UNIVERSITY OF CALIFORNIA

Los Angeles

Simulation of a Sedimenting Sphere in a Viscoelastic Fluid with OpenFOAM

A thesis submitted in partial satifaction of the requirements for the degree Master of Science

in Chemical Engineering

by

Claire Love

2024

ABSTRACT OF THE THESIS

Simulation of a Sedimenting Sphere in a Viscoelastic Fluid with OpenFOAM

by

Claire Love

Master of Science in Chemical Engineering

University of California, Los Angeles, 2024

Professor Joseph Daniel Peterson, Chair

The simulation of viscoelastic flow past a sphere, or a sedimenting sphere, has attracted considerable interest since being established as a benchmark problem in computational flow dynamics [23]. As a geometry exhibiting both shearing and extensional flows, the sedimenting sphere design has the potential to probe properties of viscoelastic fluids under various conditions. In particular, we explore the low Reynolds, high Weissenberg number flow regime, in which elastic forces dominate. Using OpenFOAM, an open-source computational fluid dynamics software, we are able to implement immersed boundary conditions so that we can demonstrate unsteady startup in addition to steady-state dynamics. We find that these conditions as specified are currently unstable for reasonable sphere densities under typical gravitational acceleration, and therefore opt for unrealistically high densities and reduced accelerations (see Table 1). Nonetheless, we are able to capture realistic steady-state conditions, which we compare to the Faxen wall correction. We find that our boundary conditions, described in section 2, perform better than the Faxen correction for certain geometries. In future research, we hope to stabilize our startup dynamics for reasonable sphere densities and gravitational acceleration.

The dissertation of Claire Love is approved.

Thaiesha Andrea Wright

Carlos Gilber Morales Guio

Joseph Daniel Peterson, Chair

University of California, Los Angeles

2024

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background and Motivation

The physics of fluids in motion is a very old and rich area of study in chemical engineering, and its history of mathematical analysis includes figures like Leonard Euler and Albert Einstein[8, 7]. One classical problem in fluid mechanics is the study of sedimenting spheres in an unbounded fluid domain - when a heavy sphere is placed in a less dense fluid, it sinks to the bottom. The manner in which it sinks (whether steady or unsteady) and the velocity at which it falls will be a function of the fluid properties, the sphere's density, and the accelerating force (e.g. gravity) [21].

For very low Reynolds number flows of Newtonain fluids (dominated by viscous forces with negligible inertial effects) a classical result is that the sphere will sediment with a constant velocity that scales positively with the bouyant forces and inversely with the fluid viscosity [5]. For viscoelastic fluids, however, elastic stresses in the fluid develop transiently and the sphere/fluid system can exhibit unsteady motion typical of any damped harmonic oscillator system [22]. In some recent experiments, certain formulations of viscoelastic fluids even show persistent chaotic fluctuations that do not decay over time - the precise mechanism for these chaotic fluctuations is often attributed to microscopic material transformations in the fluid itself [40], but in general chaotic flows of viscoelastic fluids can also arise due to the confluence of elastic and inertial forces in complex flow geometries [35].

While the sedimentation of spheres in viscoelastic fluids has been studied previously [21, 40, 23], unresolved questions surrounding experimental observations of unsteady sphere sedimentation demonstrate the need for continued computational and theoretical work on the subject. In this thesis, we primarily focus on how the choices of boundary conditions in a simulation can lead to faster and more interpretable calculations.

The organization of this thesis is as follows. Through the remainder of section 1, we will review the fundamental physics of viscoelasticity (subsection 1.3), the equations of fluid motion (subsection 1.4), the computational platform upon which this research has been built (subsection 1.5), and the dimensionless equations that govern the problem itself (subsection 1.2). Section 2 goes in depth on the computational tools and mathematical methods employed in our study, from mesh generation to boundary conditions, section 3 provides discussion for a limited selection of results, and section 4 summarizes the overall thesis and discusses future potential research directions.

1.2 Dimensionless Quantities

In computational fluid dynamics (CFD), the Reynolds number (Re) is one of the most commonly referenced dimensionless quantities. The value of Re is the ratio of inertial forces to viscous forces, and can be useful in predicting fluid flow characteristics.

$$Re = \frac{2\rho_f u R_w}{\mu} \tag{1}$$

In Equation 1, ρ_f is the density of the fluid, u is the fluid velocity, and μ is the viscosity. In the context of the sedimenting sphere problem, R_w is the radius of the cylinder into which the sphere is dropped.

However, the Reynolds number alone is not sufficient to fully specify the general problem of sphere sedimentation in a confined geometry; we must also consider the degree of confinement (i.e. the ratio of the sphere radius to this cylinder radius, R_s/R_w) and the relative importance of inertia for the sphere and the fluid (i.e. ρ_s/ρ_f where ρ_s is the density of the sphere).

These three dimensionless groups ($Re, R_s/R_w$, and ρ_s/ρ_f) are sufficient to specify the problem of sphere sedimentation in a Newtonian fluid, but where the fluid is non-Newtonian additional information is required. For viscoelastic fluids in particular (c.f. section 1.3) we must compare the rate at which the fluid is deformed against the rate at which it is capable of relaxing its stress. This ratio is often called the "Weissenberg number":

$$Wi = \frac{\lambda u}{R_s} \tag{2}$$

Equation 2 compares a typical deformation rate u/R_s against a typical relaxation rate $1/\lambda$, where λ, which describes the time it takes for a polymer in a fluid to relax from an extended state towards an equilibrium state. Using Wi and Re, the elastic forces can be expressed in relation to the inertial forces.

$$El = \frac{Wi}{Re} \tag{3}$$

This quantity introduced in Equation 3 is the Elasticity number (El). The elasticity number is not independent of Wi and Re but for low Re systems it is often conventional to represent the influence of inertia through an elasticity number rather than a Reynolds number.

As a final side note, the Reynolds number requires a specified viscosity μ, but in the context of a non-Newtonian or viscoelastic fluid the viscosity is not always a well-defined concept. For this work, we define the viscosity as an additive combination of the solvent and polymer viscosities, $\mu = \mu_s + \mu_p$, where μ_s and μ_p are the solvent/polymer viscosities respectively. Very different dynamics are to be expected in the limiting cases where the polymer viscosity is much larger/smaller than the solvent viscosity, and so we introducethe dimensionless parameter $\beta = \frac{\mu_s}{\mu}$. For $\beta \sim 1$ the fluid is primarily Newtonian, and for $\beta \ll 1$ it is primarily polymeric. Thus the properties of a viscoelastic, sedimenting-sphere flow can be described through five dimensionless parameters: Re, Wi, β, R_s/R_w, and ρ_f/ρ_s. For our study we will primarily be concerned with low Reynolds number flows $Re \ll 1$, high density spheres, $\rho_f \gg \rho_s$, and low confinement $R_s/R_w \ll 1$. We will maintain comparable viscous/elastic forces, $\beta \sim 1/2$ and vary the strength of the flow across a range of Wi.

1.3 Viscoelasticity

For Newtonian fluids such as water, the fluid stresses are a linear function of the instantaneously imposed strain rate but not a function of its overall displacement from some past reference state (i.e. total strain) [25]. For ideal elastic solids, on the other hand, material stresses depend only on the total strain and are independent of the strain rate. Viscoelastic fluids are a unique class of materials that interpolate between a viscous-type response at long timescales and an elastic-type response at short times. Because of this time-scale dependence of their material properties, viscoelastic fluids are often characterized via small-amplitude oscillatory shear flow experiments.

In oscillatory shear experiments, a fluid is placed between two infinite parallel plates (or some laboratory approximation of the same) and the two plates translate parallel to one another with some oscillating relative velocity. The strain imposed on the fluid $\gamma(t)$ oscillates sinusoidally in time, as does the strain rate $\dot{\gamma}(t)$:

$$\gamma = \gamma_0 \sin(\omega t) \tag{4}$$

$$\dot{\gamma} = \frac{d\gamma}{dt} = \omega\gamma_0 \cos(\omega t) = \omega\gamma_0 \sin(\omega t - \pi/2) \tag{5}$$

If the fluid is an ideal Newtonian fluid or an ideal elastic solid, the stress $\sigma(t)$ will evolve in proportion to the strain rate or overall strain, respectively, where the constant of proportionality is a viscosity μ in the former case or a modulus G in the latter:

$$\sigma_N = \mu\dot{\gamma} \sim \sin(\omega t - \pi/2) \tag{6}$$

$$\sigma_E = G\gamma \sim \sin(\omega t) \tag{7}$$

For the general viscoelastic case, the stress response will interpolate between these two limits via a phase shift $\delta \in [0, \pi/2]$

$$\sigma \sim \sin(\omega t - \delta) \tag{8}$$

In a Newtonian fluid, the phase shift is a constant $\delta = \pi/2$, meaning the stress is orthogonal to the strain. Alternatively, a perfectly elastic material would show a direct relationship between strain and stress such that the phase shift is $\delta = 0$. For a viscoelastic fluid demonstrating some combination of viscous and elastic responses, the phase shift would fall somewhere on the interval $\delta \in [0, \pi/2]$, and the phase shift will tend to its upper/lower limit for very slow/fast frequency deformations, respectively. Viscoelastic fluids are encountered across various fields, as most biological fluids and polymer melts exhibit both viscous and elastic properties [15, 33]. Polymer melts in particular are essential in industrial processes, both as a manufacturing precursor and as a final product, depending on the specific application.

