric stress tensor is greater than 1.0 for the first level, 2.5 for the second level, and 5.0 for the finest level. These values were chosen based on uniform simulations on a coarse grid to provide sufficient resolution at the boundaries while maintaining a coarse grid near the bottom of the domain. We note that for every simulation in which $\mathrm{Wi} > 0$, the coarsest level is always completely covered by a finer level. In the absence of fluid elasticity, we refine based on the relative magnitude of the vorticity. Figure 3.7 shows the meshes at the final time step. We run the simulations to a final time of $T = 20$ with a time step chosen to maintain a CFL number of 0.1.

Figure 3.8 shows the trace of the conformation tensor. We see exponential growth in the trace of the conformation tensor near the discontinuities in the velocity gradient. Figure 3.8 also details the vertical component of the velocity across the line $y = 0.75$. In the Newtonian fluid, we see symmetry across the $x = 0.5$ line. However, as we increase the Weissenberg number, the vertical velocity sharply increases in the right half of the domain, while only slowly decreases in the left half of the domain.

As done in Sousa *et al*˙ [29], we use the flow-type parameter $\xi = \frac{|\mathbf{D}|-|\mathbf{\Omega}|}{|\mathbf{D}|+|\mathbf{\Omega}|}$ to classify the flow locally where $|\mathbf{D}|$ and $|\mathbf{\Omega}|$ are the magnitudes of the strain rate tensor and vorticity tensor, $|\mathbf{D}| = \sqrt{\frac{1}{2}\mathbf{D} : \mathbf{D}}$

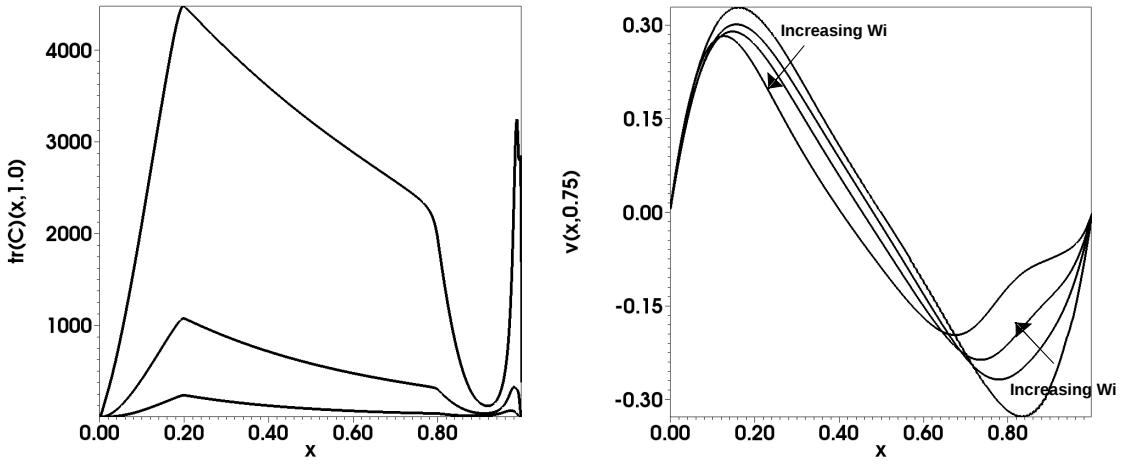

Figure 3.8: Plots of the trace of the conformation tensor at the top of the box (left) and the vertical velocity at the line $y = 0.75$ (right) as the Weissenberg number increases.

and $|\mathbf{\Omega}| = \sqrt{\frac{1}{2}\mathbf{\Omega} : \mathbf{\Omega}}$. $\xi = -1$ corresponds to a pure extensional flow, $\xi = 0$ corresponds to a pure shear flow, and $\xi = 1$ corresponds to solid-body rotation. Figure 3.9 shows the flow types. As we increase the Weissenberg number, the rotation region migrates towards the top of the box. We also see the growth of an additional shearing region near the corner singularity.

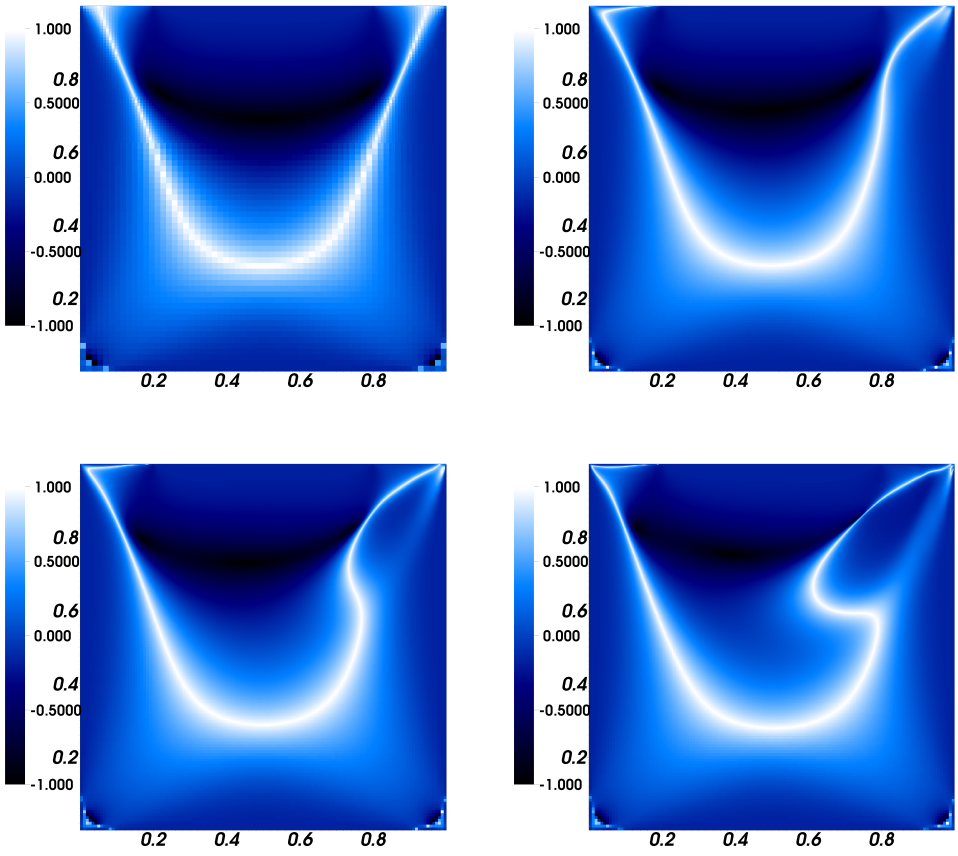

Figure 3.9: Plots of the flow type parameter ξ for Wi = 0.0 (top left), Wi = 0.3 (top right), Wi = 0.5 (bottom left), and Wi = 1.0 (bottom right). A realization of the AMR grid structure is shown in Figure 3.7.

CHAPTER 4

APPLICATIONS WITH AN IMMERSED INTERFACE MODEL

Fluid-structure interactions is a key component in the study of biological fluid flow. Swimming organisms push off the fluid in order to propel themselves forward [30]. Heart valves open and close due to the pressure of the blood flowing against them [31, 32]. Even the complex polymeric fluids models we have derived are microscopic fluid-structure interaction problems [11, 33]. This Chapter provides an overview of both the immersed boundary (IB) method and the immersed interface method (IIM) which are two different ways of modeling fluid-structure interaction. We then compare the two approaches to the classical problem for flow past a confined cylinder. We then follow up on the four-roll mill problem first introduced in the previous chapters.

4.1 The Immersed Boundary Method

The immersed boundary method was first described in 1972 to study blood flow around heart valve leaflets by Charles Peskin [34]. It has since been used to model a plethora of biomechanics problems. Some applications include heart dynamics [35], insect flight [36], aquatic locomotion [37], and cellular mechanics [3]. The fundamental approach has since been extended to form the basis of different methods, such as the immersed finite element method [38], the fictitious domain method [39], and the ghost-cell IB method [40]. The strength of the IB method is that it avoids mesh-conforming discretizations, which makes this method particularly appealing to problems with complex geometries. The IB formulation uses a single momentum equation for both the fluid and the structure expressed in a Eulerian form. The structural mechanics are then described in a Lagrangian framework approximated on a curvilinear mesh. Interaction between the two meshes are mediated with integral equations.

The IB method is agnostic to the fluid model. However, a key aspect of the IB formulation is that the momentum equation is valid both inside and outside the structure, see Figure 4.1. This modeling assumption causes structures to adopt the same fluid features as the outer fluid [41]. In some applications, these modelling assumptions can be seen as beneficial, such as in modelling

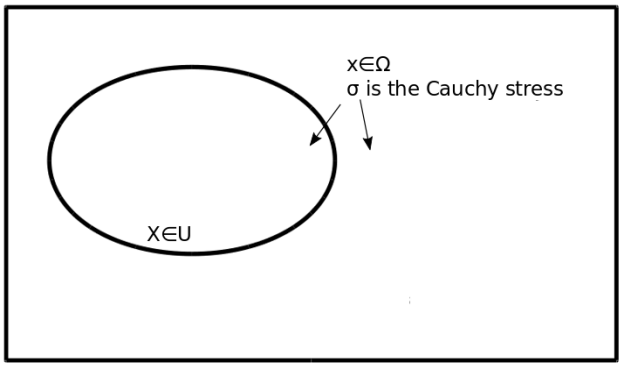

Figure 4.1: In the immersed boundary method, the physical domain Ω consists of a fluid that exists throughout the entire domain with a corresponding Cauchy stress $\boldsymbol{\sigma}$ and a thin immersed structure described by the curve U.

cellular mechanics where the inside of the cell is fluid-like as well. The momentum equation that we impose in the Eulerian frame is coupled with an incompressibility condition

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\cdot\mathbf{u}\right)=\nabla\cdot\boldsymbol{\sigma}+\mathbf{f} \tag{4.1}$$

$$\nabla\cdot\mathbf{u}=0, \tag{4.2}$$

where $\mathbf{u}$ is the fluid velocity, $\boldsymbol{\sigma}$ is the Cauchy stress of the fluid, and $\mathbf{f}$ is the force per unit volume applied by the structure to the fluid. The interactions between the fluid and immersed structure are given by

$$\mathbf{f}\left(\mathbf{x},t\right)=\int_{\Omega}\mathbf{F}\left(\mathbf{X},t\right)\delta\left(\mathbf{x}-\boldsymbol{\chi}\left(\mathbf{X},t\right)\right)d\mathbf{X} \tag{4.3}$$

$$\frac{\partial\boldsymbol{\chi}\left(\mathbf{X},t\right)}{\partial t}=\mathbf{U}\left(\boldsymbol{\chi}(\mathbf{X},t)\,,t\right)=\int_{U}\mathbf{u}\left(\mathbf{x},t\right)\delta\left(\mathbf{x}-\boldsymbol{\chi}\left(\mathbf{X},t\right)\right)d\mathbf{x}, \tag{4.4}$$

where $\mathbf{X}$ is the current configuration of the Lagrangian structure and $\mathbf{F}$ is the Lagrangian force density. Equation (4.3) spreads the Lagrangian force onto the Eulerian mesh, while equation (4.4) specifies that the structure moves at the local fluid velocity. The form of the Lagrangian force density is determined by the constitutive model of the Lagrangian structure. When we discretize the continuous equations in space, we replace the delta function with a regularized delta function. First,

recall the defining feature of the delta function $\delta(\mathbf{x}) = \delta(x)\,\delta(y)$,

$$\int_V \delta(\mathbf{x} - \mathbf{X})\, d\mathbf{x} = \begin{cases} 1 & \text{if } \mathbf{X} \in V \\ 0 & \text{otherwise.} \end{cases} \tag{4.5}$$

We wish for our regularized delta function $\delta_h(\mathbf{x}) = \frac{1}{h^2}\phi\left(\frac{x}{h}\right)\phi\left(\frac{y}{h}\right)$ to satisfy the same conditions as h approaches 0. Several axioms have been proposed for choosing the definition of ϕ [19, 42]. For our present purposes, we show one possible formulation of ϕ called the IB four-point delta function:

$$\phi_4(r) = \begin{cases} \frac{1}{8}\left(3 - 2|r| + \sqrt{1 + 4|r| - 4r^2}\right) & \text{if } 0 \le |r| < 1 \\ \frac{1}{8}\left(5 - 2|r| - \sqrt{-7 + 12|r| - 4r^2}\right) & \text{if } 1 \le |r| \le 2 \\ 0 & \text{otherwise.} \end{cases} \tag{4.6}$$

The application of a singular force layer given by equations (4.3) and (4.4) induces discontinuities at the interface due to force balance. These discontinuities are regularized by using a smooth kernel function. Smoothing out the stress discontinuities appears to result in non-convergence of stresses at the interface, although stresses can be accurately captured away from the interface.

4.2 The Direct Forcing Method

In the comparisons that follow, we use a modified version of the immersed boundary method that we refer to as the direct forcing method (DF). In this method, instead of interpolating the fluid velocity to the structure, we specify the displacement of the structure. This relaxes the no-slip condition on the structure. To approximate the no-slip condition, we apply a direct force to the momentum equation to penalize the difference between the specified motion of the structure and the computed velocity of the fluid. Mathematically, we replace equation (4.3) with

$$\mathbf{F} = \eta\left(\frac{\partial \boldsymbol{\chi}(\mathbf{X}, t)}{\partial t} - \mathbf{u}\left(\boldsymbol{\chi}(\mathbf{X}, t), t\right)\right) \tag{4.7}$$

and specify the location of the structure by eliminating equation (4.4). While we no longer have to worry about the *location* of the structure with this method, we now allow some amount of leakage through our structure. The magnitude of this leak is controlled by the penalty parameter η.

4.3 The Immersed Interface Method

A limitation of the IB method is that it yields low-order accuracy at fluid-structure interfaces [19]. An alternative approach known as the immersed interface method (IIM) is to incorporate known jump conditions about the fluid flow into the finite difference stencils [43]. A drawback for IIM is that the jump conditions must be known beforehand. We start by deriving the jump conditions for the stress [44, 45]. We remark that for the continuous equations of motion, the IB method and the IIM formulation are equivalent [46].