Turbulence in Newtonian fluid occurs due to viscous forces encountered at high *Re*, $Re \gg 1$, however in viscoelastic fluids at high Wi, chatoic flows analagous to turbulence (Elastic Turbulence, ET) can occur even at low $Re \to 0$ due to elastic forces [30]. Such elastic instabilities, once instantiated, can even delay the onset of purely inertial instabilities with increasing *Re*, resulting in a unique class of chaotic flows dubbed elasto-inertial turbulence (*EIT*) [35]. These instabilities often present potential barriers in viscoelastic fluid processing, as ET and EIT can limit processing speeds to subcritical values where flow remains stable. Therefore, it is necessary to develop a clear mechanistic understanding of viscoelastic fluid dynamics.

Towards that end, considerable research has been dedicated to accurately modeling viscoelastic behavior. The derivation of theoretical models builds from the same equations applied to Newtonian fluids.

1.4 Fluid Modeling Mathematics

In this subsection, we will develop equations of motion that describe the behavior of viscoelatic fluids. For any complex flow, we must be sure that the total mass of fluid is conserved. The general continuity equation is shown in Equation 9.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0 \tag{9}$$

Here ρ is the fluid density and $\boldsymbol{u}$ is the fluid velocity. In this current study, only incompressible fluids are considered, so that $\partial \rho / \partial t = 0$. This reduces the general continuity equation to that of an incompressible fluid, given in Equation 10.

$$\nabla \cdot \boldsymbol{u} = 0 \tag{10}$$

The continuity equation is a basic component for any constant-density fluid, as it ensures the material is conserved in the flow. To ensure conservation of momentum, we apply a momentum balance, as given in Equation 11.

$$\frac{\partial(\rho \boldsymbol{u})}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla \boldsymbol{P} + \nabla \cdot \boldsymbol{\tau} + \rho \boldsymbol{g} \tag{11}$$

This momentum balance, also called the Cauchy momentum equation, introduces pressure $P,$ the stress tensor τ, and the gravitational constant g.

By assuming constant ρ, as is characteristic of incompressible fluids, and substituting in the material derivative of $\boldsymbol{u}$,defined as $D\boldsymbol{u}/Dt = \partial \boldsymbol{u}/\partial t + \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u})$, Equation 11 can be further simplified.

$$\frac{D\boldsymbol{u}}{Dt} = -\frac{\nabla \boldsymbol{P}}{\rho} + \frac{\nabla \cdot \boldsymbol{\tau}}{\rho} + \boldsymbol{g} \tag{12}$$

Alternatively, the total stress, $\boldsymbol{\sigma}$, can be introduced to the momentum balance. The total stress differs from the shear stress by including volumetric stress due to pressure, as shown in the equation $\boldsymbol{\sigma} = -P\boldsymbol{I} + \boldsymbol{\tau}$, where $\boldsymbol{I}$ is an identity tensor. Taking the divergence of the total stress and applying the distributive property to the right-hand side results in $\nabla \cdot \boldsymbol{\sigma} = \nabla \boldsymbol{P} + \nabla \cdot \boldsymbol{\tau}$, which can be substituted into Equation 12 to give the momentum balance in terms of total stress.

$$\frac{D\boldsymbol{u}}{Dt} = \frac{\nabla \cdot \boldsymbol{\sigma}}{\rho} + \boldsymbol{g} \tag{13}$$

Taken together, Equations 10 and 13 comprise the Navier-Stokes equations for an incompressible flow. Applied in three dimensions, these are four equations (three for each velocity, one for continuity) involving ten variable; the pressure P, the three velocity components in $\boldsymbol{u}$, and the six independent values of the symmetric stress tensor $\boldsymbol{\sigma}$. Therefore, a third set of equations is needed to fully define the system; a constitutive model for stress tensor $\boldsymbol{\sigma}$, which describes the fluid response to deformation.

In a Newtonian fluid, the shear stress has a linear dependence on the velocity gradient. In general terms, this relationship could be expressed as $\boldsymbol{\tau} = \boldsymbol{T} : \nabla \boldsymbol{u}$, where $\boldsymbol{T}$ is a fourth order tensor. Imposing that the fluid must be isotropic (a property of a Stokesian fluid) means that $\boldsymbol{T}$ must be isotropic as well. Therefore, the tensor $\boldsymbol{T}$ could be described as an isotropic fourth order tensor.

$$T_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu \delta_{ik} \delta_{jl} + \gamma \delta_{il} \delta_{jk} \tag{14}$$

Equation 14 is the most general form of an isotropic fourth order tensor in index notation [14]. In Equation 14, λ, μ, and γ are constants and $\boldsymbol{\delta}$ is the Kronecker delta. The shear stress tensor must be symmetric in order to conserve angular momentum, which implies the tensor $\boldsymbol{T}$ must be symmetric. Consequently, T_{ijkl} must be equal to T_{ijkl} . This property implies that $\mu\delta_{ik}\delta_{jl}+\gamma\delta_{il}\delta_{jk} = \mu\delta_{jk}\delta_{il}+\gamma\delta_{jl}\delta_{ik}$, which can only be true if $\mu = \gamma$ [14]. Rearranging the equation for $\boldsymbol{T}$, $T_{ijkl} = \lambda\delta_{ij}\delta_{kl} + \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk})$, and substituting into the equation for shear stress gives $\tau_{ij} = [\lambda\delta_{ij}\delta_{kl} + \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk})]\partial u_k/\partial x_l$, where index notation is adopted to maintain consistency with $\boldsymbol{T}$ [14, 32]. Distributing the velocity results in $\boldsymbol{\tau}_{ij} = \lambda\delta_{ij}\frac{\delta u_k}{\delta x_k} + \mu\left(\frac{\delta u_i}{\delta x_j} + \frac{\delta u_j}{\delta x_j}\right)$. Switching back to vector notation, the equation for shear stress becomes Equation 15[32].

$$\boldsymbol{\tau} = \lambda(\nabla \cdot \boldsymbol{u})\boldsymbol{I} + \mu(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T) \tag{15}$$

Applying the previous assumption of incompressibility, the first term goes to 0, yielding Equation 16.

$$\boldsymbol{\tau} = \mu(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T) \tag{16}$$

To interpret Equation 16 in terms of physical processes, the velocity gradient tensor can be decomposed into its symmetric and anti-symmetric components: $\nabla\boldsymbol{u} = \frac{1}{2}(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T) + \frac{1}{2}(\nabla\boldsymbol{u} - (\nabla\boldsymbol{u})^T)$. From this equation, the strain tensor $\boldsymbol{\varepsilon} = \frac{1}{2}(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T)$ describes the “stretching” characteristics of the flow, and the vorticity tensor $\boldsymbol{\xi} = \frac{1}{2}(\nabla\boldsymbol{u} - (\nabla\boldsymbol{u})^T)$ captures the rotational characteristics of the flow [2]. Thus, we see that viscous stresses in a Newtonian fluid emerge in response to the rate at which the fluid is being stretched, and do not depend on its rotation:

$$\boldsymbol{\tau} = 2\mu\boldsymbol{\varepsilon} \tag{17}$$

In a Newtonian fluid the total stress tensor is given by:

$$\boldsymbol{\sigma} = -P\boldsymbol{I} + \mu(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T) \tag{18}$$

The derivation of the shear stress and substitution into Equation 18 gives a complete system of equations for describing the behavior of a Newtonian fluid.

Whereas the functional form of a constitutive relationship for stress in a Newtonian fluid can be derived on the basis of first principles, constitutive equations for non-newtonian fliuds are far more vairied in form and tend to require many more assumptions and approximations in their derivation. Here, we will focus on the derivation for one of the simplest viscoelastic constitutive relations.

First, for any non-Newtonian fluid, it is always possible to have some contribution to the stress that is Newtonian. In the context of a typical viscoelastic fluid, namely a polymer solution, we might attribute the Newtonian contribution to the solvent and the viscoelastic contribution to a polymer:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_s + \boldsymbol{\tau}_p \tag{19}$$

$$\boldsymbol{\tau}_s = 2\mu_s \boldsymbol{\epsilon} \tag{20}$$

where $\boldsymbol{\tau}_s$ and $\boldsymbol{\tau}_p$ are the solvent/polymer contributions to the stress tensor, respecitvely, and μ_s is a solvent viscosity. To solve for the viscoelastic component of the shear stress, a model for viscoelasticity must be introduced.

Maxwell's model for viscoelasticity treats the material as a damper in series with a spring, where the damper is analogous to viscous dissipation of force and the spring describes the elastic response [29]. In one dimension, this system is modelled by Equation 21.

$$\lambda \frac{d\boldsymbol{F_{e,x}}}{dt} + \boldsymbol{F_{e,x}} = \mu_p \frac{d\varepsilon_x}{dt} \tag{21}$$

In this equation, $\boldsymbol{F_{e,x}}$ is the elastic force along the x-axis and ε_x is the axial strain. Considering the spring and damper analogy, the quantity λ describes the rate of dampening and is therefore called the relaxation time, as was recovered in the derivation of Equation 15. The Maxwell model of viscoelasticity can accurately predict the dynamics of a spring-dashpot system, which serves as a physical representation of a polymer in solution. However, as a one-dimensional, frame-dependent model, Equation 21 is inadequate for describing a three-dimensional flow [22]. There are numerous mathematical methods to generalize the intuitions of equation 21, but here we will focus on those developed by Oldroyd in 1947 [28].These reformulations are shown in Equations 22 and 23.

$$\lambda \overset{\triangle}{\boldsymbol{\sigma}}_p + \sigma_p = 2\mu_p \boldsymbol{\varepsilon} \tag{22}$$

$$\lambda \overset{\triangledown}{\boldsymbol{\sigma}}_p + \sigma_p = 2\mu_p \boldsymbol{\varepsilon} \tag{23}$$

In these equations, the quantity μ is replaced with μ_p, which is the viscosity of the polymer. The lower convected derivative of the polymeric stress is defined as $\overset{\triangle}{\boldsymbol{\sigma}}_p = D\boldsymbol{\sigma_p}/Dt + \boldsymbol{\sigma_p} \bullet \nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T \bullet \boldsymbol{\sigma_p}$ and the upper convected derivative as $\overset{\triangledown}{\boldsymbol{\sigma}}_p = D\boldsymbol{\sigma_p}/Dt - \boldsymbol{\sigma_p} \bullet \nabla \boldsymbol{u} - (\nabla \boldsymbol{u})^T \bullet \boldsymbol{\sigma_p}$. Due to the frame-invariance methods applied, Equation 22 is called the Lower Convected Maxwell (LCM), and Equation 23 is called the Upper Convected Maxwell (UCM) [22]. Applying the relationship between the total stress and polymeric stress, $\boldsymbol{\sigma}_p$ can be solved with $\boldsymbol{\sigma}_p = \boldsymbol{\sigma} - 2\mu_s\boldsymbol{\varepsilon}$. Substituting this equation into Equations 22 and 23 gives $\boldsymbol{\sigma} - 2\mu_s\boldsymbol{\varepsilon} + \lambda(\overset{\triangle}{\boldsymbol{\sigma}} - 2\mu_s\overset{\triangle}{\boldsymbol{\varepsilon}}) = 2\mu_p\boldsymbol{\varepsilon}$ and $\boldsymbol{\sigma} - 2\mu_s\boldsymbol{\varepsilon} + \lambda(\overset{\triangledown}{\boldsymbol{\sigma}} - 2\mu_s\overset{\triangledown}{\boldsymbol{\varepsilon}}) = 2\mu_p\boldsymbol{\varepsilon}$, respectively. Rearranging and introducing the constants β and μ gives Equations 24 and 25.

$$\boldsymbol{\sigma} + \lambda\overset{\triangle}{\boldsymbol{\sigma}} = 2\mu(\boldsymbol{\varepsilon} + \lambda\beta\overset{\triangle}{\boldsymbol{\varepsilon}}) \tag{24}$$

$$\boldsymbol{\sigma} + \lambda\overset{\triangledown}{\boldsymbol{\sigma}} = 2\mu(\boldsymbol{\varepsilon} + \lambda\beta\overset{\triangledown}{\boldsymbol{\varepsilon}}) \tag{25}$$

In these equations, $\mu = \mu_p + \mu_s$. Expressed in this manner, Equation 24 is the Oldroyd-A constitutive equation, and Equation 25 is the Oldroyd-B consitutive equation. While both formulations are mathematically valid, only the latter is commonly used in rheology. One reason for this preference lies in the behavior of the stress tensor under shear flow [22]. The first normal stress difference, given by $\boldsymbol{\sigma}_{xx} - \boldsymbol{\sigma}_{yy}$, describes how a fluid element compresses or expands along the y-axis in response to force applied along the x-axis. Similarly, the second normal stress difference is given by $\boldsymbol{\sigma}_{yy} - \boldsymbol{\sigma}_{zz}$ and governs the behavior of the fluid along the z-axis in response to a stress along the y-axis. In a shearing flow of a Newtonian fluid along the x-axis, the first normal stress difference would describe the shear stress between the x and y planes such that $\boldsymbol{\sigma}_{xx} - \boldsymbol{\sigma}_{yy} > 0$, while the uniform response along the y-z plane would yield a second normal stress difference equal to 0, $\boldsymbol{\sigma}_{yy} - \boldsymbol{\sigma}_{zz} = 0$ [24]. This differs in non-Newtonian, viscoelastic fluids wherein the elastic nature of the fluid imparts a non-zero second normal stress difference. In the same uniaxial shearing flow, the first normal stress difference would still be greater than 0, but the second normal stress difference would be negative, $\boldsymbol{\sigma}_{yy} - \boldsymbol{\sigma}_{zz} < 0$, indicating a compressive force along the y-z plane in response to the flow [22]. The exact value of the predicted second normal stress difference differs between the two Oldroyd models; the Oldroyd-A model predicts a value such that the first normal stress difference is equal in magnitude, so that $\frac{\boldsymbol{\sigma}_{yy} - \boldsymbol{\sigma}_{zz}}{\boldsymbol{\sigma}_{xx} - \boldsymbol{\sigma}_{yy}} = -1$ [22]. The second normal stress difference in the Oldroyd-B model is relatively small in magnitude, such that this ratio is much closer to 0 [24]. Experimental data has shown that the more accurate value of this ratio is approximately -0.1, $\frac{\boldsymbol{\sigma}_{yy} - \boldsymbol{\sigma}_{zz}}{\boldsymbol{\sigma}_{xx} - \boldsymbol{\sigma}_{yy}} \approx -0.1$ [22]. This difference in magnitude manifests in different predicted behaviors for the spinning-rod experiement, in which a rod is inserted into the fluid and rotated at some frequency. The Oldroyd-B model accurately predicts that a viscoelastic fluid would climb the rod, whereas the Oldroyd-A model incorrectly predicts a depression surrounding the rod, similar to that seen in Newtonian fluids [19].

Another reason for the preference of the Oldroyd-B model is the sound theoretical basis of the upper convected derivative. Specifically, the UCM and Oldroyd-B equations can be derived from the elastic dumbbell model of viscoelastic fluids [18]. This connection between macroscopic and microscopic modelling offers additional insight into the behavior of viscoelastic solutions on multiple scales. For the same reason, the Oldroyd-B model is advantageous over alternative models, especially empirical equations. The Oldroyd-B model will be applied in our experiments, though comparison with alternative models is outside the scope of this thesis.

1.5 OpenFOAM and rheoTool

By solving the constitutive equations derived above, the behavior of a fluid can be ascertained, both in unsteady and steady-state regimes. Furthermore, the dynamics predicted by these models in various flow geometries can probe accuracy of the model and our understanding of the fluid. Finite differencing methods can be used in both spatial and temporal variables, a design suitable for implementation in computer programs. As a result, considerable research has been devoted to developing adequate simulation software, with quality evolving alongside computing capacity. Modern simulation software is capable of utilizing computing methods such as parallelization to maximize accuracy and speed of results. One such software is the Open Source Field Operation and Manipulation, or OpenFOAM, program developed by Henry Weller, David Gosman, and Drs. Issa and Jasak at the Imperial College of London [4]. Through the company OpenCFD, OpenFOAM was released as an open source CFD software in 2004, making it unique as most CFD programs of the time relied on peer-to-peer dissemination or academic licensing [4]. As an open source program, the quality and efficacy of OpenFOAM are actively maintained by both the owners and community collaborators through code-sharing sites such as GitHub. Additionally, the enhanced accessibility of open source code invites comparison and verification of published results.

To handle the spatial discretization, a CFD software would require a defined geometry and a method for dividing that geometry. OpenFOAM allows the user to define simulation geometry through the "blockMeshDict" file found within each simulation folder. In the "blockMeshDict" file, the user can first provide a list of points by providing x, y, and z coordinates. The points can then be arranged into three-dimensional blocks and two-dimensional planes called "boundaries". OpenFOAM utilizes these three-dimensional blocks, and any user-specified subdivisions, as a spatial discretization scheme to which the finite volume method (FVM) can be applied. FVM works by conserving numerical flux between cells, thus ensuring continuity between cells while applying the constitutive equations within cells to capture the evolution of characteristics across the spatial element [9]. In OpenFOAM, if a block face aligns with a user-specified boundary, as detected by comparison of plane normals, then one or more properties are modified to meet the given boundary condition. For example, one basic boundary condition provided in the OpenFOAM library is the *zeroGradient* condition, which requires the face flux to be equal to the internal field variable. This condition is commonly applied to the pressure at the outlet of the flow geometry when fully developed flows are expected. Therefore, in this scenario, the block face that aligns with the user-defined outlet would be subject to the constraint that the pressure at that face must be equal to the internal calculated pressure.