We define the jump in a function as

$$[\![f]\!](\mathbf{x}) = \lim_{\epsilon \to 0^+} f(\mathbf{x} + \epsilon \mathbf{n}) - \lim_{\epsilon \to 0^-} f(\mathbf{x} + \epsilon \mathbf{n}), \tag{4.8}$$

where $\mathbf{n}$ is the outward unit normal to the interface. The fluid velocity is continuous $[\![\mathbf{u}]\!] = 0$ because of the no-slip condition along the interface. Using this result, we can additionally show that the jump in the material derivative of the velocity is also zero

$$\begin{aligned} \frac{D[\![\mathbf{u}]\!]}{Dt} &= \frac{D\mathbf{u}(\mathbf{X}^+, t)}{Dt} - \frac{D\mathbf{u}(\mathbf{X}^-, t)}{Dt} && (4.9) \\ &= \frac{\partial \mathbf{u}(\mathbf{X}^+, t)}{\partial t} + \nabla \mathbf{u}\left(\mathbf{X}^+, t\right) \frac{\partial \mathbf{X}^+}{\partial t} \\ &\quad - \frac{\partial \mathbf{u}(\mathbf{X}^-, t)}{\partial t} + \nabla \mathbf{u}\left(\mathbf{X}^-, t\right) \frac{\partial \mathbf{X}^-}{\partial t} && (4.10) \\ &= \left[\!\!\left[\frac{D\mathbf{u}(\mathbf{X}, t)}{Dt} \right]\!\!\right] = 0. && (4.11) \end{aligned}$$

To derive the jump condition for the stress, we take an infinitesimal area δS along an interface S. We then extend δS in the normal directions $\mathbf{n}$ and $-\mathbf{n}$ by $\epsilon/2$. We denote this region by δV. We integrate equation (4.1) over the region δV and take the limit as $\epsilon \to 0$:

$$\lim_{\epsilon \to 0} \int_{\delta V} \rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) dV = \lim_{\epsilon \to 0} \int_{\delta V} (\nabla \cdot \boldsymbol{\sigma})\, dV + \lim_{\epsilon \to 0} \int_{\delta V} \mathbf{f} dV. \tag{4.12}$$

We note that the left-hand side of the equation is 0 from Reynolds transport theorem and equation

(4.9). We apply Gauss' theorem to the right-hand side integral to obtain

$$\lim_{\epsilon\to 0}\int_{\delta V}\nabla\cdot\boldsymbol{\sigma}dV = \lim_{\epsilon\to 0}\int_{\delta S}\boldsymbol{\sigma}\cdot\mathbf{n}dS \tag{4.13}$$

$$= [\![\boldsymbol{\sigma}]\!]\cdot\mathbf{n}\delta S. \tag{4.14}$$

The final term in equation (4.12) is

$$\lim_{\epsilon\to 0}\int_{\delta V}\mathbf{f}dV = \lim_{\epsilon\to 0}\int_{S}\int_{\delta V}\mathbf{F}\delta\left(\mathbf{x}-\mathbf{X}\right)dVd\mathbf{X} \tag{4.15}$$

$$= \mathbf{F}\delta s, \tag{4.16}$$

where δs is the infinitesimal area of the region in reference coordinates. Combining the above equations we get

$$[\![\boldsymbol{\sigma}]\!]\cdot\mathbf{n}\delta S + \mathbf{F}\delta s = 0 \tag{4.17}$$

or that

$$[\![\boldsymbol{\sigma}]\!]\cdot\mathbf{n} = -\mathbf{F}J^{-1}, \tag{4.18}$$

where J is the Jacobian of the mapping between the reference and current configurations. We note that this is the jump in the total stress regardless of the constitutive equation for the Cauchy stress tensor $\boldsymbol{\sigma}$. In the case of a Newtonian fluid, we can decompose the jump for the total stress into jumps for the velocity gradient and pressure. First, let $\boldsymbol{\sigma} = -p\mathbb{I} + \frac{\mu}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right)$ which reduces equation (4.18) to

$$-[\![p]\!]\,\mathbf{n} + \mu\,[\![\nabla\mathbf{u}]\!]\,\mathbf{n} = -\mathbf{F}J^{-1}. \tag{4.19}$$

Multiplying equation (4.19) by $\mathbf{n}$ and noting that $\left[\!\left[\frac{\partial\mathbf{u}}{\partial\mathbf{t}}\right]\!\right] = \left[\!\left[\frac{\partial\mathbf{u}}{\partial\mathbf{b}}\right]\!\right] = 0$ where $\mathbf{t}$ and $\mathbf{b}$ are the tangential and binormal unit vectors, we get the jump conditions for the pressure and the normal derivative in the velocity for the Newtonian case:

$$[\![p]\!] = J^{-1}\mathbf{F}\cdot\mathbf{n}, \tag{4.20}$$

$$\mu\left[\!\left[\frac{\partial\mathbf{u}}{\partial\mathbf{n}}\right]\!\right] = \frac{\left(\mathbf{F}\cdot\mathbf{n}\right)\mathbf{n}-\mathbf{F}}{J^{-1}}. \tag{4.21}$$

Although there are higher-order jump conditions for the normal derivative of the pressure as well as the second normal derivative of the velocity, the jump conditions involve the curvature of the structure [47]. Since our discretization, which will be described in more detail in the next section, involves using discrete surface representations, the curvature is not well defined. Therefore, we do not account for these jump conditions in this work. There are also jump conditions in the temporal derivatives of the velocity as the boundary moves. However, it has been demonstrated that these jumps have little effect on the overall accuracy of the method [45], and therefore we do not account for these jumps either.

We stress that while the jumps in the pressure and normal derivative of the velocity are different when changing the constitutive equations for the stress, the jump in the *total stress* is the same. This is beneficial when discretizing the equations. For clarity, we briefly discuss the discretization of the jumps here. Consider a standard second-order central difference to the first derivative of a function $\phi(\mathbf{x})$

$$\frac{\partial \phi}{\partial x}\left(\mathbf{x}_{i+\frac{1}{2},j}\right) = \frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x}, \tag{4.22}$$

where $\mathbf{x} = (x, y)$, $\phi_{i,j} = \phi(x_i, y_j)$, and Δx is the spacing in the x-direction. Using Taylor series expansions, it can be shown that if an interface has a jump at the point $\mathbf{x}_o$ such that $x_{i,j} \leq x_o < x_{i+1,j}$ such that $x_{i,j}$ and $x_{i+1,j}$ lie on different sides of the interface, then the derivative of ϕ is

$$\frac{\partial \phi}{\partial x}\left(\mathbf{x}_{i+\frac{1}{2},j}\right) = \frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x} + \frac{\operatorname{sgn}\{n^x\}}{\Delta x} \sum_{m=0}^{2} \frac{d^+}{m!} \left[\!\!\left[\frac{\partial^m \phi}{\partial x^m} \right]\!\!\right] (\mathbf{x}_o) + \mathcal{O}\left(\Delta x^2\right), \tag{4.23}$$

where $d^+ = x_{i+1,j} - x_o > 0$, n^x is the x-component of the normal vector $\mathbf{n}$ at the intersection point $\mathbf{x}_o$. The importance of equation (4.23) is that we can rearrange it so that the standard finite difference stencil is given in terms of the jump and function value

$$\frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x} = \frac{\partial \phi}{\partial x}\left(\mathbf{x}_{i+\frac{1}{2},j}\right) - \frac{\operatorname{sgn}\{n^x\}}{\Delta x} \sum_{m=0}^{2} \frac{d^+}{m!} \left[\!\!\left[\frac{\partial^m \phi}{\partial x^m} \right]\!\!\right] (\mathbf{x}_o) + \mathcal{O}\left(\Delta x^2\right). \tag{4.24}$$

Therefore, it doesn't matter which stencil in the divergence of the Cauchy stress the jumps are applied to, the components will sum together appropriately.

4.4 Discretization

The numerical discretization for the fluid equations is described in detail in the previous chapter. The details for the fluid-structure interaction component as well as the modified finite difference stencils are described in detail in Kolahdouz *et al*˙ [48]. Here, we provide a short summary of the relevant components. An important implementation is that while we are building the correct *stress jumps* into the conservation of momentum equation, we use standard central differences for the velocity gradient when evolving the extra stress tensor. At the time of this work, the jumps in the normal derivative of the velocity are unknown.

4.4.1 Eulerian Discretization

We use a staggered-grid (MAC) discretization of the fluid equations where we approximate the fluid velocity along the cell edges, while the pressure and components of the conformation tensor are approximated at the cell centers. We use standard second-order finite differences for the gradient, divergence, and the Laplacian. The discrete divergence $\nabla_h \cdot \mathbf{u}$ and gradient of the velocity $\nabla_h \mathbf{u}$ are evaluated at the cell centers. The discrete gradient of the pressure $\nabla_h p$, the discrete Laplacian $\nabla_h^2 \mathbf{u}$ and the discrete divergence of the conformation tensor $\nabla \cdot \mathbb{C}$ are evaluated at cell sides. We use a wave-propagation algorithm with WENO reconstructions to evaluate the nonlinear advection terms $\mathbf{u}\nabla \cdot \mathbf{u}$ and $\mathbf{u}\nabla \cdot \mathbb{C}$ [23, 24]. At regions of high-stress gradients, we use a locally refined Eulerian discretization generated by an adaptive mesh refining algorithm. Near the interface, the finite difference stencils for the conservation of momentum equation are modified to include the jump conditions by using modified Taylor series expansions. However, we currently do not include the jump conditions in the stencils for the evolution of the conformation tensor. In particular, the gradient of the velocity $\nabla_h \mathbf{u}$ used in the nonlinear stretching term uses standard second-order finite differences at all points in the domain.

4.4.2 Lagrangian Discretization

We use a finite element representation of the interface. We discretize the structure Ω_s using a triangulation $T_h = \cup_e K^e$. On T_h, we define Lagrangian basis functions $\{\phi_l(\mathbf{X})\}_{l=1}^m$ where m is the total number of nodes in the triangulation. We approximate the interfacial deformation by

$$\boldsymbol{\chi}_h(\mathbf{X}, t) = \sum_{l=1}^{m} \boldsymbol{\chi}_l(t)\phi_l(\mathbf{X}), \tag{4.25}$$

where $\boldsymbol{\chi}_l(t)$ is the time-dependent physical position of the Lagrangian node indexed by l. Here we use C^0 basis functions which provides a discontinuous approximation of the deformation gradient. In particular, the normal direction of the interface is discontinuous. Because of this, we project the jump conditions onto the space spanned by the Lagrangian basis.

4.4.3 Time Integration

Starting from values of $\boldsymbol{\chi}^n$, $\mathbf{u}^n$, and $\mathbb{C}^n$ at time t^n and $p^{n-\frac{1}{2}}$ at time $t^{n-\frac{1}{2}}$, we must determine values of $\boldsymbol{\chi}^{n+1}$, $\mathbf{u}^{n+1}$, $\mathbb{C}^{n+1}$, and $p^{n+\frac{1}{2}}$. We first determine an approximate structure position at an intermediate time

$$\frac{\hat{\boldsymbol{\chi}}^{n+1} - \boldsymbol{\chi}^n}{\Delta t} = \mathcal{J}^n \mathbf{u}^n, \quad \boldsymbol{\chi}^{n+\frac{1}{2}} = \frac{\hat{\boldsymbol{\chi}}^{n+1} - \boldsymbol{\chi}^n}{2}, \tag{4.26}$$

where $\mathcal{J}^n$ is the velocity Eulerian to Lagrangian interpolation. Using this position approximation, we then solve for $\mathbf{u}^{n+1}$, $\mathbb{C}^{n+1}$, and $p^{n+\frac{1}{2}}$:

$$\rho\left(\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \mathbf{N}_1^{n+\frac{1}{2}}\right) = -\nabla_h p^{n+\frac{1}{2}} + \mu_s \nabla_h^2 \mathbf{u}^{n+\frac{1}{2}} + \frac{\mu_p}{\lambda} \nabla_h \cdot \mathbb{C}^{n+\frac{1}{2}} + \mathbf{f}^{n+\frac{1}{2}} \tag{4.27}$$

$$\nabla_h \cdot \mathbf{u}^{n+1} = 0 \tag{4.28}$$

$$\frac{\mathbb{C}^{n+1} - \mathbb{C}^n}{\Delta t} + \mathbf{N}_2^{n+\frac{1}{2}} - \mathbf{N}_3^{n+\frac{1}{2}} = \mathbf{S}\left(\mathbb{C}^{n+\frac{1}{2}}\right), \tag{4.29}$$

where $\mathbf{N}_1^{n+\frac{1}{2}}$ and $\mathbf{N}_2^{n+\frac{1}{2}}$ are approximations to the nonlinear advection terms, $\mathbf{N}_3^{n+\frac{1}{2}}$ is the approximation to the nonlinear stretching term, and $\mathbf{f}^{n+\frac{1}{2}}$ is the correction terms involving the jump conditions at time $t^{n+\frac{1}{2}}$ at position $\boldsymbol{\chi}^{n+\frac{1}{2}}$. Instead of solving the fully coupled problem, we cycle between two separate equations. The first is the conservation of mass and momentum equation

$$\rho\left(\frac{\mathbf{u}^{n+1,k+1} - \mathbf{u}^n}{\Delta t} + \mathbf{N}_1^{n+\frac{1}{2},k}\right) = -\nabla_h p^{n+\frac{1}{2},k+1} + \mu_s \nabla_h^2 \mathbf{u}^{n+\frac{1}{2},k+1} + \frac{\mu_p}{\lambda} \nabla_h \cdot \mathbb{C}^{n+\frac{1}{2},j} + \mathbf{f}^{n+\frac{1}{2}} \tag{4.30}$$

$$\nabla_h \cdot \mathbf{u}^{n+1,k+1} = 0, \tag{4.31}$$

where $\mathbb{C}^{n+\frac{1}{2},j}$ is the most recent approximation to $\mathbb{C}^{n+\frac{1}{2}}$. The second problem is

$$\frac{\mathbb{C}^{n+1,j+1} - \mathbb{C}^n}{\Delta t} + \mathbf{N}_2^{n+\frac{1}{2},j} - \mathbf{N}_3^{n+\frac{1}{2},j} = \mathbf{S}\left(\mathbb{C}^{n+\frac{1}{2},j}\right), \tag{4.32}$$

where $\mathbf{u}^{n+\frac{1}{2},k}$ is the most recent approximation to $\mathbf{u}^{n+\frac{1}{2}}$. We use a fixed number of fixed-point iterations n_{ns} for the Navier-Stokes equations and n_{av} for the stress evolution equation. At the end of the iterations we set $\mathbf{u}^{n+1} = \mathbf{u}^{n+1,n_{ns}}$, $p^{n+\frac{1}{2}} = p^{n+\frac{1}{2},n_{ns}}$, and $\mathbb{C}^{n+1} = \mathbb{C}^{n+1,n_{av}}$. Finally, using the new velocity, we solve for $\boldsymbol{\chi}^{n+1}$:

$$\frac{\boldsymbol{\chi}^{n+1} - \boldsymbol{\chi}^{n}}{\Delta t} = \mathcal{J}^{n+\frac{1}{2}}\left(\frac{\mathbf{u}^{n+1} + \mathbf{u}^{n}}{2}\right). \tag{4.33}$$

The resulting linear systems of equations for the velocity and pressure are solved using the flexible GMRES algorithm with the projection method as a preconditioner [22].

4.4.4 Implementation

The solver is built into IBAMR [19], an open-source adaptive and distributed memory parallel implementation of the immersed boundary method. IBAMR is built on top of SAMRAI [26] which provides infrastructure for structured adaptive mesh refinement. The resulting linear systems are solved using implementations provided by PETSc [27]. The finite element discretization is provided by libMesh [49].

4.5 Results

4.5.1 Flow Past a Cylinder

A common benchmark problem for complex fluid solvers is flow past a confined cylinder. Despite possessing a smooth geometry, flow past a confined cylinder remains a challenging problem for viscoelastic fluids. Thin stress layers along the edge of the cylinder and in the immediate wake of the cylinder give strong spatial restrictions to accurately resolve boundary features. There has been recent success in using complicated finite element [2, 50, 51] and finite volume [52, 53] methods with body-conforming grids to resolve simulations at larger Weissenberg numbers. However, for Weissenberg numbers above $\mathrm{Wi} \geq 0.7$, converged stress results remain an unsolved problem. Here we test whether a much simpler fluid solver can accurately resolve the features of flow past a confined cylinder with the help of adaptive mesh refinement.

We use an immersed interface representation of the fluid-structure interaction. A penalty force is applied to the surface of the cylinder,

$$\mathbf{F}(\mathbf{X}, t) = \kappa\left(\mathbf{X} - \boldsymbol{\chi}\left(\mathbf{X}, t\right)\right) + \eta\left(\mathbf{U} - \mathbf{u}\right), \tag{4.34}$$

where κ is the spring stiffness penalty parameter and η is the damping penalty parameter. As $\kappa \to \infty$, the formulation approaches that of a rigid body. Numerical experiments have shown that the damping parameter allows for a more stable simulation that can be run at much higher advective CFL numbers. For convergence studies in which the displacement of the cylinder $\mathbf{D} = \mathbf{X} - \mathbf{x} = \mathcal{O}\left(h^2\right)$, because the force $\mathbf{F} = \mathcal{O}\left(1\right)$ under grid refinement, we require that κ satisfy $\kappa = \mathcal{O}\left(\frac{1}{h^2}\right)$. In a similar argument, in order for the velocity $\mathbf{V} = \mathbf{U} - \mathbf{u} = \mathcal{O}\left(h\right)$, we require that $\eta = \mathcal{O}\left(\frac{1}{h}\right)$. We choose the form $\kappa = \kappa_0 \frac{1}{h^2}$ and $\eta = \eta_0 \frac{1}{h}$, and then choose κ_0 and η_0 so that the largest possible stable value is used on the finest grid considered in the convergence study for a given time step.

In order to compare against literature results, we set the viscosity ratio as $\frac{\mu_N}{\mu_N + \mu_p} = 0.5$ and set the initial inflow and outflow boundary conditions to be that of fully developed pipe flow. No-slip boundary conditions for the velocity with homogeneous Neumann boundaries for the conformation tensor are used for the top and bottom of the domains. The radius of the cylinder is $R = 1$, and the width of the channel is $L_y = 4R$. The length of the computational domain is $L_x = 20R$. This ensures that our assumption of a fully developed pipe flow is valid. We run at an advective CFL number of ≈ 0.001 until a final time of $T = 40$. While the fluid solver is stable up to a convective CFL number of 0.3, a significantly smaller timestep is required in order to ensure the cylinder moves no more than a fraction of a grid cell. The computational grid is discretized so that the smallest level uses $N_y = 8$ grid points in the y-direction and $N_x = N_y \frac{L_x}{L_y}$. The computational grid is shown in Figure 4.2. To compare against literature results, we measure the drag coefficient along the surface of the cylinder. Most of the literature results are run at the zero Reynolds number; however, our adaptive discretization breaks down as $\mathrm{Re} \to 0$, therefore we run at a small but non-zero Reynolds number of $\mathrm{Re} \approx 10^{-5}$. We therefore do not expect to get the same drag coefficients or results as those in the literature. The drag coefficient can be found by integrating the normal stress along the surface of the cylinder:

$$C_d = \frac{1}{\eta} \int_S \boldsymbol{\sigma} \cdot \mathbf{n} \cdot \mathbf{e}_x da, \tag{4.35}$$

where $\mathbf{n}$ is the outward unit normal and $\mathbf{e}_x$ is the unit vector in the x-direction. Since the structure is stationary, the penalty force is balanced with the stress along the cylinder. Therefore, we can

instead integrate the penalty force

$$C_d = \frac{1}{\eta} \int_S -\mathbf{F} \cdot \mathbf{e}_x da. \tag{4.36}$$

A comparison of drag coefficients against literature values is presented in Table 4.1. For moderate Weissenberg numbers $0.4 \leq \text{Wi} \leq 0.7$, we achieve differences around one percent when compared against literature values. However, for small Weissenberg numbers $\text{Wi} \leq 0.3$, we achieve differences that are much smaller than one percent. Note that since other methods do not see convergence for high Weissenberg numbers, we do not attempt to run at these values.

Figure 4.3 shows the plot of the top right component of the conformation tensor $\mathbb{C}_{xx}$. Note the formation of an extra recirculation region for the component for larger Weissenberg numbers as found in other studies [2]. It has been suggested that to achieve converged results, the recirculation region must be sufficiently resolved [2]. We believe the tools described in this paper will eventually allow us to have converged results in this recirculation region.

To assess the convergence rates of flow past a cylinder, we use Richardson extrapolation and focus on the $\text{Re} = 1.0$ and $\text{Wi} = 0.3$ case. Given solutions on the three grids $\mathbf{u}_h$, $\mathbf{u}_{2h}$ and $\mathbf{u}_{4h}$, the convergence rate can be found by $q = \log_2 \frac{|\mathbf{u}_{4h} - \mathbf{u}_{2h}|}{|\mathbf{u}_{2h} - \mathbf{u}h|}$. We run on a grid with three levels with a refinement ratio of two. Initially, the coarsest level uses 8 grid cells in the y-direction. The finer grids use 16 and 32 grid cells on the coarsest level in the y-direction respectively. We then run to a final time of $T = 1$. To compute the difference between the two solutions, we interpolate the fine grid onto the coarser grid. Table 4.2 shows the convergence rates. We do not see convergence in the max norm for the components of the conformation tensor. However, if we remove the area within five grid cells of the interface from the computations of the norms, we restore first-order accuracy in the max norm.

A comparison between the drag coefficients computed from the immersed interface (IIM), direct forcing (DF), and immersed boundary (IB) methods is given in Figure 4.3. In these comparisons, we use the same time step and grid and select the penalty parameters to be as large as possible and still have a stable simulation. In order to achieve a stable simulation with the IB method, we must use a timestep that is four times smaller than the other methods. We achieve slightly better drag coefficients in the low Weissenberg number cases for the IIM method, however as we increase the

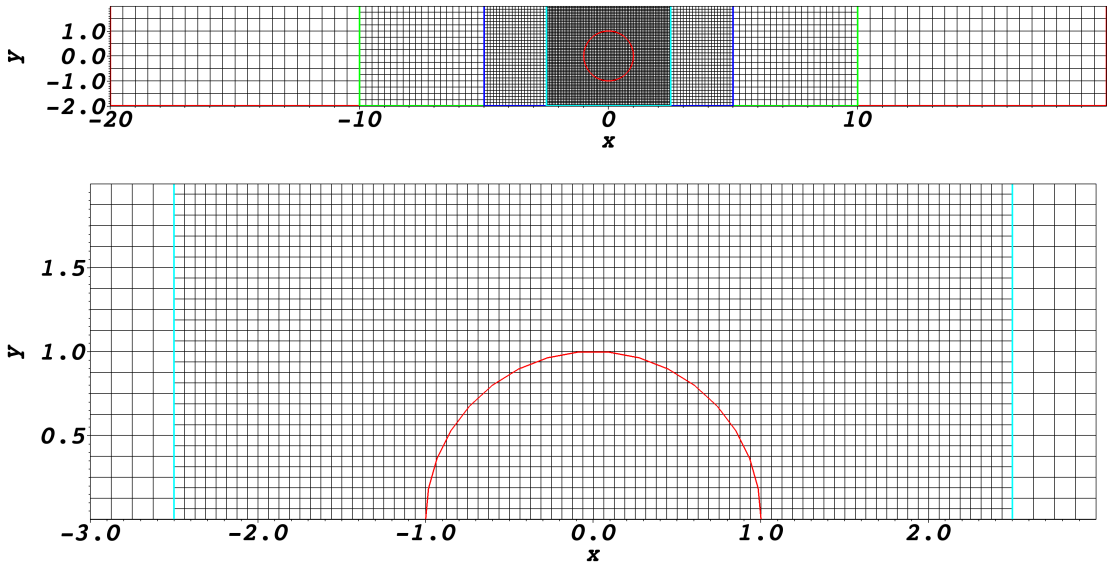

Figure 4.2: The computational setup for flow past a cylinder. Top: The full computational grid. Bottom: A close up near the surface of the cylinder.

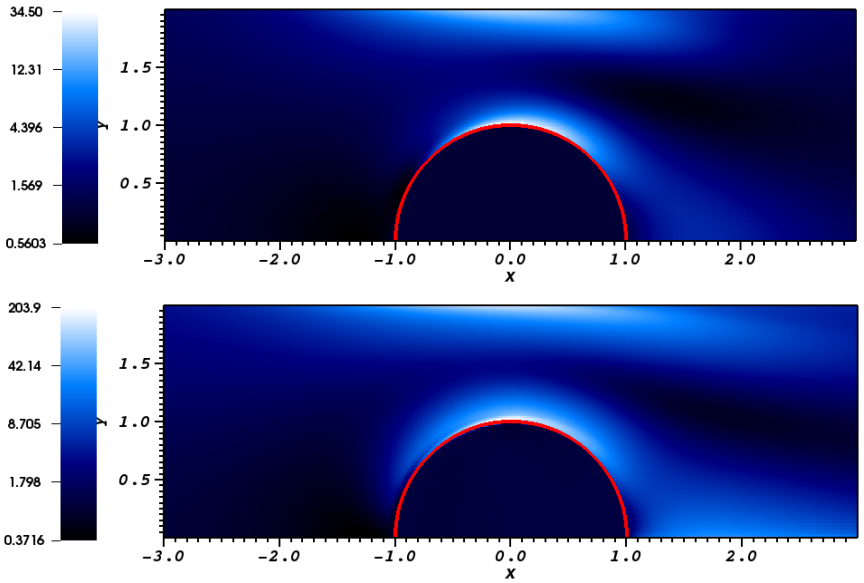

Figure 4.3: The top right component of the conformation tensor $\mathbb{C}_{xx}$ for Weissenberg numbers $\mathrm{Wi} = 0.3$ (top) and $\mathrm{Wi} = 0.7$ (bottom) for a Reynolds number of $\mathrm{Re} = 1.0$.

	Re = 0.0			Re = 0.01		
Wi	C_d	Claus *et al*˙ [2]	Percent Difference	C_d	Claus *et al*˙ [2]	Percent Difference
0.1	130.152	130.364	0.163	130.154	130.364	0.161
0.2	126.704	126.626	0.0614	126.707	126.627	0.0632
0.3	123.919	123.192	0.590	123.922	123.194	0.591
0.4	122.107	120.593	1.26	122.110	120.595	1.26
0.5	121.208	118.826	2.00	121.218	118.831	2.01
0.6	121.014	117.776	2.75	121.020	117.781	2.75
0.7	121.414	117.316	3.49	121.421	117.323	3.49
	Re = 0.1			Re = 1.0		
Wi	C_d	Claus *et al*˙ [2]	Percent Difference	C_d	Claus *et al*˙ [2]	Percent Difference
0.1	130.172	130.368	0.150	130.558	130.609	0.039
0.2	126.734	126.636	0.0774	127.187	126.938	0.196
0.3	123.951	123.211	0.601	124.457	123.597	0.696
0.4	122.143	120.622	1.26	122.719	121.106	1.33
0.5	121.254	118.868	2.01	121.918	119.460	2.06
0.6	121.075	117.831	2.75	121.874	118.542	2.81
0.7	121.489	117.387	3.50	122.436	118.233	3.55

Table 4.1: Drag coefficients and percent differences between Clause *et al*˙ [2] and our simulations at varying Reynolds numbers. Because our AMR solvers are currently limited to treating Re > 0, we approximate the zero Reynolds number case by Re $= 10^{-5}$.

	$\mathbb{C}_{xx}$		$\mathbb{C}_{yy}$	
	Full domain	Excluding interface	Full domain	Excluding interface
L_1 Norm	1.092684643	1.161744269	1.327913467	1.699105396
L_2 Norm	0.5174536385	1.032426126	0.5309904965	1.464332698
L_∞ Norm	0.07403476449	0.9422156601	-0.5494829672	0.9136947393
	$\mathbb{C}_{xy}$		$\mathbf{U}$	
	Full domain	Excluding interface	Full domain	Excluding interface
L_1 Norm	1.284044673	1.450108172	2.009731988	5.390432122
L_2 Norm	0.4589699265	1.445853863	1.964305873	1.929699451
L_∞ Norm	-0.8085236839	0.7283321233	1.263609924	0.8164007733

Table 4.2: Convergence rates for flow past a cylinder computed using Richardson extrapolation. Convergence rates are given for both the entire domain and a domain that excludes the interface. Note that while the velocity is first order in the max norm in either case, we do not see convergence for the components for the conformation tensor unless we exclude the interface.

	Re = 1.0			
Wi	C_d - II	C_d - DF	C_d - IB	Claus *et al˙* [2]
0.1	130.558	131.143	132.681	130.364
0.2	127.187	127.895	129.384	126.626
0.3	124.457	125.037	126.746	123.192
0.4	122.719	122.998	125.248	120.593
0.5	121.918	121.735	124.899	118.826
0.6	121.874	121.080	124.788	117.787
0.7	122.436	120.904	125.532	117.291

Table 4.3: Drag coefficients and relative differences between Clause *et al˙* [2], the immersed interface method (IIM), the direct forcing IB method (DF), and the conventional immersed boundary method (IB) at Reynolds number Re = 1.0.

Weissenberg number, the DF method gives more accurate drag coefficients. For all Weissenberg numbers tested, the IB method gives consistently less accurate results. Figure 4.4 shows the normal component of the velocity along the surface of the cylinder at the final time for the IIM and DF methods. We get significantly more flow through the cylinder using the DF method as it does a relatively poor job of enforcing the no-slip condition. In particular, the only place where the fluid velocity is not negligible for the IIM method is near the top and bottom of the cylinder, where the shearing creates large stress gradients in the polymeric stress. The drag coefficients computed in the DF method, therefore, do not account for the excess fluid velocity. In these cases, the IIM method is able to enforce the no-slip constraint better than the direct forcing method. Figure 4.5 shows the top right component of the conformation tensor for the DF method for the two Weissenberg numbers 0.3 and 0.7. We see comparable values and locations of stress for the low Weissenberg number of 0.3. However, for the largest Weissenberg number tested, we see a significant decrease in the maximum values of the conformation tensor as well as additional stress components forming inside the cylinder.