While geometric parameters are managed through the *blockMeshDict*, time discretization and accompanying variables are largely contained within the *controlDict*. Therein the user can define the size of the time step, the maximum simulation time, and the time interval at which data should be recorded. Additionally, the user has the option to allow the program to automatically adjust the time step within an acceptable range so that the Courant number (Co) is kept at a reasonable value. This can be a very useful tool considering Co, defined as $Co = u\Delta t/\Delta x$, strongly affects the stability and accuracy of the simulation, with values greater than or equal to 1 often leading to crashes or aphysical results. However, very small values for the time step, Δt, would result in very long simulation run times.

With time and spatial discretization parameters handled by the *controlDict* and *blockMeshDict*, respectively, the remaining system files in a simulation folder could be broadly classified as solver files. Of these files, two of the most important are *fvSchemes* and *fvSolution*. The *fvSchemes* file contains information for how the program should solve the constitutive equations, including time discretization method and methods for calculating gradients and other mathematical quantities. The *fvSolution* file includes various criteria the program must meet in its calculation steps, such as acceptable variation between solutions. Moreover, the *fvSolution* file allows the user to select from a range of solution algorithms included in OpenFOAM, which can dictate the workflow of the calculation steps or even specify how separate flow variables should be handled in the calculations. Considering these tools, OpenFOAM provides a workable platform for approaching complex flow calculations that may prove unyielding in alternative softwares.

The limitations of the original OpenFOAM software were apparent in the study of viscoelastic fluids, particularly at high Wi flows [34]. In the original OpenFOAM software, such simulations would be unstable and result in crashes due to the loss of positive definiteness in the conformation tensor [16]. This conformation tensor describes the deformation history of polymeric fluid elements, and a negative eigenvalue would demand physically invalid transformations that cannot be handled by the base simulation software [17]. To prevent this instability, taking the logarithm of the conformation tensor can contain the exponential growth of component terms, which occurs in flow regions of high deformation [16]. This is the solution employed by the open-source rheoTool software in its log conformation models. RheoTool builds and expands from OpenFOAM, offering numerous improvements and additional features, though for this thesis the log conformation modeling is the most relevant. For these reasons, rheoTool is chosen as the CFD software for our

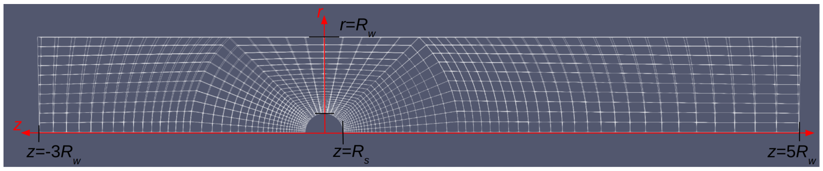

Figure 1: A Sample Mesh used for the Sedimenting Sphere ($n_{\text{base}} = 10$, $R_s = 0.02$ m, $R_w = 0.1$ m)

simulations.

2 Methods

2.1 Meshing

The first step in preparing a finite volume simulation is to create a suitable mesh; to balance competing demands for accuracy and speed, the mesh must be finely detailed near the sphere interface and sparsely detailed farther away [41]. Without sufficient resolution at the sphere surface, simulations would suffer from inaccurate or unstable results. In areas farther away from the sphere, the computational run time can be minimized by designing a coarser mesh without a significant loss in accuracy. Therefore, optimized mesh designs often partition the mesh into upstream, downstream, and sphere-containing blocks with different characteristic cell sizes. Transitions between these blocks must be sufficiently smooth to prevent instabilities caused by abrupt changes in cell size. Figure 1 shows a mesh designed for sedimenting sphere simulations, with a coarser meshing applied at the upstream and downstream positions and much finer meshing applied around the sphere.

Other potential sources of error or inaccuracy include mesh nonorthogonality, skew, and stretching. In the finite volume method, field variables are solved at cell centers by evaluation of the constitutive equation, and these values are then interpolated to face centers. The method of interpolation assumes that a vector connecting the centers of two adjacent cells passes through the center of the shared face, and an interpolation technique such as Gaussian quadrature or least-squares is applied along the vector [42]. This technique maintains second-order accuracy as long as the vector does pass through the face center [20]. In the case of nonorthogonal or skewed meshes however, this assumption becomes invalid and the interpolated values can become inaccurate estimates of field variables along the face of the cell.

Therefore, an appropriate mesh design should minimize nonorthogonality, especially in regions with sharp flow gradients, and maintain moderate aspect ratios between each dimension.

As shown in Figure 1, the potential issue of mesh nonorthogonality is handled by enclosing the sphere-containing block in arcs that are concentric with the sphere. The transition from the curved surface to the rectilinear wedge blocks is pushed to a coarser region away from the sphere, where the flow does not have any sharp gradients. We maintain moderate cell aspect ratios in the x-y plane by the use of appropriate scalars controlling the number cells along each axis in each block.

Balancing each of these design factors can be a challenging and iterative process. To facilitate the construction and validation of the discretization scheme, a custom script was written to generate mesh files with user-specified parameters. In addition to handling geometric alterations (e.g. the dimensions of the sphere, length/diameter of confining cylinder), the script allows the user to define the coarseness of the mesh in terms of a "base number" of cells. Beginning from a reference mesh of base number $n_{\text{base}} = 10$ as shown in 1, the density of grid lines in both the x and y directions is increased by a factor of the new base number n_{base}. The overall mesh resolution increases with increasing n_{base} while maintaining consistent relative resolution in the nearfield and farfield of the sphere surface. In this way, the influence of mesh resolution can be systematically explored.

To ensure that predictions are not impacted by the resolution of the mesh, we conducted a convergence test using a simple flow case with a steadily increasing inlet velocity. The maximum velocity is low so that both *Re* and *Wi* are kept on the order of 10^{-3} to 10^{-2}, which allows us to approximate the flow as Newtonian when calculating an expected drag force. Equation 26 calculates the Newtonian drag for a given sphere radius R_s, viscosity μ, and far-field (e.g. inlet) velocity $\boldsymbol{u}_\infty$.

$$\boldsymbol{F}_{D,N} = 6\pi\mu R_s \boldsymbol{u}_\infty \tag{26}$$

We first conducted a convergence test using the z-component of velocity as measured by a point probe. In the cylindrical coordinates used for Figure 1, we positioned this probe at $z = 2.5R_s$ and $r \approx 0$. Since field values are not extrapolated to the central axis, the probe could not be placed at exactly $r = 0$. For the purpose of convergence testing, the z-component of velocity measured with the $n_{\text{base}} = 100$ mesh was used as the "true" solution, and we calculated the error using Equation 27.

$$\epsilon = \sqrt{\Sigma_{i=0}^{n} (\frac{u_{z,i} - \hat{u_{z,i}}}{\hat{u_{z,n}}})^2} \tag{27}$$

Equation 27 expresses the experimental velocity component at time step i as $u_{z,i}$, for a total number of time steps n. The "true" solution is denoted with $\hat{u}_{z,i}$. The resulting convergence curve is shown in Figure 2.

As Figure 2 demonstrates, the error as calculated by the velocity at the probe point is confined to the fourth decimal place even for low n_{base}. Because our probe point is not necessarily representative of convergence across the full solution domain, we also test for convergence in terms of the normalized drag coefficient (C_D).

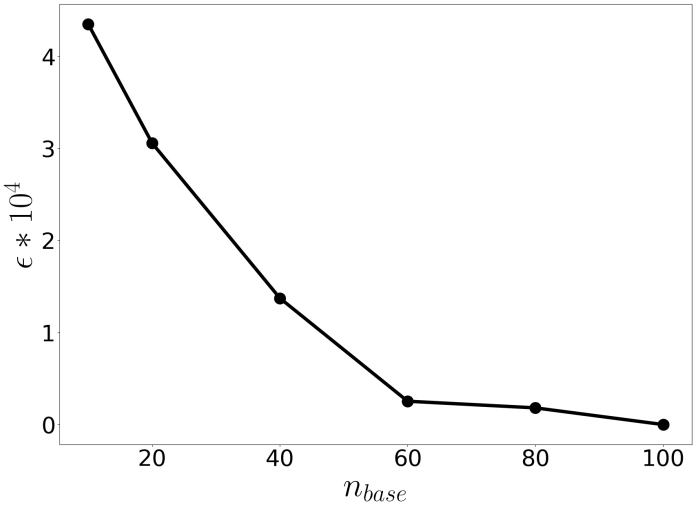

Figure 2: Convergence Test based on Velocity over Sphere ($\rho_f = 1000\ kg/m^3$, $\beta \approx 0.57$, $\mu = 14\ Pa \cdot s$, $\lambda = 0.8\ s$, $\frac{R_s}{R_w} = 0.2$, $u_{ss} = 0.001\ m/s$, $n = 2000$)

$$C_D = \frac{|\boldsymbol{F}_D|}{|\boldsymbol{F}_{D,N}|} \tag{28}$$

As Equation 28 shows, the drag coefficient is equal to the experimentally determined drag force, F_D, divided by the Newtonian drag force, $F_{D,N}$, as calculated by Equation 26. Figure 3 shows the values of C_D with increasing mesh refinent, along with simulation run time.