4.5.2 Four-Roll Mill

The four-roll mill was a physical device invented by Taylor [54] to asses the physical properties and formation of emulsions. These devices can be used to generate various types of flows from pure rotational to pure extensional by varying the rotational speed of the rollers and are now utilized to assess rheological properties fluids. Recently, Thomases and Shelley [7] demonstrated the singularities that arise with Maxwell type models by using an idealized four-roll mill geometry by applying a background force to generate extensional points in the computational domain. The

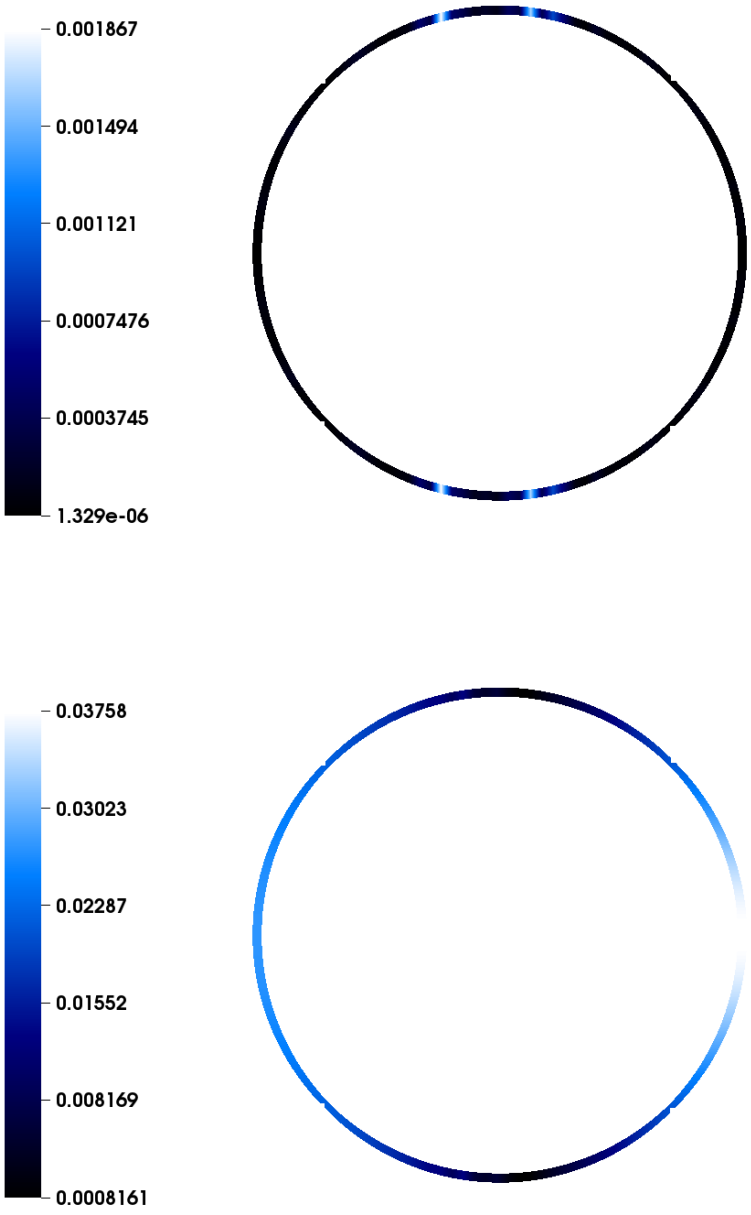

Figure 4.4: The normal component of the velocity along the cylinder for the immersed interface (top) and direct forcing (bottom) methods for a Reynolds number of Re = 1.0 and Weissenberg number of Wi = 0.7.

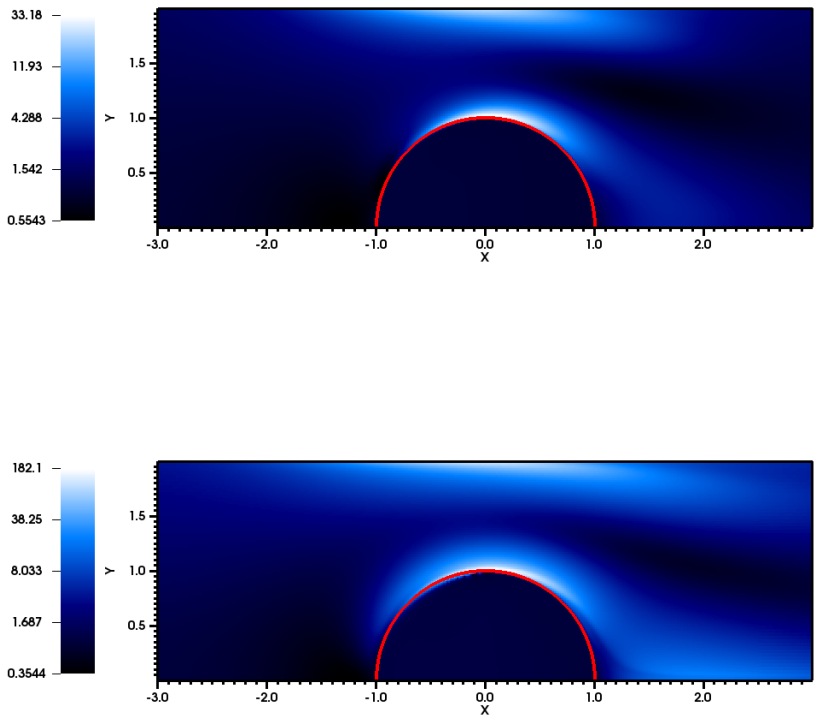

Figure 4.5: The top right component of the conformation tensor $\mathbb{C}_{xx}$ for Weissenberg numbers $\mathrm{Wi} = 0.3$ (top) and $\mathrm{Wi} = 0.7$ (bottom) for the direct forcing method for a Reynolds number of $\mathrm{Re} = 1.0$.

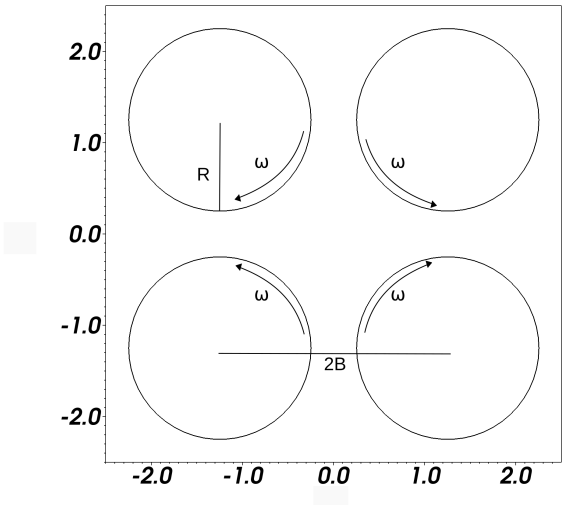

Figure 4.6: The computational domain for the four-roll mill geometry.

idealized geometry consists of a background force density acting directly on the fluid to produce counter-rotating vortices in a Newtonian fluid. When applying this background force to complex fluids, they showed two bifurcations where the trace of the conformation tensor goes from smooth to forming a cusp and finally to growing exponentially in time. This section demonstrates that the same qualitative features arise in a realistic geometry.

We model the four-roll mill geometry by immersing four disks of radius $R = 1$ in a computational domain $\Omega = [-2.5, 2.5] \times [-2.5, 2.5]$. Each disk is centered a distance $B = 1.25$ in each coordinate away from the center of the domain giving a blockage size of $R - B = 0.5$ between each disk [55]. To match results to Thomases and Shelley [7], we use bi-periodic boundary conditions. This gives a computational domain that simulates an infinite array of disks each of distance 0.5 away from each other. At time $t = 0$, we specify the rotational rate of each disk to be $\omega = 1$. We specify the rotational direction of the disks in quadrant one and three to be the opposite of the rotational direction of the disks in quadrant two and four. This produces an extensional point in the center of the domain. We define the Weissenberg number to be $\mathrm{Wi} = \frac{\omega}{\lambda}$, where λ is the relaxation time of the fluid. Following Thomases and Shelley, we specify the fluid viscosity ratio $\beta = \frac{\mu_p}{\mu_n} = 0.5$. We fix the Reynolds number at $\mathrm{Re} = 1.0$. Simulations are run to a final time of $T = 30\mathrm{Wi}$ so that a steady-state solution is obtained. Figure 4.6 shows the computational grid.

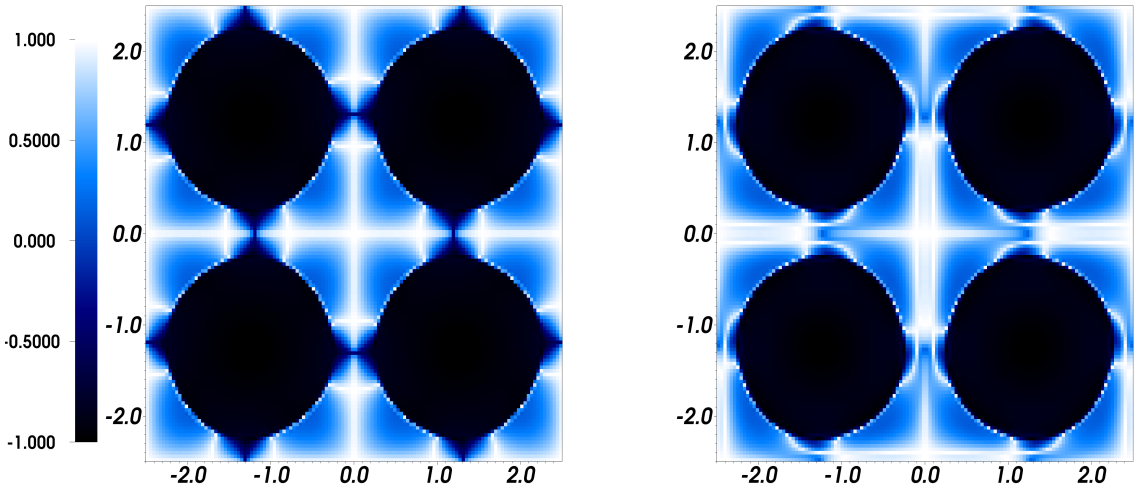

Figure 4.7: A plot of the flow-type parameter ξ for Weissenberg number Wi = 0.1 (left) and Wi = 0.9 (right).

As done by Lee *et al*˙ [56], the type of fluid motion can be described by the flow-type parameter $\xi = \frac{|\mathbf{E}|-|\Omega|}{|\mathbf{E}|+|\Omega|}$, where $|\mathbf{E}|$ and $|\Omega|$ are the magnitudes of the deformation and vorticity tensors respectively. Using this parameter, the flow can be described as either purely rotational $\xi = -1$, purely shear $\xi = 0$, or purely extensional $\xi = 1$. Figure 4.7 shows the flow type parameter for Weissenberg numbers Wi = 0.1 and Wi = 0.9. As we increase the Weissenberg number, the flow becomes less rotational and more extensional.

Figure 4.3 shows the trace of the conformation tensor for varying Weissenberg numbers. As the Weissenberg number increases, we see the formation of singular structures near extensional points in the domain. We note that the location of the singular features differs from the described by Thomases and Shelley [7]. We suspect that this is because of the distance between the disks. The idealized geometry effectively removed the distance between the cylinders, thus removing an extensional point. Figure 4.9 shows a slice of the trace of the conformation tensor along the line $y = 0$ through the center of the domain. We see similar features with a bifurcation happening around a critical Weissenberg number of Wi ≈ 0.5, where a cusp begins to form at the center of the domain.

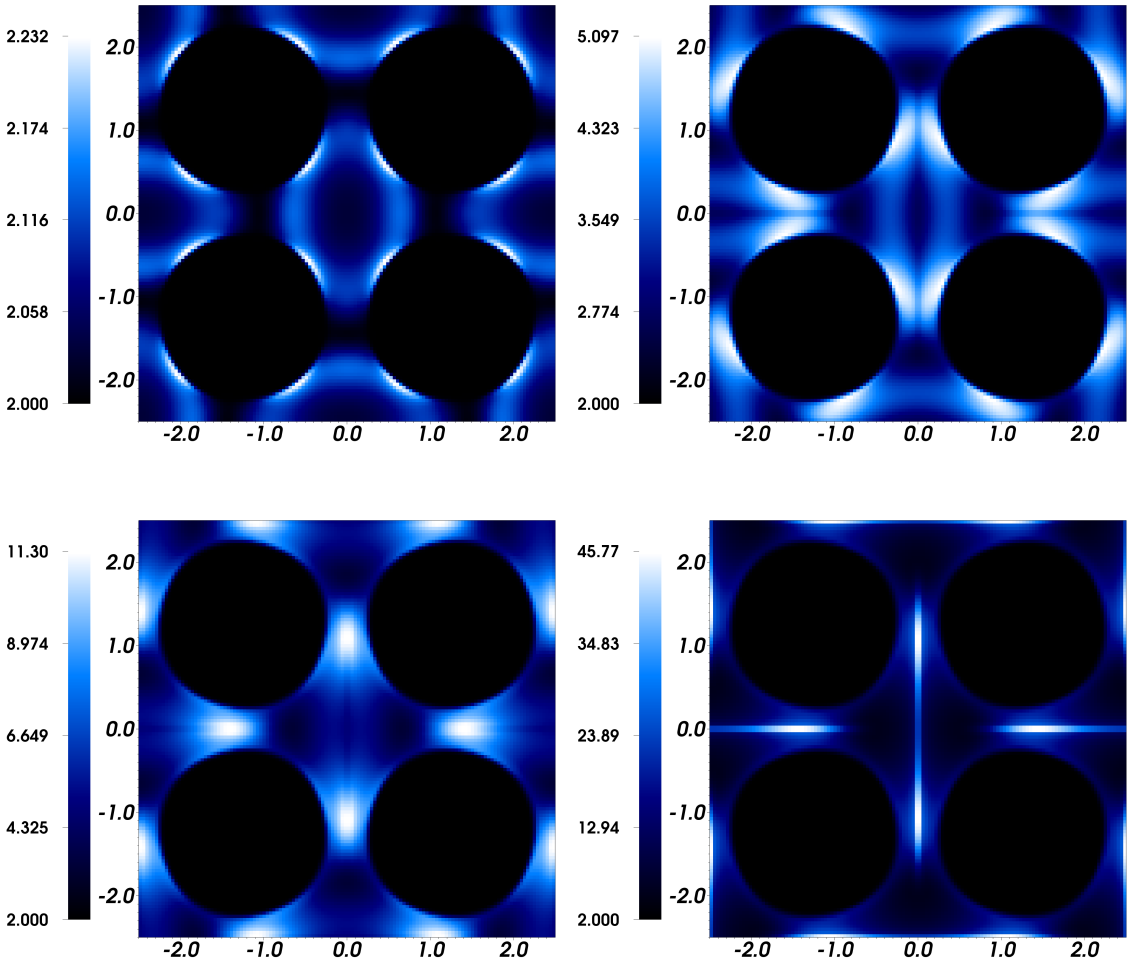

Figure 4.8: The trace of the conformation tensor for Weissenberg numers Wi = 0.1 (top left), Wi = 0.4 (top right), Wi = 0.6 (bottom left), and Wi = 0.9 (bottom right).

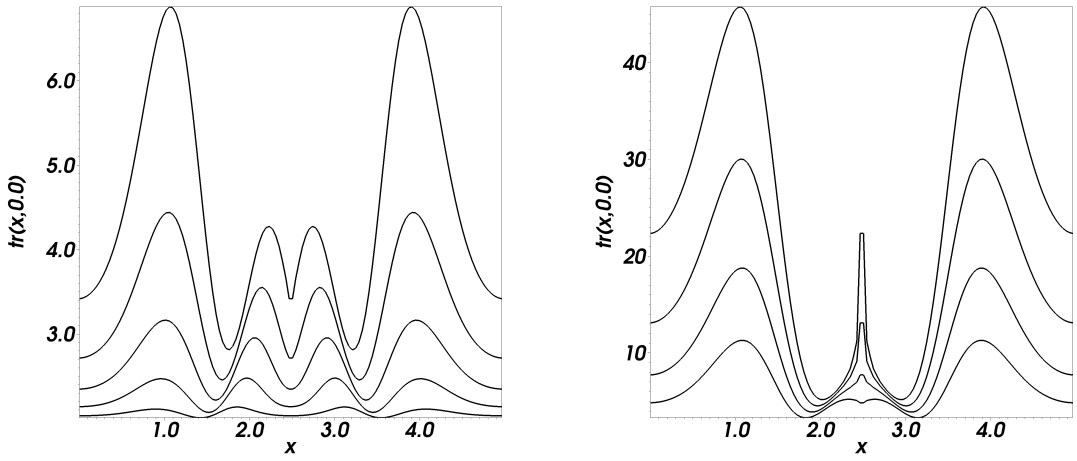

Figure 4.9: A slice of the trace of the conformation tensor for Weissenberg numbers $\mathrm{Wi} = 0.1 - 0.5$ (left), $\mathrm{Wi} = 0.6 - 0.9$ (right) through the line $y = 0$.