Figure 3 shows the tradeoff of increasing runtime with mesh refinement level, with a runtime of more than 8 hours for the $n_{\text{base}} = 100$ mesh. The change in C_D across the range of mesh refinements is on the order of 10^{-3}, with a difference between $n_{\text{base}} = 20$ and $n_{\text{base}} = 100$ of approximately $3 \cdot 10^{-3}$. Therefore, we chose to conduct our sedimenting-sphere simulations with an $n_{\text{base}} = 20$ mesh, as the drag coefficient and error show minimal change beyond that refinement level, and the runtime is moderate at less than 1 hour.

2.2 fvSchemes and fvSolutions

The fvSchemes file, as discussed in subsection 1.5, dictates the various discretization schemes used by rheoTool in its calculations. In general, our selection of solvers for individual field variables aligns well with past studies of viscoelastic flows in rheoTool [10, 26].

Spatial Derivatives: As outlined by previous work in rheoTool, the Gauss linear scheme for gradient and divergence evaluation is a suitable choice for many complex flow regimes [36]. This scheme uses Gaussian quadrature to obtain field values, interpolating between cell-center values and face-center values. The

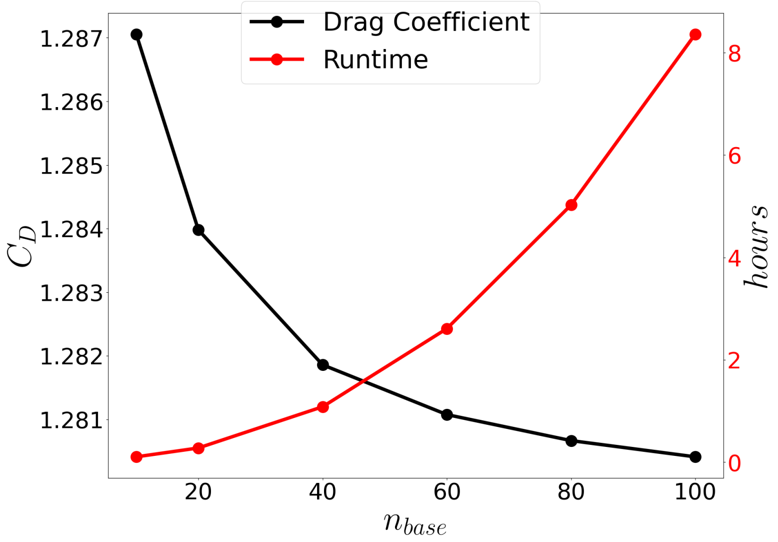

Figure 3: Convergence of Drag Coefficient with Runtime ($\rho_f = 1000\ kg/m^3$, $\beta \approx 0.57$, $\mu = 14\ Pa \cdot s$, $\lambda = 0.8$ s, $\frac{R_s}{R_w} = 0.2$, $u_{ss} = 0.001\ m/s$)

keyword “linear” then specifies the method for interpolation. The Laplacian schemes require an additional keyword for determining surface-normal gradients, with “corrected” being the most common choice. This scheme applies an implicit and explicit correction term, which can mitigate the influence of non-orthogonality between cell faces, ensuring numerical stability.

Linear Algebra (Preconditioner and Smoother): The function of a preconditioner is to produce a matrix of lower condition number, in order to produce a robust solution to an equation of the form $Ax = B$ [3]. In the case of a Cholesky preconditioner, the A matrix, which must be both positive definite and symmetric, is broken into a new candidate matrix L and the transpose of L, such that $A = LL^T$. The solver must then evaluate the matrix L, which should have a lower condition number. However, in some calculations the solution to L can yield subnormal numbers that are computationally expensive to compute and apply, reducing the efficacy of the preconditioner [27]. Subnormal numbers arise when the exponential representiation of the number requires an exponent with a magnitude exceeding some maximum allowable magnitude, resulting in underflow. As a countermeasure, the diagonal-based incomplete Cholesky method (DIC) replaces the subnormal values with zeros; the DIC preconditioner has proven suitable for numerical simulations of viscoelastic fluids in complex flows.

A smoother adjusts field values to ensure continuous and physically reasonable function interpolations between mesh points. The DIC method can function as a kind of smoother, with the A matrix becoming a smoothing matrix to be applied to x such that B is the new solution to field values. Because the DIC method requires x to be both positive definite and symmetric, an alternative preconditioner and solver is needed for asymmetric matrices. The diagonal-based incomplete LU (DILU) method is able to simplify the $Ax = B$ equation by factoring A, similar to the DIC method. However, instead of proposing a single matrix and its transpose, an LU method introduces a lower triangular matrix L and an upper triangular matrix U such that $A = LU$, where A can be asymmetric. Similar to a full Cholesky method, a regular LU algorithm can yield a matrix with high computational cost. The indirect LU approach solves this problem in the same manner as the DIC method: by replacing filled-in entries with zeros. This again gives an approximate solution of the form $A \approx LU$, but the resulting matrices are adequate as preconditioned candidates.

Pressure and Velocity Solvers: Through a trial-and-error exploration of parameter values, we found that the options provided by fvSolutions strongly influence the accuracy and stability of the simulation. For example, simultaneous solution of coupled pressure/velocity equations is a useful option in the rheoTool library that often allows for better accuracy and/or stability in numerical simulations of complex flows. For the specific problem of interest (unsteady sphere sedimentation), however, implementing pressure/velocity coupling leads to inaccurate and non-physical solutions. The issues with pressure/velocity coupling are only seen when the inlet velocity is unsteady, and they appear whether or not a sphere is present to obstruct the flow (i.e. they can arise due to flow inertia alone). A complete explanation of the underlying mathematical problem with pressure-velocity coupling is beyond the scope of this thesis, and elucidating the exact cause would require further investigation. Regardless, pressure-velocity coupling is evidently not compatible with the boundary conditions for the present study, and good results have been obtained by disabling this option in rheoTool's library of numerical methods.

We used the generalized geometric-algebraic multi-grid (GAMG) solver for both the pressure and velocity fields with a DIC preconditioner and smoother. Rather than attempting to solve all field variables at once across the entire mesh, the GAMG solver functions by coarsening the mesh, performing calculations within the coarsened grids, interpolating to predict values across the finer grid, then finally comparing residuals with those of the finer grid to ascertain accuracy via agreement [11]. This process can be implemented with additional layers of mesh coarsening, effectively producing a hierarchy of grids with increasing coarseness and faster individual solution times. While this can generally describe any multi-grid method, the exact technique used for coarsening differs between geometric multi-grid (GMG) and algebraic multi-grid (AMG) approaches. The GMG method simply reduces the number of points within the grids, and this works best for problems with well-defined geometries [39]. Alternatively, AMG performs coarsening through direct interaction with matrix values, producing matrices of reduced dimensions that constitute coarser "grids" [37]. This can be useful when exact geometric information is not available. Furthermore, in calculations running on few cores with low stencil sizes, the AMG method offers improvements in run time compared to GMG [38]. In order to leverage the benefits of both techniques, the GAMG solver performs larger calculations using GMG, and delegates coarse-grid solutions to AMG solvers. This method can be efficiently implemented in a parallel computing architecture as separate grid operations can be allocated to separate cores [38].

Overall Solver Algorithm: Having established the methods of numerical evaluation, the fvSolution file allows the user to specify an algorithm for solving the constitutive equations. As with our choice of solvers, we based our selection of solution algorithm on previous work with viscoelastic fluids. Therefore, we chose the semi-implicit method for pressure-linked equations, or SIMPLE, algorithm [12]. The SIMPLE algorithm begins with an estimate of the pressure, which is then used to propose a preliminary estimate of other field variables, with corrective adjustments applied iteratively [31]. Within this algorithm, the user can specify the number of iterations and the number of corrective steps through which a converged solution can be assured. One useful modification to the SIMPLE algorithm proposed by Doormaal and Raithby is the consistent application of an assumption found in the original SIMPLE algorithm [6]. Specifically, the SIMPLE algorithm assumes that summation across neighboring cells of velocity correction terms multiplied by finite volume coefficients is a negligible term in the velocity correction equation. This assumption is used to eliminate one such term from the equation, but leaves another in place. The consistent formulation of SIMPLE, or SIMPLEC, eliminates this other term. The resulting rearrangment via SIMPLEC allows for the removal of an underrelaxation coefficient in the pressure correction equation, further simplifying the algorithm. The SIMPLE or SIMPLEC algorithm structure also allows for the coupling of the pressure equation to velocity and shear stress so that the field values are solved concurrently. This technique can be valuable in obtaining accurate steady-state solutions, though, we have found this coupling causes our simulations to diverge and crash, as previously mentioned.

While SIMPLEC negates the influence of a pressure field relaxation factor, we have found that the remaining field and equation relaxation factors are vital to the stability of the simulation. In an iterative process, the “relaxation factor” modulates the rate at which the solution advances from its current value to the presumed “next best guess”. A low relaxation factor is more conservative, with previous guesses strongly factoring into each iteration. Likewise, a field relaxation factor approaching unity is more aggressive (and potentially unstable), relying on the “next best guess” without being restrained by past guesses. In simple flowcases, either approach may be sufficient to reach the same, converged solution. Being an unsteady simulation with a viscoelastic fluid, our sedimenting sphere simulation is only stable when running at low relaxation values. Higher relaxation factors, particularly in the velocity field, may attempt to smooth over field gradients, leading to nonphysical flow fields and, ultimately, a simulation crash. With these considerations in mind, we opted for field and equation relaxation parameters of 0.1, allowing for some slight correction without overcorrecting.

2.3 Boundary Conditions

The boundaries of the sedimenting sphere mesh, as shown in Figure 1, include 6 defined faces: the inlet, outlet, wall, sphere surface, and the front and back faces. The velocity inlet boundary condition for our simulation is controlled by a momentum balance on the sphere, while the outlet is assumed to have a developed flow such that the velocity gradient is equal to 0. The pressure at the inlet is also assumed to have a gradient equal to 0, thus assuming a developed flow without overspecifying the system. The outlet pressure is assigned a value of 0 Pa. The wall of the mesh is treated with an OpenFOAM-defined "slip" condition. Our intention in selecting this boundary is to approximate the behavior of a freely sedimenting sphere in an unbounded domain, absent of wall effects that would be expected in a confined domain with a "no-slip" condition. A commonly used alternative approach is to apply the Faxen wall correction for low Re, as shown in Equation 29 [13].

$$\frac{u}{u_\infty} = 1 - 2.104(R_s/R_w) + 2.09(R_s/R_w)^3 - 0.95(R_s/R_w)^5 \tag{29}$$

We will compare the performance of the slip condition to that of Equation 29 at various R_s/R_w by calculating the ratio of our experimental $\boldsymbol{u}$ to the Newtonian free-stream velocity, $\boldsymbol{u}_\infty$.

A standard "no-slip" condition was applied to the sphere surface itself. The front and back faces of the mesh enforce the axisymmetric approximation through the "wedge" condition, which enforces mass conservation but otherwise does not interfere with flow.

2.4 Sedimenting Sphere Momentum Balance

In order to capture the unsteady velocity of a sedimenting sphere, we implemented an immersed boundary condition consisting of a momentum balance on the sphere enforcing a particular inlet velocity. By constructing a force balance on the sphere in which the buoyant force is subtracted from the summation of surface forces, we can calculate the acceleration of the sphere per Equation 30.

$$\frac{d\boldsymbol{u}}{dt} = \frac{\int_A [\boldsymbol{\tau} + (\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)\mu_s - p\boldsymbol{I}] \cdot \boldsymbol{n} dA - (\rho_f - \rho_s)V_s \boldsymbol{a}}{m_s} \tag{30}$$

In Equation 30, V_s is the volume of the sphere and m_s is the mass of the sphere. The acceleration, a, applied in computing the buoyant force would, ideally, be equal to gravitational acceleration. However, we found that the simulation was unstable for realistic values of m_s with a equal to gravitational acceleration. The probable cause of the instability was the that the small sphere mass in the denominator of Equation 30 resulted in a large $d\boldsymbol{u}/dt$; greater acceleration can result in unstable simulations due to rapidly changing

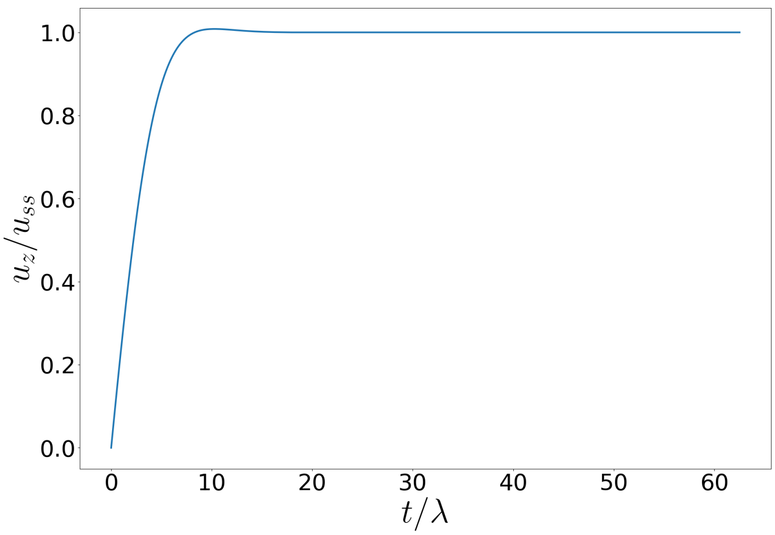

Figure 4: Sample Inlet Velocity Profile with ($n_{\text{base}} = 20$, $\rho_f = 1000\ kg/m^3$, $\frac{\Delta\rho}{\rho_f} = 748$, $a = 9.10 \cdot 10^{-7}\ m/s^2$, $\beta \approx 0.57$, $\mu = 14\ Pa \cdot s$, $\lambda = 0.8\ s$, $\frac{R_s}{R_w} = 0.2$)

field variables. To stabilize the simulation, we chose a large sphere mass by specifying an unrealistically high sphere density, which dampened the acceleration $d\boldsymbol{u}/dt$. The buoyant force would have grown proportionally, so to compensate for these densities we calculated accelerations such that the quantity $(\rho_f - \rho_s)a$ resulted in the desired steady state velocities. In future work we hope to develop techniques that can stabilize the simulation when the true sphere mass is used. As found in the integral of Equation 30, A is the surface area of the sphere, and $\boldsymbol{n}$ is the surface unit normal on the sphere. Once this calculation is completed, the simulation can prepare an estimate for the next inlet velocity by applying a numerical time derivative. We chose to use Euler's method for this purpose, resulting in Equation 31.

$$u_{z,n+1} = u_{z,n} + t\frac{\partial u_{z,n}}{\partial t} \tag{31}$$

Here the subscript n represents a given timestep with $n+1$ denoting the subsequent time step, so that $u_{z,n}$ is the recorded inlet velocity at timestep n and $\partial u_{z,n}/\partial t$ is the corresponding acceleration as calculated by Equation 30. The inlet velocity for the next timestep can then be set equal to $u_{z,n+1}$ and the simulation can proceed accordingly. As a demonstration of this boundary condition, Figure 4 plots the inlet velocity of a sedimenting sphere in a viscoelastic fluid at $Wi \approx 1 \cdot 10^{-2}$ and $Re \approx 4 \cdot 10^{-3}$. The relaxation time λ is 0.8 seconds.

$\rho_s(kg/m^3)$	$a\ (m/s^2)$	$\Delta\rho/\rho_f$	$u_{ss}(m/s)$	Wi	Re	El
$7.49 \cdot 10^5$	$9.10 \cdot 10^{-7}$	748	$3.11 \cdot 10^{-4}$	0.0125	0.00444	2.80
$7.49 \cdot 10^5$	$6.55 \cdot 10^{-4}$	748	0.00311	0.125	0.0444	2.80
$7.49 \cdot 10^5$	0.00524	748	0.0249	0.997	0.356	2.80
$7.49 \cdot 10^5$	0.0157	748	0.0747	2.99	1.07	2.80
$7.49 \cdot 10^5$	0.0262	748	0.125	4.98	1.78	2.80
$7.49 \cdot 10^5$	0.0524	748	0.249	9.97	3.56	2.80