CHAPTER 5

APPLICATIONS WITH BACTERIAL FLAGELLA

Single-celled organisms use a variety of techniques to swim [57, 58]. Organisms like *Ecoli* use multiple rotating flagellar filaments that are connected to a rotary motor by a flexible short hook. These organisms switch between two different phases of motion, a *run* and *tumble* phase [59]. In the run phase, all the flagella motors rotate counterclockwise. The hydrodynamic interactions cause the flagella to bundle together and form a superflagellum that propels the organism forward, see Figure 5.1. In the tumble phase, one or more motors switch directions, causing those flagella to change the chirality of the helical shape. This transformation causes the bundle to deform, which causes the cell to rotate and tumble. Eventually, all the motors resume the original counter-clockwise motion, which forms the superflagellum and allows the organism to resume its forward swimming motion.

The environment plays an important role in the motility of organisms [60]. Organisms frequently navigate a complex environment composed of a viscous Newtonian solvent with suspended microstructures such as proteins and filament networks. While the physics of motion is well understood in Newtonian fluids [30], a complete understanding of the motion of microorganisms in a complex fluid is still lacking. In a single-phase viscoelastic fluid, Lauga [61] showed that an infinite undulating sheet is always hindered by viscoelasticity. However, Teran *et al.* [62] showed that a finite sheet can achieve a speed boost in velocity when its amplitude is tuned with the relaxation time of the fluid. Thomases and Guy [63] showed that both body shape and body elasticity must be tuned together to achieve the speed up for undulating sheets. Fu *et al.* [64] showed that small pitch angle helical swimmers are hindered by elasticity. For sufficiently large angles, the swimming speed can increase with a peek found near a Deborah number of $\mathrm{De} \approx 1.0$ [65, 66]. Wrobel *et al.* [11] used a discrete model of viscoelasticity to show that swimmers can get an increase in speed, but they need to use additional power to maintain their gait.

Elastic bacterial flagella can be modelled using Kirchhoff rod theory because the flagellum is long ($\approx 10\,\mu\mathrm{m}$) but very thin ($\approx 10\,\mathrm{nm}$). Kirchhoff rod theory describes the forces and torques generated

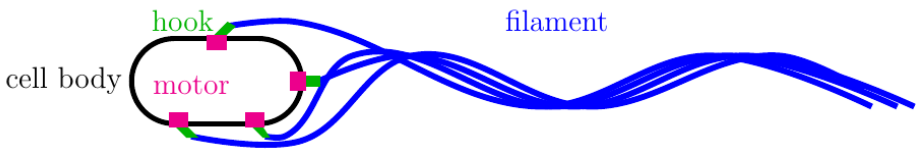

Figure 5.1: A cell body with four flagella. When each flagella's motor spins counter-clockwise, the flagella combine to form a superflagellum which propels the cell forward. This image is reproduced from Ko *et al.* [1].

by an elastic rod in terms of the position of the centerline and an orthonormal set of director vectors attached to the center line. Goldstein *et al.* [67] introduced a bistable energy formulation for Kirchoff rod theory to permit two stable helical configurations. Darnton and Berg [68] used Kirchoff rod theory to estimate the bending rigidity of the flagellum by fitting a Kirchoff rod model to experimental data. Lim and Peskin [69] incorporated hydrodynamic interactions by creating a generalized immersed boundary method based on Kirchoff rod theory. Ko *et al.* [1] incorporated a bistable energy formulation into the generalized immersed boundary method to model polymorphic transformation.

In this Chapter, we use the bistable energy formulation utilized by Ko *et al.* [1] and explore the effects of viscoelasticity on the shape as well as the pumping efficiency of the flagellum.

5.1 Mathematical Model

We consider the motion of a single helical flagellum immersed in an incompressible Maxwell-type fluid. The flagellar element is described by Kirchoff rod theory, in which the flagella is modelled as a thing rod and stresses are applied on the cross-section along the rod. The configuration of the flagella is given by the centerline $\mathbf{X}(s,t)$ and an orthonormal set of director vectors $\{\mathbf{D}^1(s,t), \mathbf{D}^2(s,t), \mathbf{D}^3(s,t)\}$, where s is the arclength along the rod in its reference configuration and t is the current time. To describe the force and torque balance along the filament, let $\mathbf{F}^{\text{rod}}(s,t)$ and $\mathbf{N}^{\text{rod}}(s,t)$ be the force and moment respectively that are generated across a cross-section of the rod at point s along the centerline. Let $\mathbf{F}(s,t)$ and $\mathbf{N}(s,t)$ be the applied force and torque densities.

Then the momentum and angular momentum balance equations are

$$0 = \mathbf{F} + \frac{\partial \mathbf{F}^{\text{rod}}}{\partial s}, \tag{5.1}$$

$$0 = \mathbf{N} + \frac{\partial \mathbf{N}^{\text{rod}}}{\partial s} + \left(\frac{\partial \mathbf{X}}{\partial s} \times \mathbf{F}^{\text{rod}} \right). \tag{5.2}$$

Here, we expand $\mathbf{F}^{\text{rod}}$ and $\mathbf{N}^{\text{rod}}$ in the basis of the local director vectors

$$\mathbf{F}^{\text{rod}} = \sum_{i=1}^{3} F_i \mathbf{D}^i, \quad \mathbf{N}^{\text{rod}} = \sum_{i=1}^{3} N_i \mathbf{D}^i. \tag{5.3}$$

The constitutive relations are given as

$$F_i = b_i \left(\frac{\partial \mathbf{X}}{\partial s} \cdot \mathbf{D}^i - \delta_{3i} \right), \quad i = 1, 2, 3 \tag{5.4}$$

$$N_1 = a_1 \left(\Omega_1 - \kappa_1 \right), \tag{5.5}$$

$$N_2 = a_2 \left(\Omega_2 - \kappa_2 \right), \tag{5.6}$$

$$N_3 = a_3 \left(\Omega_3 - \tau_1 \right) \left(\Omega_3 - \tau_2 \right) \left(\Omega_3 - \frac{\tau_1 + \tau_2}{2} \right) - \gamma^2 \frac{\partial^2 \Omega_3}{\partial s^2}, \tag{5.7}$$

where δ_{3i} is the Kronecker delta function and γ is the twist-gradient coefficient. The coefficients a_1 and a_2 are the bending moduli and a_3 is the twist modulus. In standard Kirchoff rod theory, D_1, D_2, and D_3 are constrained to be a set of local orthonormal director vectors. In the present work, equation (5.4) is a penalty force to maintain the director vectors. The coefficients b_1 and b_2 are the shear moduli and b_3 is the stretching modulus. The strain twist vector $(\Omega_1, \Omega_2, \Omega_3)$, where $\Omega_i = \frac{\partial \mathbf{D}^j}{\partial s} \cdot \mathbf{D}^k$ and (i, j, k) is a cyclic permutation of $(1, 2, 3)$, determines the properties of the helical flagellum. The parameter $\kappa = \sqrt{\kappa_1^2 + \kappa_2^2}$ is the intrinsic curvature and τ_1 and τ_2 are the intrinsic twists. The intrinsic twist strain vector $(\kappa_1, \kappa_2, \tau)$ determines the equilibrium shape of the helical flagellum characterized by the radius R and pitch P

$$R = \frac{\kappa}{\kappa^2 + \tau^2}, \tag{5.8}$$

$$P = \frac{2\pi\tau}{\kappa^2 + \tau^2}, \tag{5.9}$$

Parameter	Symbol	Value
Helical Radius	R_0	$0.2067\,\mu m$
Length	L	$6\,\mu m$
Shear modulus	b_1, b_2	$0.8\,g\,\mu m/s^2$
Stretch modulus	b_3	$0.8\,g\,\mu m/s^2$
Bending modulus	a_1, a_2	$3.5 \times 10^{-3}\,g\,\mu m^3/s^2$
Twist modulus	a_3	$1.0 \times 10^{-4}\,g\,\mu m^5/s^2$
Twist-gradient coefficient	γ	$1.0 \times 10^{-3}\,g^{1/2}\,\mu m^{3/2}/s$
Intrinisic curvature	κ	$1.3057\,\mu m^{-1}$
Right-hand intrinsic twist	τ_1	$-2.1475\,\mu m^{-1}$
Left-hand intrinsic twist	τ_2	$2.1475\,\mu m^{-1}$
Newtonian viscosity	μ_n	$1.0 \times 10^{-6}\,g/\mu m\,s$
Polymeric viscosity	μ_p	$1.0 \times 10^{-6}\,g/\mu m\,s$
Fluid density	ρ	$1.0 \times 10^{-12}\,g/\mu m^3$
Nonlinearity parameter	α	0.3
Time step size	Δt	$2.5 \times 10^{-7}\,s$
Filament grid size	Δs	$0.06\,\mu m$
Fluid grid size	Δx	$0.078\,125\,\mu m$

Table 5.1: Table of physical and computational parameters for both the model flagellum and the fluid.

where τ is either τ_1 or τ_2. In general, the values of κ_1, κ_2, τ_1, and τ_2 need not be constant along the flagellum. We note that the constitutive relations are derived from a variational argument of the elastic energy potential. In particular, equation (5.7) comes from a bistable energy formulation allowing for two different stable handed helices.

The flagellum is immersed in a viscoelastic fluid described by the Giesekus equations where the Cauchy stress tensor consists of a Newtonian solvent $\boldsymbol{\tau}_\mathrm{n}$ and embedded polymers $\boldsymbol{\tau}_\mathrm{p}$:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = \nabla\cdot\boldsymbol{\sigma} + \mathbf{g}, \tag{5.10}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{5.11}$$

$$\boldsymbol{\sigma} = \boldsymbol{\tau}_\mathrm{n} + \boldsymbol{\tau}_\mathrm{p}, \tag{5.12}$$

$$\boldsymbol{\tau}_\mathrm{n} = -p\mathbb{I} + \mu_\mathrm{n} 2\mathbf{D}, \tag{5.13}$$

$$\overset{\triangledown}{\boldsymbol{\tau}}_\mathrm{p} = \frac{\mu_\mathrm{p}}{\lambda} 2\mathbf{D} - \frac{1}{\lambda}\boldsymbol{\tau}_\mathrm{p} - \frac{\alpha}{\lambda}\boldsymbol{\tau}_\mathrm{p}^2, \tag{5.14}$$

where $\mathbf{D} = \frac{1}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right)$ is the rate of strain tensor, $\mathbf{u}$ is the Eulerian velocity, p is the isotropic pressure, $\mathbf{g}$ is the external body force density acting on the fluid, and $\overset{\triangledown}{\boldsymbol{\tau}}$ is the upper convected

derivative defined by

$$\overset{\triangledown}{\boldsymbol{\tau}} = \frac{\partial \boldsymbol{\tau}}{\partial t} + \mathbf{u} \cdot \nabla \boldsymbol{\tau} - \left(\boldsymbol{\tau} \nabla \mathbf{u}^T + \nabla \mathbf{u} \boldsymbol{\tau} \right). \tag{5.15}$$

Instead of evolving the polymeric stress tensor, we evolve the conformation tensor $\mathbb{C}$, which represents an ensemble average of the spatial arrangements of the immersed polymers. The conformation tensor is linearly related to the stress tensor via $\boldsymbol{\tau}_p = G(\mathbb{C} - \mathbb{I})$, where $G = \frac{\mu_p}{\lambda}$ is the elastic modulus of the polymers. Then the evolution of the conformation tensor is

$$\overset{\triangledown}{\mathbb{C}} = \frac{1}{\lambda}(\mathbb{C} - \mathbb{I}) + \frac{1}{\lambda}(\mathbb{C} - \mathbb{I})^2. \tag{5.16}$$

The force density $\mathbf{g}$ is generated by the deformation of the rotating elastic flagellum, given by

$$\mathbf{g} = \int_0^L -\mathbf{F}(s,t)\, \delta_w(\mathbf{x} - \mathbf{X}(s,t))\, ds + \frac{1}{2} \nabla \times \int_0^L -\mathbf{N}(s,t)\, \delta_w(\mathbf{x} - \mathbf{X}(s,t))\, ds, \tag{5.17}$$

where $\delta_w(\mathbf{x}) = \delta_w(x_1)\delta_w(x_2)\delta_w(x_3)$ is a smooth compactly supported kernel function that mediates coupling between the Lagrangian and Eulerian variables. The linear and angular velocities at the filament are computed using by interpolating the Eulerian velocity onto the filament

$$\frac{\partial \mathbf{X}}{\partial t} = \int_\Omega \mathbf{u}(\mathbf{x},t)\, \delta_w(\mathbf{X} - \mathbf{x})\, d\mathbf{x}, \tag{5.18}$$

$$\frac{\partial \mathbf{D}^i}{\partial t} = \frac{1}{2} \int_\Omega (\nabla \times \mathbf{u}(\mathbf{x},t))\, \delta_w(\mathbf{X} - \mathbf{x})\, d\mathbf{x}, \quad \text{for } i = 1,2,3. \tag{5.19}$$

We note that the presence of the smooth kernel function δ_w appears in both the continuum equations, as well as the discretized equations. This is in contrast to traditional IB methods, where δ_w only appears in the discretized equation. In traditional IB methods, w represents a numerical parameter proportional to the Eulerian mesh width. In this context, the parameter w is a physical parameter of the model which can be viewed as controlling the effective thickness of the rod. Larger values of w will yield rods with larger effective thickness.

A description of the Lagrangian discretization can be found in Griffith and Lim [70] and is implemented in the open-source software IBAMR. The Eulerian discretization is discussed in the previous chapters.

5.2 Results

The helical filament is initialized as

$$\mathbf{X}(s,t) = \begin{cases} 0 & 0 \le s \le L_h \\ R_0\left(1 - e^{-c(s-L_h)^2}\right) & L_h \le s \le L_h + L_f. \end{cases} \tag{5.20}$$

The helical radius is 0 for the hook and gradually increases to a radius of R_0. Here, we choose $c = 2$. The radius R, as well as the other parameters are given in Table 5.1. The vector $\mathbf{D}^3$ is initially defined as the unit tangent vector to the flagellum, whereas $\mathbf{D}^1$ and $\mathbf{D}^2$ are the normal and binormal unit vectors. The flagellum is driven by a rotary motor. We fix the point at $s = 0$ in space and specify the rotation of the director vectors as

$$\mathbf{D}^1(0,t) = (\cos(2\pi\omega t), -\sin(2\pi\omega t), 0), \tag{5.21}$$

$$\mathbf{D}^2(0,t) = (\sin(2\pi\omega t), \cos(2\pi\omega t), 0), \tag{5.22}$$

$$\mathbf{D}^3(0,t) = (0,0,1), \tag{5.23}$$

where ω is the specified rotation rate. The sign of ω determines the direction of the rotation. The rotation of the motor generates a torque that is then transmitted to the flagellum through the compliant hook. Figure 5.2 shows a schematic view of the filament. The filament is placed in a computational domain Ω which is a cube of length $H = \frac{20}{3}L = 40\,\mu\text{m}$. We specify zero viscous stress at the boundaries, and zero flux on the components of the conformation tensor at outflow boundaries. The computational domain is discretized using $N = 512$ points in each direction on the finest level of the AMR grid. Figure 5.2 shows the flagellum in the computational domain. We discretize the Lagrangian structure so that the structure contains about one point per grid cell.