Table 1: Predicted Simulation Parameters for Various Sphere Densities

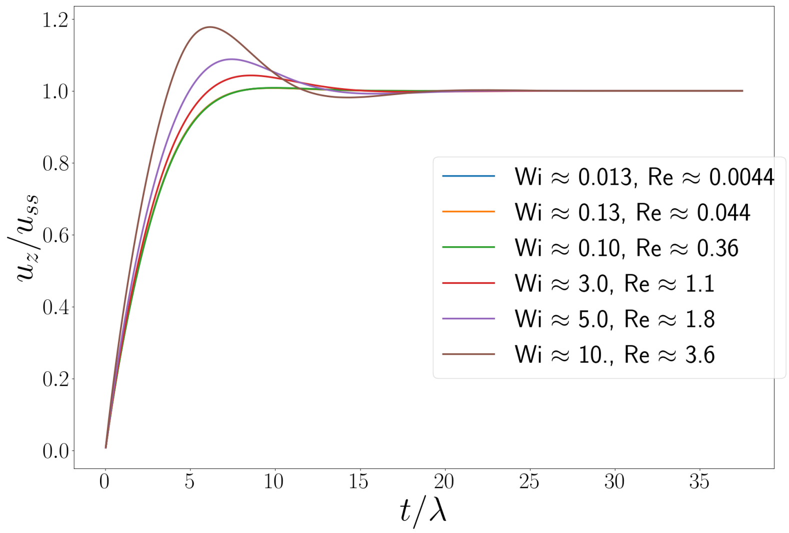

Figure 5: Scaled Velocities for Spheres of Increasing Density ($n_{\text{base}} = 20$, $\rho_f = 1000\ kg/m^3$, $\frac{\Delta\rho}{\rho_f} = 748$, $\beta \approx 0.57$, $\mu = 14\ Pa \cdot s$, $\lambda = 0.8\ s$, $\frac{R_s}{R_w} = 0.2$)

3 Results and Discussion

3.1 Sedimenting Sphere of Increasing Density

In order to examine the effect of increasing Wi on sedimenting sphere velocity, we prepared simulations featuring a range of sphere densities in a highly viscous medium with $\mu = 14\ Pa \cdot s$, $\beta \approx 0.57$, $\lambda = 0.8\ s$, and the approximate density of water, $1000\ kg/m^3$. The increasing sphere densities resulted in the steady-state velocities, Wi, Re, and El shown in Table 1.

The velocities of these spheres, scaled by their corresponding steady-state velocities are shown in Figure 5.

As the figure demonstrates, for increasing Wi, the velocity overshoot increases, and, in the cases of $Wi \approx 5$ and $Wi \approx 10$, is accompanied by a proportionate undershoot before reaching the steady-state value. In Figure 6, we plot the values of these overshoots, which may be considered a ratio of the maximum velocity to the steady-state velocity, illustrating their dependence on Wi.

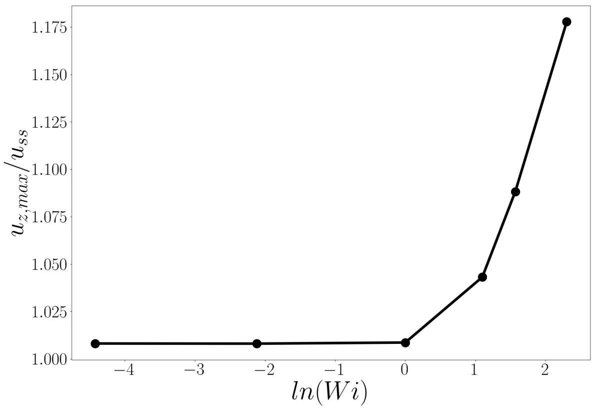

Figure 6: Velocity Overshoot Coefficients as a Function of $ln(Wi)$ ($n_{\text{base}} = 20$, $\rho_f = 1000\ kg/m^3$, $\frac{\Delta\rho}{\rho_f} = 748$, $\beta \approx 0.57$, $\mu = 14\ Pa \cdot s$, $\lambda = 0.8\ s$, $\frac{R_s}{R_w} = 0.2$)

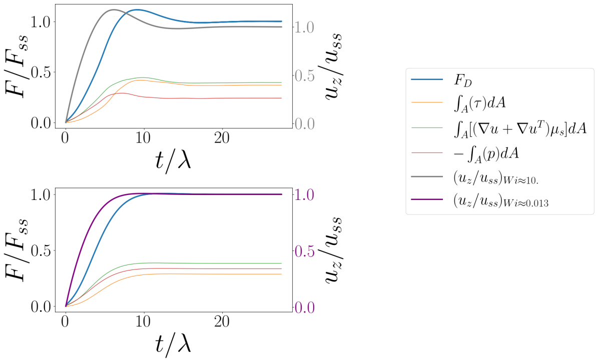

Figure 7: Drag Force with each Term for $Wi \approx 10$ and $Wi \approx 0.013$ ($n_{\text{base}} = 20$, $\rho_f = 1000\ kg/m^3$, $\frac{\Delta\rho}{\rho_f} = 748$, $\beta \approx 0.57$, $\mu = 14\ Pa \cdot s$, $\lambda = 0.8\ s$, $\frac{R_s}{R_w} = 0.2$)

Figure 6 confirms that, for $Wi \leq 1$ or $\ln(Wi) \leq 0$, the sedimenting sphere in a viscoelastic medium shows approximately Newtonian dynamics. The increasing overshoot with respect to Wi marks a growing delay in the viscous drag response, as is expected in a viscoelastic fluid [1]. To further investigate the dynamics driving velocity overshoot at high Wi, we calculated the values of each term in the drag force equation, as given in Equation 30, using OpenFOAM's *controlDict*. In Figure 7, we plot these terms alongside the total drag force for $Wi \approx 10$ and $Wi \approx 0.013$.

Comparing the plots of individual drag force terms in Figure 7, the relative contribution from the form drag (i.e. pressure) appears to be lower at higher Wi with an overshoot preceding that of the overall drag force. Further conclusions may require additional data, though this figure demonstrates a potential modality for mathematical analysis in rheoTool.

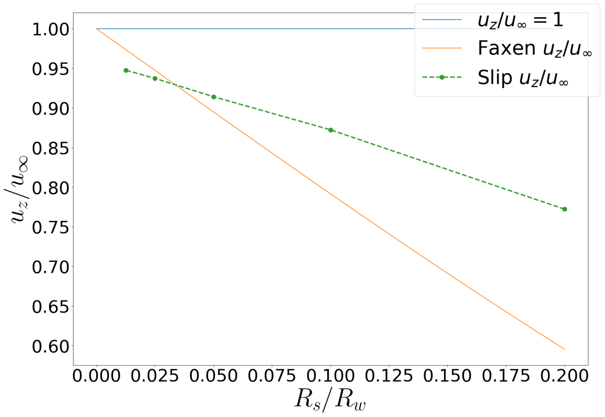

Figure 8: Comparison of Slip Condition with Faxen Correction at Steady State ($n_{\text{base}} = 20$, $\rho_f = 1000$ kg/m^3, $\frac{\Delta\rho}{\rho_f} = 748$, $a = 9.10 \cdot 10^{-7}$ m/s^2, $\beta \approx 0.57$, $\mu = 14$ $Pa \cdot s$, $\lambda = 0.8$ s)

3.2 Comparison with Faxen Wall Correction

We compare the value of u_z/u_∞ to that predicted by the Faxen wall correction, as given in Equation 29. While our unsteady dynamics are impacted by our stabilizing use of a high sphere density and varying gravitational acceleration, the steady state velocities are comparable to those of realistic spheres under typical gravitational acceleration. In containing both ρ_s and our adjusted acceleration a, the quantity $(\rho_s - \rho_f)a$, which is part of the driving buoyant force, has the same magnitude as $(\rho_s - \rho_f)g$ for much lower values of ρ_s. For example, the sphere in Table 1 with $\rho_s = 7.49 \cdot 10^5$ kg/m^3 and $a = 9.10 \cdot 10^{-7}$ m/s^2 gives approximately the same buoyant force as a sphere with $\rho_s \approx 1005$ kg/m^3 and $a = 9.8$ m/s^2, and both buoyant forces yield a steady state velocity of approximately $3 \cdot 10^{-4}$ m/s. Therefore, while our unsteady dynamics will require refinement in future work, our steady state results can be analyzed and compared with existing experimental data. In particular, we compare our steady state velocities under our set of boundary conditions to the velocities predicted by the Faxen correction, shown in Equation 29.The result is given in Figure 8.

Figure 8 shows that, for part of the range of R_s/R_w values tested at low Re, the slip condition offers a slightly better estimate of the velocity for a freely sedimenting sphere. The Faxen correction equation appears to offer better predictions at low R_s/R_w, however the R_s/R_w value for our simulations presented in Figure 5 was 0.2. At this value, the slip condition offers a better estimate of freely sedimenting velocity. This suggests that applying the slip condition could be a better technique for this geometry than using a standard no-slip condition and subsequently applying a correction equation.

4 Conclusion

We have demonstrated how OpenFOAM can be modified to simulate an unsteady, sedimenting sphere to approximate startup dynamics by solving a momentum balance on the sphere. In future work we hope to stably implement an accurate momentum balance based on the mass of the sphere. Moreover, we can probe the behavior of specific terms in the drag force to gain insight into viscoelastic fluid mechanics. While this thesis explores the onset of elastic force with a moderate Wi of 10, the methods applied can be expanded to study elastic instability at high Wi. Thus future research building on the methods developed here may be capable of offering new models of elastic instability accompanied by detailed mechanistic explanations. These methods can also be used to probe the parameter space in the sedimenting sphere problem, so that the behavior of viscoelastic fluids in this geometry can be predicted and understood up to and including the onset of elastic instability.