The motor is attached to the helical body by a flexible structure called the hook. The hook for an *E. coli* bacterium ranges from 50 µm to 80 µm. Here we specify the hook's length to be $2\Delta s = 60\,\mu\text{m}$. To make the hook flexible, we specify the bending modulus to be two orders of magnitude smaller than the filament. We also assume the hook to be intrinsically straight so that $\tau = \kappa = 0$.

Figure 5.3 shows the shape of superimposed flagella spinning in fluids of varying Deborah numbers. We note that the initial configuration of the flagellum is initialized in a left-handed helix, but spinning the motor counter-clockwise switches the stable configuration to a right-handed helix. Figure 5.4

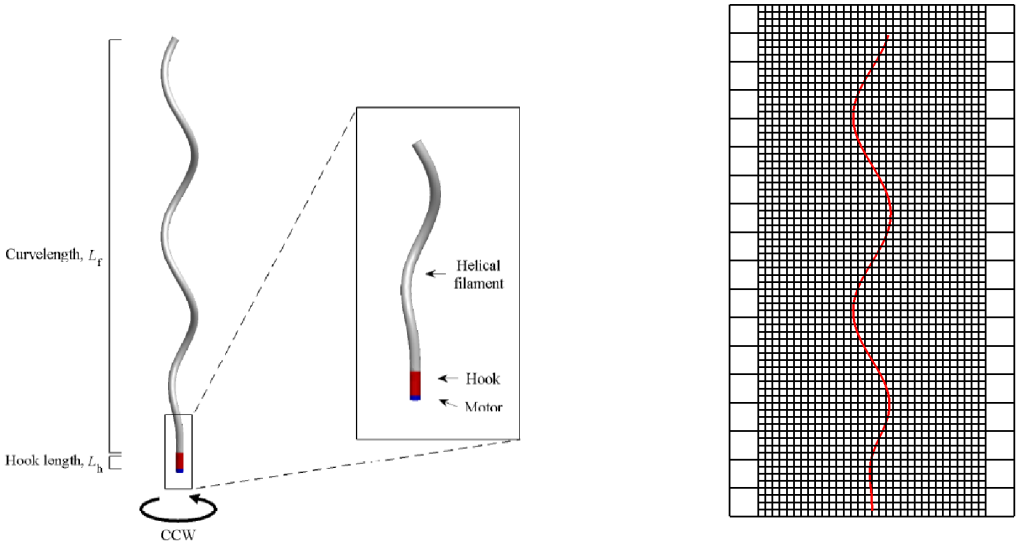

Figure 5.2: A schematic view of the flagellum (left). The body of the flagellum is attached to the rotary motor by a compliant hook. The right magnifies the computational domain in the viscinity of the flagellum. The edge of the finest level of the AMR grid is shown on the edges. The flagellum is discretized so that there is approximately one Lagrangian point per grid cell.

shows the pitch and radius from equation (5.8). A clear transition in handedness can be seen as the transition propagates upward along the flagellum. A zoomed image of the radius and pitch is shown in Figure 5.5. The flagellum in a Newtonian fluid with $\mathrm{De} = 0$ shows a much looser helix as well as being more twisted. Adding in the complex fluid tightens the helix by decreasing the radius as well as reducing the pitch. However, as we decrease the Deborah number, we decrease the radius and pitch even more.

We measure the volumetric flow rate through a disk D of radius $R = 1\,\mu\mathrm{m}$ at the plane $z = 5.5\,\mu\mathrm{m}$. While the flagellum's length is $6\,\mu\mathrm{m}$, the flagellum's coiled length is approximately $5.25\,\mu\mathrm{m}$. Therefore, the tip of the flagellum never passes through the disk D. The volumetric flow rate is calculated as

$$Q = \int_D \mathbf{u} \cdot \mathbf{n} dA, \tag{5.24}$$

where $\mathbf{n}$ is the outward unit normal. We note that the choice of region D determines the flux measure, because in the limit as the radius R tends to infinity, the volumetric flow rate Q tends to zero as we encapsulate more regurgitation flow in the Newtonian limit. Figure 5.6 shows the region through which the flow rate is calculated. Since we are specifying the rotation rate of the motor, we

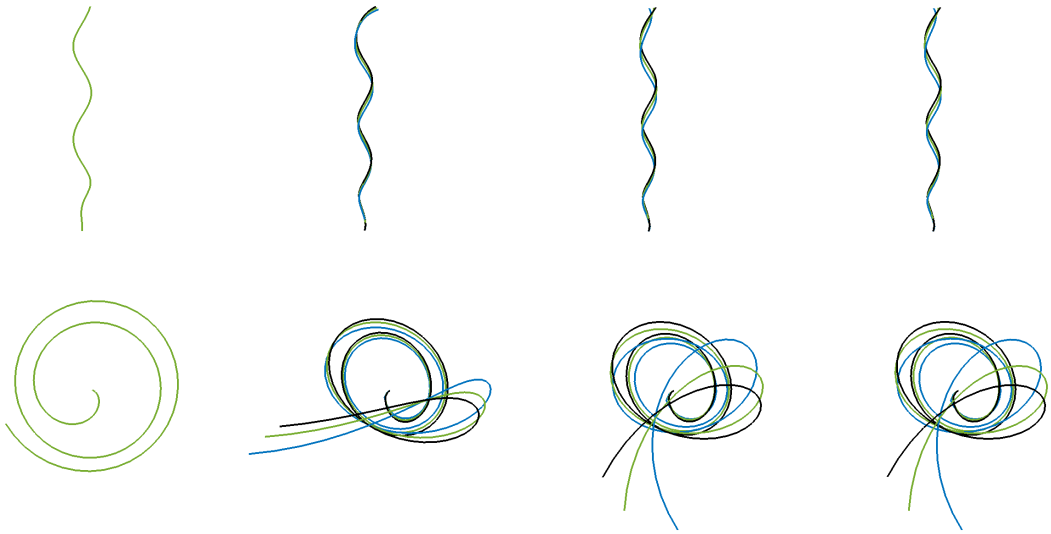

Figure 5.3: Views from the side (top row) and top (bottom row) of the flagella for varying Deborah numbers. Images are shown for the total rotations of the motor $t_{\mathrm{p}} = 0, 5, 10,$ and 12. Note that after 10 rotations, an approximate steady rotation of the flagella is reached.

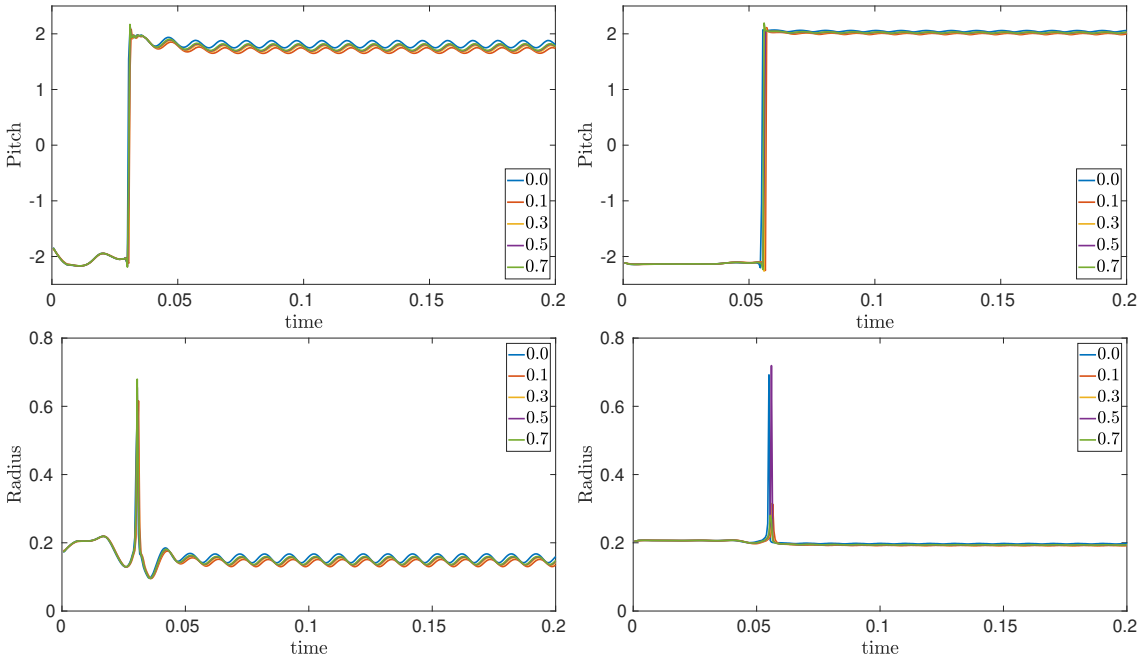

Figure 5.4: Pitch (top) and radius (bottom) for a point at the middle (left) and top end (right) of the flagellum for varying Deborah numbers. Note the transition of the handedness of the helix from left- to right-handed.

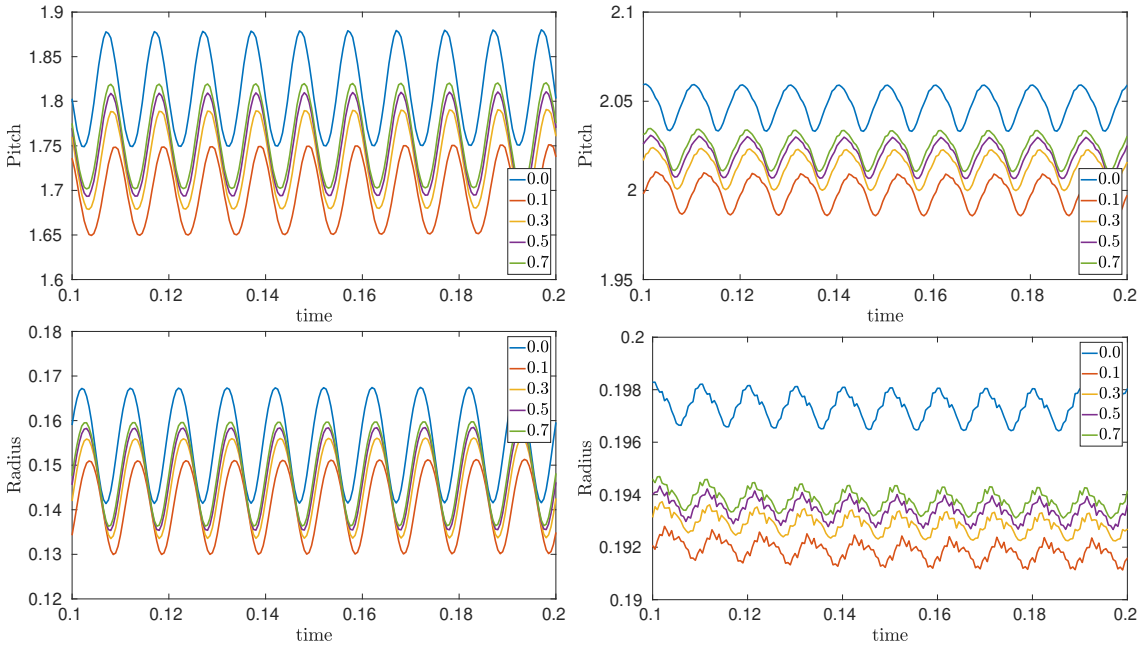

Figure 5.5: Magnification of Figure 5.4 showing the pitch (top) and radius (bottom) for a point at the middle (left) and top end (right) of the flagellum for varying Deborah numbers.

can also calculate the power required to turn the motor at the specified frequency. We calculate the total torque acting the hook from equation (5.2). Note that this represents torque being applied by the motor as well as the restoring torque from the rest of the filament.

Figure 5.7 shows the flow rate Q through the region D as well as the total torque acting on the hook. A positive torque represents the motor supplying greater power to the hook, while a negative torque represents the rest of the flagellum pushing down on the hook. A flagellum spinning in a fluid with a Deborah number of $\mathrm{De} = 0.1$ has the smallest net torque acting on the hook. As we increase the Deborah number, we see more negative torque acting on the hook. A flagellum spinning in a Newtonian fluid has about the same torque applied to the hook as a complex fluid with Deborah number $\mathrm{De} = 0.3$. This suggests that a complex fluid can *decrease* the net torque acting on the hook, possibly because the motor is supplying less power to maintain its rotation rate.

We fit a sine curve $f(t) = \bar{Q} + A\sin(wt + \phi)$ through the last ten rotations of the flagellum. We fix the frequency of the fit to be $w = 2\pi\,100\,\mathrm{rad/s}$ to match the rotation rate of the motor. The value $\bar{Q}$ is the average flow rate through the region D, A is the magnitude of the oscillation, and *phi* is the phase of the oscillation. Table 5.2 shows the average torques and the parameters of the fit for time $0.1 \leq t \leq 0.2$ after the flagellum has reached a steady spinning rate. Although the

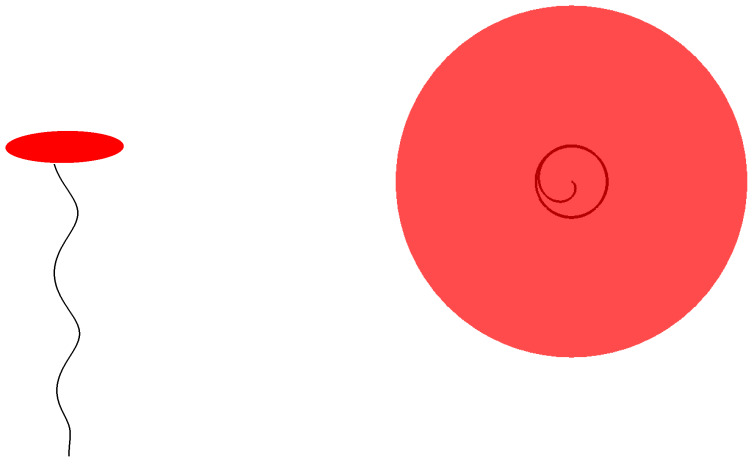

Figure 5.6: The red region indicates the the area through which the volumetric flow rate is computed.

De	average torque	$\bar{Q}$	A	ϕ
0.0	$-1.219 \times 10^{-4}\,\mathrm{g\,\mu m/s^2}$	$45.51\,\mu\mathrm{m}^3/\mathrm{s}$	$5.649\,\mu\mathrm{m}^3/\mathrm{s}$	$0.956\,\mathrm{rad}$
0.1	$-9.601 \times 10^{-5}\,\mathrm{g\,\mu m/s^2}$	$45.15\,\mu\mathrm{m}^3/\mathrm{s}$	$1.646\,\mu\mathrm{m}^3/\mathrm{s}$	$-0.481\,\mathrm{rad}$
0.3	$-1.250 \times 10^{-4}\,\mathrm{g\,\mu m/s^2}$	$46.17\,\mu\mathrm{m}^3/\mathrm{s}$	$3.317\,\mu\mathrm{m}^3/\mathrm{s}$	$0.411\,\mathrm{rad}$
0.5	$-1.310 \times 10^{-4}\,\mathrm{g\,\mu m/s^2}$	$45.81\,\mu\mathrm{m}^3/\mathrm{s}$	$4.120\,\mu\mathrm{m}^3/\mathrm{s}$	$0.583\,\mathrm{rad}$
0.7	$-1.324 \times 10^{-4}\,\mathrm{g\,\mu m/s^2}$	$45.58\,\mu\mathrm{m}^3/\mathrm{s}$	$4.583\,\mu\mathrm{m}^3/\mathrm{s}$	$0.669\,\mathrm{rad}$

Table 5.2: The average torque supplied to the hook and the parameters of the sine curve fit for varying Weissenberg numbers for the time $0.1 \leq t \leq 0.2$.

net torque decreases significantly for De $= 0.1$, the average flow rate remains relatively constant for all the Deborah numbers tested. The amplitude, however, shows a significant variation among the Deborah numbers. The flagellum spinning in a Newtonian fluid achieves the largest change in amplitude in the flow rate, while the flagellum spinning in a complex fluid with Deborah number De $= 0.1$ achieves the lowest variation. As we increase the Deborah number further, we achieve larger amplitudes further approaching the Newtonian value. This suggests that the relaxation rate of the polymers can significantly hinder the ability of the flagellum to successfully pump fluid. There is also a significant phase lag between the oscillations in the flux that follows the same pattern as the amplitude which suggests the presence of the complex fluid affects how quickly the motion of the motor travels up the flagellum. We expect that additional probing of the Deborah numbers to yield more insight into these discussed patterns.

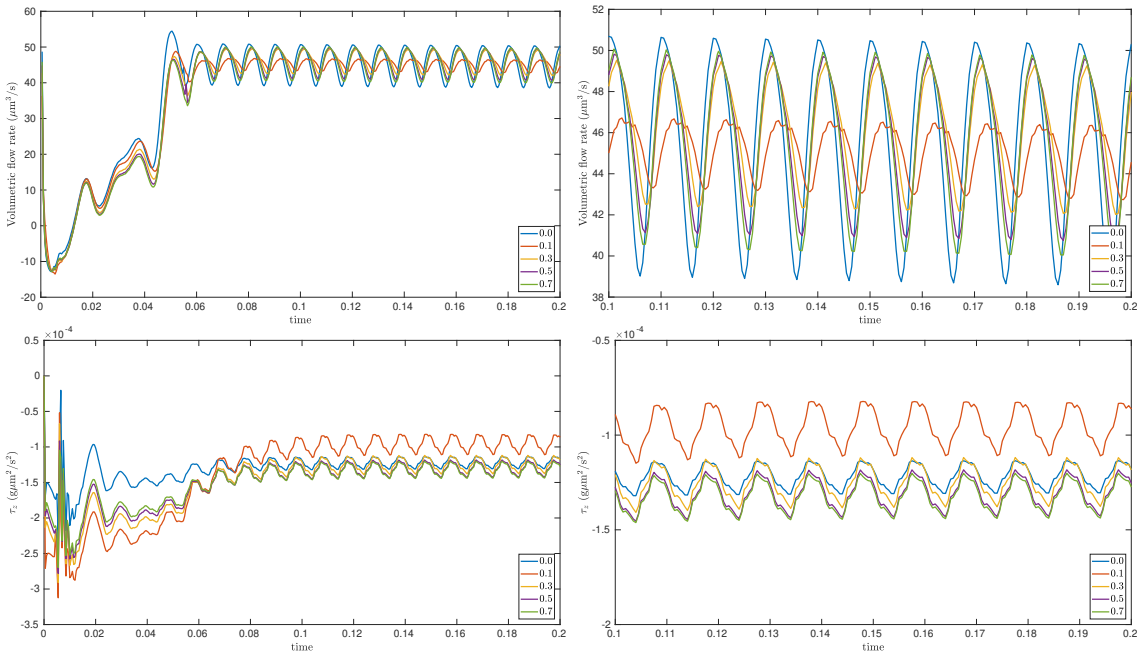

Figure 5.7: The volumetric flow rate and torque generated by a flagellum spinning in fluids of varying Deborah numbers. The right column shows a zoomed in image of the left.

APPENDIX A

WENO INTERPOLATION

Here we describe the process of WENO interpolation and compare it against standard interpolation. We start with a review of Newton interpolation. We present the analyis in one spatial dimension. For higher dimensions, we proceed in a dimension by dimension reconstruction. The fundamental interpolation problem is given a set of data points $\{(x_i, f(x_i) = y_i)\}_{i=0}^{N}$, we wish to construct a polynomial $p(x)$ that interpolates the data points, $p(x_j) = y_j$. If we write $p(x) = \sum_{i=0}^{N} \tilde{a}_i x^i$, then apply the interpolation condition, we have

$$p(x_j) = y_j = \sum_{i=0}^{N} \tilde{a}_i x_j^i \text{ for } j = 0, \ldots, N, \tag{A.1}$$

which is an $(N+1)$ by $(N+1)$ system that we can solve for the coefficients $\tilde{a}_i$. However, this results in a matrix that is ill conditioned as N becomes large and the points get closer together. Alternatively, in the Newton form of the interpolating polynomial, we choose the basic form of the polynomial to be

$$p(x_j) = a_0 + a_1 (x - x_0) + a_2 (x - x_0)(x - x_1) + \cdot + a_N (x - x_0) \cdot (x - x_{N-1}). \tag{A.2}$$

With this form of $p(x)$, we can evaulate the polynomial at the interpolant points:

$$\begin{aligned} p(x_0) &= a_0, \\ p(x_1) &= a_0 + a_1 (x_1 - x_0), \\ p(x_2) &= a_0 + a_1 (x_2 - x_0) + a_2 (x_2 - x_0)(x_2 - x_1). \end{aligned}$$

This allows us to define the coefficients in a recursive fashion

$$f[x_i] = y_i, \quad f[x_i, \cdots, x_{i+j}] = \frac{f[x_{i+1}, \cdots, x_{i+j}] - f[x_i, \cdots, x_{i+j-1}]}{x_{i+j} - x_i}. \tag{A.3}$$

These are known as *divided differences*, and the polynomial becomes

$$p(x) = \sum_{i=0}^{N} f[x_0, \cdots, x_i] \prod_{k=0}^{i-1} (x - x_k) \,. \tag{A.4}$$

We consider three test functions,

$$f_1(x) = \cos(\pi x) \,, \tag{A.5}$$

$$f_2(x) = \frac{1}{1 + 25x^2}, \tag{A.6}$$

$$f_3(x) = H(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ 0 & \text{if } x < 0. \end{cases} \tag{A.7}$$

Figure A.1 shows the resulting interpolants. For functions with derivatives with controlled oscillations, such as $f_1(x)$, we obtain excellent results. For functions where the derivatives have large gradients, such as $f_2(x)$, or for which the derivative is discontinuous, such as $f_3(x)$, we see oscillations in the interpolant.

We can fix this issue by returning to the divided difference. For smooth functions, the divided difference satisfies $f[x_i, \cdots, x_{i+j}] = \frac{f^{(j)}(\xi)}{j!}$. However, if $f(x)$ is not smooth, then $f[x_i, \cdots, x_{i+j}] \sim \mathcal{O}\left((\Delta x)^{-j}\right)$. In other words, the divided difference gives us a notion of the smoothness of the function. If the divided difference is small, then the derivative of $f(x)$ is small, making $f(x)$ smooth. If the divided difference is large, either $f(x)$ is discontinuous or the derivative has large changes in it. We can use this idea to build up a smooth stencil for our interpolation. We start with two points $\{x_i, x_{i+1}\}$ so that $p_1(x) = f[x_i] + f[x_i, x_{i+1}](x - x_i)$. We now add a second point to get our second interpolant

$$L(x) = p_1(x) + f[x_{i-1}, x_i, x_{i+1}](x - x_i)(x - x_{i+1}) \,,$$

$$R(x) = p_2(x) + f[x_i, x_{i+1}, x_{i+2}](x - x_i)(x - x_{i+1}) \,.$$

The only difference between the two interpolants is the choice of divided difference. We then choose the interpolant that has the *smallest* divided difference. We can then repeat this process until we've hit the desired degree of polynomial N. This method is called Essentially Non-Oscillatory (ENO)

interpolation because it builds up a stencil so that the interpolating polynomial is smooth.

When choosing our stencil, we looked at a total of $2N-1$ points, but ended up choosing a stencil that only had N points. At smooth points, we could use all $2N-1$ points to get a polynomial of degree $2N-2$. However, when the function is not smooth, we would like to use a smaller number of points in a weighted direction to get a smoother interpolant. Suppose we have k stenciles $S_r(i) = \{x_{i-r}, \cdot, x_{i-r+k-1}\}$ for $r = 0, \cdot, k-1$ that give k different reconstructions $y^r_{i+\frac{1}{2}} = \sum_{j=0}^{k-1} c_{rj} q_{i-r+j}$. Then a Weighted Essentially Non-Oscillatory (WENO) reconstruction uses a convex combination of all the reconstructions

$$y_{i+\frac{1}{2}} = \sum_{r=0}^{k-1} w_r y^r_{i+\frac{1}{2}}, \quad w_r \geq 0, \quad \sum_{r=0}^{k-1} w_r = 1. \tag{A.8}$$

If $f(x)$ is smooth in all the candidate stencils, then there exists constants d_r such that

$$y_{i+\frac{1}{2}} = \sum_{r=0}^{k-1} d_r y^r_{i+\frac{1}{2}} = f(x_{i+\frac{1}{2}}) + \mathcal{O}\left(\Delta x^{2k-1}\right). \tag{A.9}$$

We would like $w_r \approx d_r$ when $f(x)$ is smooth in stencil S_r and $w_r \approx 0$ otherwise. We choose w_r to be of the form

$$w_r = \frac{\alpha_r}{\sum_{i=0}^{k-1} \alpha_i}, \quad \alpha_r = \frac{d_r}{(\epsilon + \beta_r)^2}, \tag{A.10}$$

where β_r is a smoothness indicator of the function $f(x)$ in the stencil S_r. We would like β_r to be *large* when near a discontinuity. Here, we choose β_r to be

$$\beta_r = \sum_{l=1}^{k-1} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \Delta x^{2l-1} \left(\frac{\partial^l p_r(x)}{\partial x^l}\right)^2 dx. \tag{A.11}$$

To demonstrate the ability of WENO interpolation to capture sharp features, we interpolate the function $f_4(x) = \sin(2\pi x)^4$. This function is a classical function used to test interpolation methods due to its steep gradients despite being a C^∞ function. We evaluate the function $f_4(x)$ on cell sides $x_i = i\Delta x$ and then reconstruct the function at cell centers $x_{i+\frac{1}{2}} = \left(i + \frac{1}{2}\right)\Delta x$ using a 7th order WENO approximation. Figure A.2 shows the results. The function is shown in blue with the values of the interpolant shown in red circles. The right figures show the weights w_r that are used for the interpolation. The circles are the actual weights that are being used by the interpolation, while the

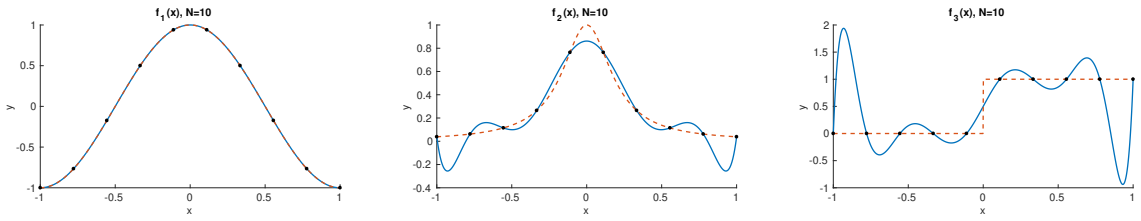

Figure A.1: The three test functions $f_1(x) = \cos(\pi x)$, $f_2(x) = \frac{1}{1+25x^2}$, and $f_3(x) = H(x)$ and the interpolants with $N = 10$ points.

lines show the weights that would be required to get 7th order approximations. Note that as we increase the number of points, we use weights that are closer to the ideal weights.

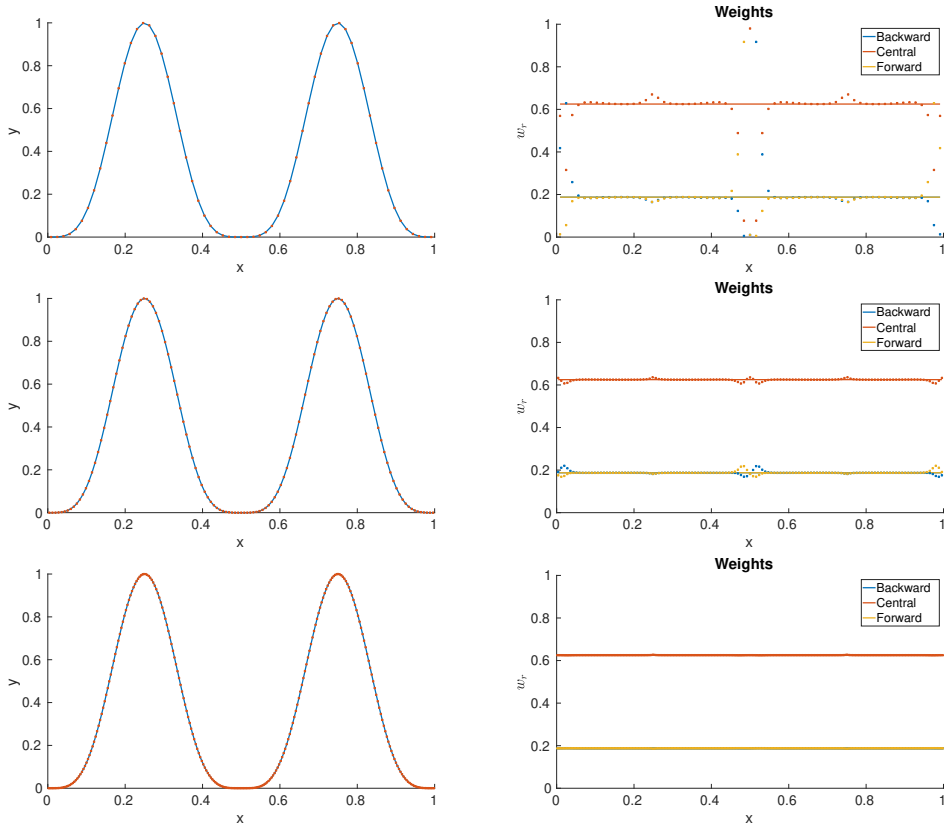

Figure A.2: Seventh-order WENO reconstructions of $f_4(x) = \sin(2\pi x)^4$ for $N = 64, 128$, and 256 points. Left: The function is shown in blue with the resulting interpolant evaluations shown in red circles. Right: The weights used in the reconstruction.