We also demonstrate that the use of OpenFOAM's "slip" boundary condition along the walls of the wedge gives a closer estimate of a freely sedimenting sphere than the Faxen wall correction. In combination with the prior conclusions, we show that our simulation design offers an improved method for the simulation of a freely sedimenting sphere.

References

[1] M.T Arigo, D Rajagopalan, N Shapley, and Gareth H McKinley. The sedimentation of a sphere through an elastic fluid. part 1. steady motion. *Journal of Non-Newtonian Fluid Mechanics*, 60(2):225–257, 1995.

[2] L. Cammoun, C.A. Castaño-Moraga, E. Muñoz-Moreno, D. Sosa-Cabrera, B. Acar, M.A. Rodriguez-Florido, A. Brun, H. Knutsson, and J. P. Thiran. *A Review of Tensors and Tensor Signal Processing*, pages 1–32. Springer London, London, 2009.

[3] David Casasent and Anjan Ghosh. Lu and cholesky decomposition on an optical systolic array processor. *Optics Communications*, 46(5):270–273, 1983.

[4] Goong Chen, Qingang Xiong, Philip J Morris, Eric G Paterson, Alexey Sergeev, and Y Wang. Openfoam for computational fluid dynamics. *Notices of the AMS*, 61(4):354–363, 2014.

[5] R H Davis and A Acrivos. Sedimentation of noncolloidal particles at low reynolds numbers. *Annual Review of Fluid Mechanics*, 17(1):91–118, 1985.

[6] J. P. Van Doormaal and G. D. Raithby. Enhancements of the simple method for predicting incompressible fluid flows. *Numerical Heat Transfer*, 7(2):147–163, 1984.

[7] Albert Einstein et al. On the motion of small particles suspended in liquids at rest required by the molecular-kinetic theory of heat. *Annalen der physik*, 17(549-560):208, 1905.

[8] Leonhard Euler. Principes généraux du mouvement des fluides. *Mémoires de l'académie des sciences de Berlin*, pages 274–315, 1757.

[9] Robert Eymard, Thierry Gallouet, and Raphaele Herbin. Finite volume methods. In *Solution of Equation in R, Techniques of Scientific Computing (Part 3)*, volume 7 of *Handbook of Numerical Analysis*, pages 713–1018. Elsevier, 2000.

[10] Ahmad Fakhari and Celio Fernandes. Single-bubble rising in shear-thinning and elastoviscoplastic fluids using a geometric volume of fluid algorithm. *Polymers*, 15(16), 2023.

[11] Robert D Falgout. An introduction to algebraic multigrid. Technical report, Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 2006.

[12] J.L. Favero, A.R. Secchi, N.S.M. Cardozo, and H. Jasak. Viscoelastic fluid analysis in internal and in free surface flows using the software openfoam. *Computers Chemical Engineering*, 34(12):1984–1993, 2010. 10th International Symposium on Process Systems Engineering, Salvador, Bahia, Brasil, 16-20 August 2009.

[13] V Fidleris and R L Whitmore. Experimental determination of the wall effect for spheres falling axially in cylindrical vessels. *British Journal of Applied Physics*, 12(9):490, sep 1961.

[14] Richard Fitzpatrick. Isotropic tensors, January 2016.

[15] Y. C. Fung. *Bio-viscoelastic Fluids*, pages 174–195. Springer New York, New York, NY, 1981.

[16] Florian Habla, Ming Wei Tan, Johannes Hasslberger, and Olaf Hinrichsen. Numerical simulation of the viscoelastic flow in a three-dimensional lid-driven cavity using the log-conformation reformulation in openfoam. *Journal of Non-Newtonian Fluid Mechanics*, 212:47–62, 2014.

[17] Ismail Hameduddin, Charles Meneveau, Tamer A. Zaki, and Dennice F. Gayme. Geometric decomposition of the conformation tensor in viscoelastic turbulence. *Journal of Fluid Mechanics*, 842:395â427, 2018.

[18] Markus Herrchen and Hans Christian Ãttinger. A detailed comparison of various fene dumbbell models. *Journal of Non-Newtonian Fluid Mechanics*, 68(1):17–42, 1997.

[19] John Hinch and Oliver Harlen. Oldroyd b, and not a? *Journal of Non-Newtonian Fluid Mechanics*, 298:104668, 2021.

[20] Tomoaki Ikeda and Paul A. Durbin. Mesh stretch effects on convection in flow simulations. *Journal of Computational Physics*, 199(1):110–125, 2004.

[21] Anthony J. C. Ladd. Dynamical simulations of sedimenting spheres. *Physics of Fluids A: Fluid Dynamics*, 5(2):299–310, 02 1993.

[22] Ronald G. Larson. Chapter 3 - continuum theories. In Ronald G. Larson, editor, *Constitutive Equations for Polymer Melts and Solutions*, Butterworths Series in Chemical Engineering, pages 59–91. Butterworth-Heinemann, 1988.

[23] L.G. Leal, M.M. Denn, and R. Keunings. Lake arrowhead workshop special issue papers-introduction. *Journal of Non-Newtonian Fluid Mechanics*, 29:1–8, 1988.

[24] O. Maklad and R.J. Poole. A review of the second normal-stress difference; its importance in various flows, measurement techniques, results for various complex fluids and theoretical predictions. *Journal of Non-Newtonian Fluid Mechanics*, 292:104522, 2021.

[25] David S. Malkus, John A. Nohel, and Bradley J. Plohr. Dynamics of shear flow of a non-newtonian fluid. *Journal of Computational Physics*, 87(2):464–487, 1990.

[26] Anele Mavi and Tiri Chinyoka. Finite volume computational analysis of the heat transfer characteristic in a double-cylinder counter-flow heat exchanger with viscoelastic fluids. *Defect and Diffusion Forum*, 424:19–43, 6 2023.

[27] Thai Anh Nhan and Niall Madden. An analysis of diagonal and incomplete cholesky preconditioners for singularly perturbed problems on layer-adapted meshes. *Journal of Applied Mathematics and Computing*, 65, 07 2020.

[28] J. G. Oldroyd. A rational formulation of the equations of plastic flow for a bingham solid. *Mathematical Proceedings of the Cambridge Philosophical Society*, 43(1):100–105, 1947.

[29] Tim Osswald and Natalie Rudolph. Polymer rheology. *Carl Hanser, München*, 2015.

[30] Peyman Pakdel and Gareth H. McKinley. Elastic instability and curved streamlines. *Phys. Rev. Lett.*, 77:2459–2462, Sep 1996.

[31] S. Patankar. *Numerical Heat Transfer and Fluid Flow*. CRC Press, 2018.

[32] J Pedlosky. The stress tensor for a fluid and the navier stokes equations. *Fluid Dynamics of the Atmosphere and Ocean*, page 32, 2014.

[33] Christopher J. S. Petrie and Morton M. Denn. Instabilities in polymer processing. *AIChE Journal*, 22(2):209–236, 1976.

[34] F. Pimenta and M.A. Alves. Stabilization of an open-source finite-volume solver for viscoelastic fluid flows. *Journal of Non-Newtonian Fluid Mechanics*, 239:85–104, 2017.

[35] Devranjan Samanta, Yves Dubief, Markus Holzner, Christof Schafer, Alexander N. Morozov, Christian Wagner, and Bjorn Hof. Elasto-inertial turbulence. *Proceedings of the National Academy of Sciences*, 110(26):10557–10562, 2013.

[36] Takshak Shende, Vahid Niasar, and Masoud Babaei. Upscaling non-newtonian rheological fluid properties from pore-scale to darcys scale. *Chemical Engineering Science*, 239:116638, 2021.

[37] Klaus Stüben et al. An introduction to algebraic multigrid. *Multigrid*, pages 413–532, 2001.

[38] Hari Sundar, George Biros, Carsten Burstedde, Johann Rudi, Omar Ghattas, and Georg Stadler. Parallel geometric-algebraic multigrid on unstructured forests of octrees. In *SC '12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, pages 1–11, 2012.

[39] P. Wesseling and C.W. Oosterlee. Geometric multigrid with applications to computational fluid dynamics. *Journal of Computational and Applied Mathematics*, 128(1):311–334, 2001. Numerical Analysis 2000. Vol. VII: Partial Differential Equations.

[40] Shijian Wu and Hadi Mohammadigoushki. Sphere sedimentation in wormlike micelles: Effect of micellar relaxation spectrum and gradients in micellar extensions. *Journal of Rheology*, 62(5):1061–1069, 09 2018.

[41] Bin Yang and Bamin Khomami. Simulations of sedimentation of a sphere in a viscoelastic fluid using molecular based constitutive models. *Journal of Non-Newtonian Fluid Mechanics*, 82(2):429–452, 1999.

[42] Donghyun You, Rajat Mittal, Meng Wang, and Parviz Moin. Analysis of stability and accuracy of finite-difference schemes on a skewed mesh. *Journal of Computational Physics*, 213(1):184–204, 2006.