REFERENCES

[1] W. Ko, S. Lim, W. Lee, Y. Kim, H. C. Berg, and C. S. Peskin, "Modeling polymorphic transformation of rotating bacterial flagella in a viscous fluid," *Physical Review E*, vol. 95, no. 6, pp. 1–12, 2017.

[2] S. Claus and T. Phillips, "Viscoelastic flow around a confined cylinder using spectral/hp element methods," *Journal of Non-Newtonian Fluid Mechanics*, vol. 200, pp. 131–146, oct 2013.

[3] L. J. Fauci and A. McDonald, "Sperm motility in the presence of boundaries," *Bulletin of Mathematical Biology*, vol. 57, pp. 679–699, sep 1995.

[4] R. Bansil, J. P. Celli, J. M. Hardcastle, and B. S. Turner, "The Influence of Mucus Microstructure and Rheology in Helicobacter pylori Infection," *Frontiers in Immunology*, vol. 4, p. 310, oct 2013.

[5] P. A. Vasquez, Y. Jin, E. Palmer, D. Hill, and M. G. Forest, "Modeling and Simulation of Mucus Flow in Human Bronchial Epithelial Cell Cultures - Part I: Idealized Axisymmetric Swirling Flow," *PLOS Computational Biology*, vol. 12, p. e1004872, aug 2016.

[6] A. Morozov and S. E. Spagnolie, "Introduction to Complex Fluids," in *Complex Fluids in Biological Systems*, pp. 3–52, 2015.

[7] B. Thomases and M. Shelley, "Emergence of singular structures in Oldroyd-B fluids," *Physics of Fluids*, vol. 19, p. 103103, oct 2007.

[8] Y. Fan, R. Tanner, and N. Phan-Thien, "Galerkin/least-square finite-element methods for steady viscoelastic flows," *Journal of Non-Newtonian Fluid Mechanics*, vol. 84, pp. 233–256, aug 1999.

[9] P. Pakdel, S. H. Spiegelberg, and G. H. McKinley, "Cavity flows of elastic liquids: Two-dimensional flows," *Physics of Fluids*, vol. 9, no. 11, pp. 3123–3140, 1997.

[10] R. Larson, *Constitutive Equations for Polymer Melts and Solutions*. Butterworth Publishers, 1988.

[11] J. K. Wróbel, S. Lynch, A. Barrett, L. Fauci, and R. Cortez, "Enhanced flagellar swimming through a compliant viscoelastic network in Stokes flow," *Journal of Fluid Mechanics*, vol. 792, pp. 775–797, apr 2016.

[12] M. A. Hulsen, "A sufficient condition for a positive definite configuration tensor in differential models," *Journal of Non-Newtonian Fluid Mechanics*, vol. 38, pp. 93–100, jan 1990.

[13] J. M. Dealy, "Weissenberg and Deborah Numbers - their definition and use," *Rheol. Bull.*, vol. 79, pp. 14–18, 2010.

[14] M. J. Berger and J. Oliger, "Adaptive mesh refinement for hyperbolic partial differential equations," *Journal of Computational Physics*, vol. 53, pp. 484–512, mar 1984.

[15] M. J. Berger and P. Colella, "Local Adaptive Mesh Refinement for Shock Hydrodynamics," tech. rep., 1989.

[16] M. T. Jones and P. E. Plassmann, "Adaptive refinement of unstructured finite-element meshes," *Finite Elements in Analysis and Design*, vol. 25, pp. 41–60, mar 1997.

[17] S. Muzaferija, *Adaptive Finite Volume Method for Flow Prediction Using Unstructured Meshes and Multigrid Approach*. PhD thesis, University of London, 1994.

[18] F. H. Harlow and J. E. Welch, "Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface," *Physics of Fluids*, vol. 8, p. 2182, dec 1965.

[19] B. E. Griffith, R. D. Hornung, D. M. McQueen, and C. S. Peskin, "An adaptive, formally second order accurate version of the immersed boundary method," *Journal of Computational Physics*, vol. 223, no. 1, pp. 10–49, 2007.

[20] P. Colella and P. R. Woodward, "The Piecewise Parabolic Method (PPM) for gas-dynamical simulations," *Journal of Computational Physics*, vol. 54, no. 1, pp. 174–201, 1984.

[21] W. J. Rider, J. A. Greenough, and J. R. Kamm, "Accurate monotonicity- and extrema-preserving methods through adaptive nonlinear hybridizations," *Journal of Computational Physics*, vol. 225, no. 2, pp. 1827–1848, 2007.

[22] B. E. Griffith, "An accurate and efficient method for the incompressible Navier-Stokes equations using the projection method as a preconditioner," *Journal of Computational Physics*, vol. 228, no. 20, pp. 7565–7595, 2009.

[23] D. I. Ketcheson, M. Parsani, and R. J. LeVeque, "High-Order Wave Propagation Algorithms for Hyperbolic Systems," *SIAM Journal on Scientific Computing*, vol. 35, pp. A351–A377, jan 2013.

[24] C.-W. Shu, "Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws," in *ICASE Report*, no. 97-65, pp. 325–432, 1998.

[25] R. D. Guy and A. L. Fogelson, "A wave propagation algorithm for viscoelastic fluids with spatially and temporally varying properties," *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 25-28, pp. 2250–2264, 2008.

[26] R. D. Hornung and S. R. Kohn, "Managing application complexity in the SAMRAI object-oriented framework," *Concurrency and Computation: Practice and Experience*, vol. 14, pp. 347–368, apr 2002.

[27] S. Balay, A. Abhyankar, A. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. Gropp, D. Karpeyev, D. Kaushik, M. Knepley, D. May, L. Curfman McInnes, R. Mills, T. Munson, K. Rupp, P. Sanan, B. Smith, S. Zampini, H. Zhang, and H. Zhang, "PETSc Users Manual," tech. rep., Argonne National Laboratory, 2018.

[28] R. Fattal and R. Kupferman, "Time-dependent simulation of viscoelastic flows at high Weissenberg number using the log-conformation representation," *Journal of Non-Newtonian Fluid Mechanics*, vol. 126, pp. 23–37, feb 2005.

[29] R. Sousa, R. Poole, A. Afonso, F. Pinho, P. Oliveira, A. Morozov, and M. Alves, "Lid-driven cavity flow of viscoelastic liquids," *Journal of Non-Newtonian Fluid Mechanics*, vol. 234, pp. 129–138, aug 2016.

[30] E. Lauga and T. R. Powers, "The hydrodynamics of swimming microorganisms," *Reports on Progress in Physics*, vol. 72, p. 096601, sep 2009.

[31] C. S. Peskin and D. M. McQueen, "Mechanical equilibrium determines the fractal fiber architecture of aortic heart valve leaflets," *American Journal of Physiology-Heart and Circulatory Physiology*, vol. 266, pp. H319–H328, jan 1994.

[32] C. S. Peskin and D. M. McQueen, "A three-dimensional computational method for blood flow in the heart I. Immersed elastic fibers in a viscous incompressible fluid," *Journal of Computational Physics*, vol. 81, pp. 372–405, apr 1989.

[33] J. K. Wróbel, R. Cortez, and L. Fauci, "Modeling viscoelastic networks in Stokes flow," *Physics of Fluids*, vol. 26, p. 113102, nov 2014.

[34] C. S. Peskin, "The immersed boundary method," *Acta Numerica*, vol. 11, pp. 479–517, jan 2002.

[35] B. E. Griffith, R. D. Hornung, D. M. McQueen, and C. S. Peskin, "Parallel and Adaptive Simulation of Cardiac Fluid Dynamics," in *Advanced Computational Infrastructures for Parallel and Distributed Adaptive Applications*, pp. 105–130, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2009.

[36] L. A. Miller and C. S. Peskin, "Flexible clap and fling in tiny insect flight.," *The Journal of experimental biology*, vol. 212, pp. 3076–90, oct 2009.

[37] C. L. Hamlet, K. A. Hoffman, E. D. Tytell, and L. J. Fauci, "The role of curvature feedback in the energetics and dynamics of lamprey swimming: A closed-loop model," *PLOS Computational Biology*, vol. 14, p. e1006324, aug 2018.

[38] L. Zhang, A. Gerstenberger, X. Wang, and W. K. Liu, "Immersed finite element method," *Computer Methods in Applied Mechanics and Engineering*, vol. 193, pp. 2051–2067, may 2004.

[39] S. Haeri and J. Shrimpton, "On the application of immersed boundary, fictitious domain and body-conformal mesh methods to many particle multiphase flows," *International Journal of Multiphase Flow*, vol. 40, pp. 38–55, apr 2012.

[40] Y.-H. Tseng and J. H. Ferziger, "A ghost-cell immersed boundary method for flow in complex geometry," *Journal of Computational Physics*, vol. 192, pp. 593–623, dec 2003.

[41] B. E. Griffith and X. Luo, "Hybrid finite difference/finite element immersed boundary method," *Journal of bacteriology*, vol. 90, pp. 443–445, dec 2016.

[42] Y. Liu and Y. Mori, "Properties of Discrete Delta Functions and Local Convergence of the Immersed Boundary Method," *SIAM Journal of Numerical Analysis*, vol. 50, no. 6, pp. 2986–3015, 2012.

[43] R. J. LeVeque and Z. Li, "The Immersed Interface Method for Elliptic Equations with Discontinuous Coefficients and Singular Sources," *SIAM Journal on Numerical Analysis*, vol. 31, pp. 1019–1044, aug 1994.

[44] S. Xu and Z. J. Wang, "Systematic Derivation of Jump Conditions for the Immersed Interface Method in Three-Dimensional Flow Simulation," *SIAM Journal on Scientific Computing*, vol. 27, pp. 1948–1980, jan 2006.

[45] S. Xu and Z. J. Wang, "An immersed interface method for simulating the interaction of a fluid with moving boundaries," *Journal of Computational Physics*, vol. 216, no. 2, pp. 454–493, 2006.

[46] Z. Li and M.-C. Lai, "The Immersed Interface Method for the Navier-Stokes Equations with Singular Forces," *Journal of Computational Physics*, vol. 171, pp. 822–842, aug 2001.

[47] L. Yang, *The Immersed Interface Method for Flow Around Non-Smooth Boundaries and Its Parallelization*. PhD thesis, Southern Methodist University, 2017.

[48] E. M. Kolahdouz, A. P. S. Bhalla, B. A. Craven, and B. E. Griffith, "An Immersed Interface Method for Discrete Surfaces," *(Submitted)*, pp. 1–41, dec 2018.

[49] B. S. Kirk, J. W. Peterson, R. H. Stogner, and G. F. Carey, "libMesh : a C++ library for parallel adaptive mesh refinement/coarsening simulations," *Engineering with Computers*, vol. 22, pp. 237–254, dec 2006.

[50] R. G. Owens, C. Chauvière, and T. N. Philips, "A locally-upwinded spectral technique (LUST) for viscoelastic flows," *Journal of Non-Newtonian Fluid Mechanics*, vol. 108, no. 1-3, pp. 49–71, 2002.

[51] P. J. Oliveira and A. I. P. Miranda, "A numerical study of steady and unsteady viscoelastic flow past bounded cylinders," *Journal of Non-Newtonian Fluid Mechanics*, vol. 127, no. 1, pp. 51–66, 2005.

[52] H.-S. Dou and N. Phan-Thien, "The flow of an Oldroyd-B fluid past a cylinder in a channel: adaptive viscosity vorticity (DAVSS-ω) formulation," *Journal of Non-Newtonian Fluid Mechanics*, vol. 87, pp. 47–73, oct 1999.

[53] M. Alves, F. Pinho, and P. Oliveira, "The flow of viscoelastic fluids past a cylinder: finite-volume high-resolution methods," *Journal of Non-Newtonian Fluid Mechanics*, vol. 97, pp. 207–232, 2001.

[54] G. I. Taylor, "The Formation of Emulsions in Definable Fields of Flow," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 146, pp. 501–523, oct 1934.

[55] B. J. Bentley and L. G. Leal, "A computer-controlled four-roll mill for investigations of particle and drop dynamics in two-dimensional linear shear flows," *Journal of Fluid Mechanics*, vol. 167, p. 219, jun 1986.

[56] J. S. Lee, R. Dylla-Spears, N. P. Teclemariam, and S. J. Muller, "Microfluidic four-roll mill for all flow types," *Applied Physics Letters*, vol. 90, p. 074103, feb 2007.

[57] M. J. Pelczar, I. C. Gunsalus, and R. Y. Stanier, "The Bacteria, A Treatise on Structure and Function. Volume I: Structure," *AIBS Bulletin*, vol. 10, p. 37, dec 1960.

[58] H. C. Berg, "The Rotary Motor of Bacterial Flagella," *Annual Review of Biochemistry*, vol. 72, pp. 19–54, jun 2003.

[59] N. C. Darnton, L. Turner, S. Rojevsky, and H. C. Berg, "On torque and tumbling in swimming Escherichia coli.," *Journal of bacteriology*, vol. 189, pp. 1756–64, mar 2007.

[60] S. J. Lighthill, *Mathematical Biofluiddynamics*. Society for Industrial and Applied Mathematics, jan 1975.

[61] E. Lauga, "Propulsion in a viscoelastic fluid," *Physics of Fluids*, vol. 19, p. 083104, aug 2007.

[62] J. Teran, L. Fauci, and M. Shelley, "Viscoelastic Fluid Response Can Increase the Speed and Efficiency of a Free Swimmer," *Physical Review Letters*, vol. 104, p. 038101, jan 2010.

[63] B. Thomases and R. D. Guy, "Mechanisms of Elastic Enhancement and Hindrance for Finite-Length Undulatory Swimmers in Viscoelastic Fluids," *Physical Review Letters*, vol. 113, p. 098102, aug 2014.

[64] H. C. Fu, C. W. Wolgemuth, and T. R. Powers, "Swimming speeds of filaments in nonlinearly viscoelastic fluids.," *Physics of fluids (Woodbury, N.Y. : 1994)*, vol. 21, p. 33102, mar 2009.

[65] S. E. Spagnolie, B. Liu, and T. R. Powers, "Locomotion of Helical Bodies in Viscoelastic Fluids: Enhanced Swimming at Large Helical Amplitudes," *Physical Review Letters*, vol. 111, p. 068101, aug 2013.

[66] B. Liu, T. R. Powers, and K. S. Breuer, "Force-free swimming of a model helical flagellum in viscoelastic fluids.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 108, pp. 19516–20, dec 2011.

[67] R. E. Goldstein, A. Goriely, G. Huber, and C. W. Wolgemuth, "Bistable Helices," *Physical Review Letters*, vol. 84, pp. 1631–1634, feb 2000.

[68] N. C. Darnton and H. C. Berg, "Force-extension measurements on bacterial flagella: triggering polymorphic transformations.," *Biophysical journal*, vol. 92, pp. 2230–6, mar 2007.

[69] S. Lim and C. S. Peskin, "Fluid-mechanical interaction of flexible bacterial flagella by the immersed boundary method," *Physical Review E*, vol. 85, p. 036307, mar 2012.

[70] B. E. Griffith and S. Lim, "Simulating an elastic ring with bend and twist by an adaptive generalized immersed boundary method," *Communications in Computational Physics*, 2012.